Adaptive Methods for Robust Document Image Understanding

Dissertation

zur
Erlangung des Doktorgrades (Dr. rer. nat.)
der
Mathematisch-Naturwissenschaftlichen Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt
von

Iuliu Konya
aus
Baia Mare

Bonn, Juli 2012

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinischen Friedrich-Wilhelms-Universität Bonn

1. Gutachter: Prof. Dr. Christian Bauckhage

2. Gutachter: Prof. Dr. Reinhard Klein

Tag der Promotion: 13.03.2013

Erscheinungsjahr: 2013

Adaptive Methods for
Robust Document Image Understanding

Iuliu Konya

Abstract

A vast amount of digital document material is continuously being produced as part of major digitization efforts around the world. In this context, generic and efficient automatic solutions for document image understanding represent a stringent necessity. We propose a generic framework for document image understanding systems, usable for practically any document types available in digital form. Following the introduced workflow, we shift our attention to each of the following processing stages in turn: quality assurance, image enhancement, color reduction and binarization, skew and orientation detection, page segmentation and logical layout analysis. We review the state of the art in each area, identify current deficiencies, point out promising directions and give specific guidelines for future investigation. We address some of the identified issues by means of novel algorithmic solutions putting special focus on generality, computational efficiency and the exploitation of all available sources of information. More specifically, we introduce the following original methods: a fully automatic detection of color reference targets in digitized material, accurate foreground extraction from color historical documents, font enhancement for hot metal typesetted prints, a theoretically optimal solution for the document binarization problem from both computational complexity- and threshold selection point of view, a layout-independent skew and orientation detection, a robust and versatile page segmentation method, a semi-automatic front page detection algorithm and a complete framework for article segmentation in periodical publications. The proposed methods are experimentally evaluated on large datasets consisting of real-life heterogeneous document scans. The obtained results show that a document understanding system combining these modules is able to robustly process a wide variety of documents with good overall accuracy.

Keywords: document image analysis, document image understanding, image enhancement, character enhancement, document binarization, color reduction, skew detection, orientation detection, page segmentation, geometric layout analysis, article segmentation, logical layout analysis

Contents

Chapter 1

Introduction

Science, my boy, is composed of errors, but errors that it is right to make, for they lead step by step to the truth.

– J. Verne *(A Journey to the Centre of the Earth)*

Over the last decade, an ever increasing number of cultural heritage organizations along with major publishers around the world started their (joint) efforts for the complete digitization of their assets. Consequently, millions of books, old manuscripts, periodicals, paintings, films, analog audio storage media, museum objects and archival records exist now in digital form and many more still await digitization. Well-known examples of mass digitization projects are: Google's Books (formerly known as Book Search) [8, 278], the Theseus project (with its use case Contentus [17]) endorsed by the German Federal Ministry of Economics and Technology, the IMPACT project [10] supported by the European Commission, the German Digital Library (Deutsche Digitale Bibliothek) [6], Europeana [7] and the European Library [16]. Among the content providers involved in these projects are many world-renowned universities, such as Harvard, Stanford, Oxford, major libraries, like the German National Library, the New York Public Library, the British library, the French National Library as well as famous museums, such as the Louvre from Paris and the Rijksmuseum in Amsterdam. At the end of 2010, the European Library already contained over 24 million pages of full text content along with another 7 million digital objects coming from more than 2000 institutions across Europe. One of the youngest digitization projects, the German Digital Library has the ambitious goal of building a common digital repository for no less than 30 000 cultural and scientific institutions. Even more impressive is the envisioned end goal of combining the results of the projects – an initiative which is already underway. An example in this direction is the integration of the research results, content and software infrastructure from Theseus, IMPACT, the European Library and the German Digital Library into the Europeana portal.

Many factors contribute to the sustained growth in the number and size of digitization projects: storage space is now more plentiful, more reliable and cheaper than ever, digitization solutions offering good quality are affordable and the Internet infrastructure allows access from theoretically all over the world – thus offering the potential to reach out to huge audiences. In addition, electronic documents in special present many advantages over their paper-based counterparts: the existence of a logical document structure allows the application of a multitude of electronic document tools, including markup, hyperlinking, hierarchical browsing and component-based retrieval [256]. The high-level structuring of the information allows humans to handle electronic documents in a manner which is more natural and efficient. In this sense, the portals now in construction are a concrete step forward towards the reality of a true “semantic Web”.

For the extraction of rich semantic information from the vast and diverse digital material produced, there is a stringent need for intelligent automatic analysis algorithms. Unfortunately at his point, a generic solution for this problem is only foreseeable far in the future. However, systems capable of (partially) analyzing images belonging to more restricted classes of digitized material/documents have been proposed in the specialized literature. Among printed documents, structured materials such as books and periodicals doubtlessly represent one, if not THE most information-rich document types. Printed periodicals have had a very long existence, many still existing major publishers being founded more than two centuries ago. The earliest printed books date even much earlier than this and go back as far as the year 1440 – year in which the printing press was invented by the German goldsmith Johannes Gutenberg. While the current work shall have as main focus *structured documents*, many of the methods described in chapter 2 are equally applicable to any kind of digital still images.

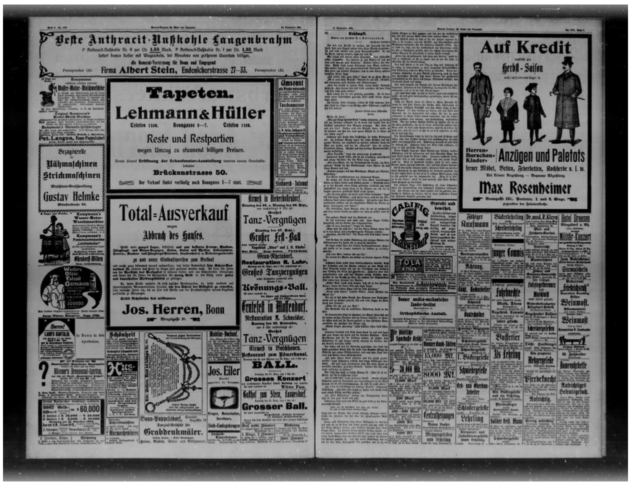
Figure 1.1: Mass processing of digitized periodicals and books is the main focus of the current work, as they represent two of the most widespread and information-rich printed media types

1.1 Motivation and Goals

A large number of scientific articles have been published on topics concerning specific areas of document image understanding, as one may see from the sheer number of available reviews [66, 126, 142, 177, 178, 198, 201, 202, 239, 252]. The prevailing focus of scientific work is on structured and semi-structured documents, most notably (scientific) journals, forms and books. Despite this fact, the document image understanding problem for such materials still remains unsolved. Consequently, no fully automated solution exists (neither in academia nor on the market) which can be employed with reliable, satisfactory results for the problem of understanding structured document images. One of the main causes for this situation is that many proposed systems are described as integrated, monolithic solutions [114, 195, 273]. Such integrated solutions generally have a higher runtime efficiency, tend to achieve better results for restricted data sets due to easier parameter tuning and can be developed faster than modular systems. However, these advantages come at the cost of low flexibility and the inability of performing partial tests by using standardized data sets. Recently, the need for modular systems coupled with exact benchmarks on standardized and diverse data sets for algorithms belonging to all document processing stages has been recognized as an important issue by many authors [37, 177, 201].

The number of publications covering parts of the topic of document image understanding has been growing at a very fast pace. This resulted in the unfortunate situation that at present practically no up-to-date exhaustive surveys exist. While many surveys have been published in the last two decades, such as [66, 126, 142, 177, 178, 198, 201, 202, 239, 252], however one may easily notice that they are either not up-to-date or cover just highly restricted areas.

Another issue worth mentioning is that mass digitization projects pose the requirement that automatic approaches cope with a vast variety of printed document layouts. A more recent comparison of page segmentation algorithms by Shafait et al. [239] shows that no single method is uniformly optimal. As argued by Baird and Casey [52], versatility is the key requirement for successful document analysis systems working on large scale collections. Even for the same publisher, the layout of its publications changes drastically over time. This is especially visible when looking at samples of publications scattered over the time span of several decades or centuries. As a general rule, the more recently printed documents shown an increased layout complexity and the difference between the layouts used by different publishers becomes more pronounced. Thus it is highly challenging to have algorithms consistently delivering good results over the whole possible range of document layouts, even without considering the additional complication posed by image quality variations.



Beſte Anthracit-Nußkohle Langenbrahm

Firma Albert Stein, Endenicherstrasse 27–33.

Tapeten.

Lehmann & Hüller

Telefon 1108. Bonngasse 5—7. Telefon 1108.

Reste und Restpartien

wegen Umzug zu staunend billigen Preisen.

Brückenstrasse 50.

Nähmaſchinen

Strickmaſchinen

Gustav Helmke

Total-Ausverkauf

Abbruch des Hauſes.

Jos. Herren, Bonn

Tanz-Vergnügen

Großer Feſt-Ball

Restauration H. Lühr.

Großes Tanzvergnügen

Krönungs-Ball.

Erntefeſt in Muffendorf.

Tanz-Vergnügen

BALL.

Grosses Konzert

Grosser Ball.

Jos. Eiler

Grabdenkmäler.

Auf Kredit

Herbſt-Saiſon

Herren- Burschen- Kinder- Anzügen und Paletots

ferner Möbel, Betten, Federbetten, Kochherde u. ſ. w.

Max Rosenheimer

CALIFIG

FEIGEN-SYRUP

TOLA

Orthopädische Anstalt.

Junger Commis

Buffetier

Als Lehrling

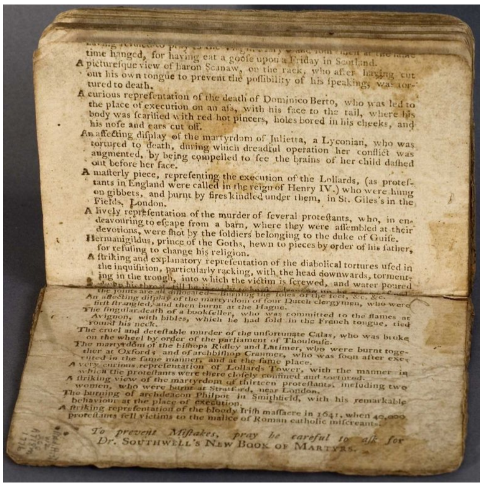
Pferdeknecht

Weinmoſt.

15,000 Mk.

8000 Mk.

Lehrling



time hanged, for having eat a gooſe upon a Friday in Scotland.

A pictureſque view of baron Scanaw, on the rack, who after having cut out his own tongue to prevent the poſſibility of his ſpeaking, was tortured to death.

A curious repreſentation of the death of Dominico Berto, who was led to the place of execution on an aſs, with his face to the tail, where his body was ſcarified with red hot pincers, holes bored in his cheeks, and his noſe and ears cut off.

An affecting diſplay of the martyrdom of Julietta, a Lyconian, who was tortured to death, during which dreadful operation her conflict was augmented, by being compelled to ſee the brains of her child daſhed out before her face.

A maſterly piece, repreſenting the execution of the Lollards, (as proteſtants in England were called in the reign of Henry IV.) who were hung on gibbets, and burnt by fires kindled under them, in St. Giles's in the Fields, London.

A lively repreſentation of the murder of ſeveral proteſtants, who, in endeavouring to eſcape from a barn, where they were aſſembled at their devotions, were ſhot by the ſoldiers belonging to the duke of Guiſe.

Hermanigildus, prince of the Goths, hewn to pieces by order of his father, for refuſing to change his religion.

A ſtriking and explanatory repreſentation of the diabolical tortures uſed in the inquiſition, particularly racking, with the head downwards, tormenting in the trough, into which the victim is ſcrewed, and water poured

the joints are all diſlocated—Burning the ſoles of the feet, &c. &c.

An affecting diſplay of the martyrdom of four Dutch clergymen, who were firſt ſtrangled, and then burnt at the Hague.

The ſingular death of a bookſeller, who was committed to the flames at Avignon, with bibles, which he had ſold in the French tongue, tied round his neck.

The cruel and deteſtable murder of the unfortunate Calas, who was broke on the wheel by order of the parliament of Thoulouſe.

The martyrdom of the biſhops Ridley and Latimer, who were burnt together at Oxford; and of archbiſhop Cranmer, who was ſoon after executed in the ſame manner, and at the ſame place.

A very curious repreſentation of Lollards Tower, with the manner in which the proteſtants were there cloſely confined and tortured.

A ſtriking view of the martyrdom of thirteen proteſtants, including two women, who were burnt at Stratford, near London.

The burning of archdeacon Philpot in Smithfield, with his remarkable behaviour at the place of execution.

A ſtriking repreſentation of the bloody Iriſh maſſacre in 1641, when 40,000 proteſtants fell victims to the malice of Roman catholic miſcreants.

To prevent Miſtakes, pray be careful to aſk for Dr. SOUTHWELL'S NEW BOOK OF MARTYRS.

Figure 1.2: (Color) Prints exhibiting complex layouts (such as newspapers and modern magazines) and/or paper/ink/compression degradations pose special challenges for automated document image analysis. Right-side photo source: Chris, the book swede

The work at hand represents a direct extension of the master thesis [153], concluded at the end of 2006. Whereas the master thesis provided a proof-of-concept design and implementation of a newspaper image understanding system, the current work significantly extends the range of processable documents. All newly-introduced methods have the objective of an immediate incorporation and applicability as part of the Fraunhofer Document Image Understanding system in the processing of large scale collections of digitized documents.

The overarching goal of the work at hand is to take the next step towards a fully automated document understanding system suited for mass digitization of structured and semi-structured documents. Throughout the current thesis we shall put a special emphasis on the following objectives supporting the overall goal:

- Provide a holistic overview of the document image understanding area, including up-to-date surveys of the state of the art.

- Minimize or eliminate the necessary human interaction, both during the system training time and its operation on the target document sets. Accomplishing this requires algorithms capable of handling a wide variety of documents while keeping the number of necessary parameters to a minimum.

- The developed algorithms must be able to combine and make use of all available information (e.g. color, human expertise) and strive to provide some kind of feedback about the confidence or accuracy with which they have accomplished their task.

- The computational complexity of all newly-developed algorithms must allow the efficient mass processing of digitized material.

We note that the complete removal of the human factor from the processing chain is both undesirable and unfeasible on a short-to-medium term. While in most cases human interaction represents a bottleneck for a mass digitization project, it ultimately comes up to human observers to assess the quality of the output and provide valuable feedback to the system. Furthermore, in some cases advanced metadata which may be necessary for higher-level processing stages can only be generated by humans. As such, it is important that the automated analysis systems provide algorithms and interfaces allowing for a (lightweight) human involvement. On a different note, it is interesting to observe that although computing power is still growing, its associated costs (time and money) are slowly but surely approaching a plateau. When that point is reached, it is only through the reduction of computational complexity that any further gains can be made. A further factor making computational efficiency important is the continual increase in sensor spatial and color resolution which is directly reflected in the growing size of digitized materials.

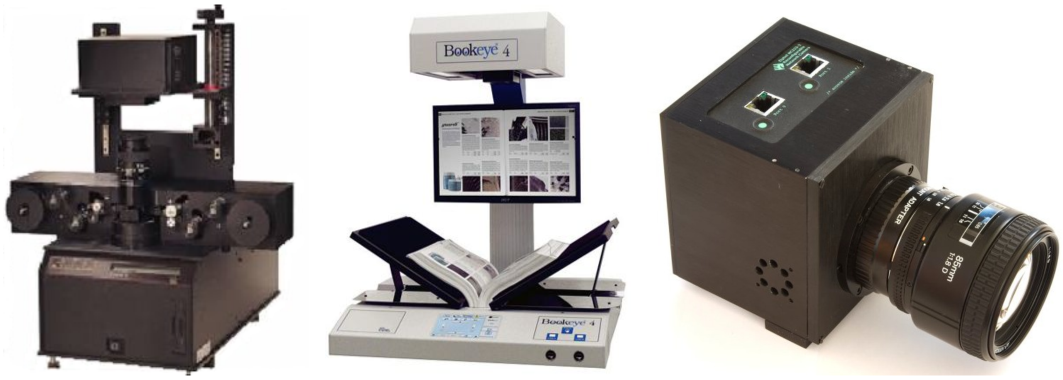


Figure 1.3: Examples of popular digitization solutions: left side – microfilm scanner (source: Document Imaging Brokers), center – book scanner (source: MCS Computer Services), right side – camera system (source: Elphel Inc.). Each type of digitization solution introduces its own set of typical artifacts

1.2 Document Image Understanding – A Bird's-eye View

Electronic documents have many advantages over paper documents, including efficient retrieval, search, copying and fast transmission. Consequently, since the early 1980s there has been extensive research on the problem of converting paper-based documents into fully-featured electronic ones. Automated document image understanding is still nowadays the topic of many scientific papers. Within the scientific community, it is generally accepted that *document image understanding* (DIU) is the process that transforms all informative content of a digital document image into an electronic format outlining its logical content [66]. In contrast, document image analysis (DIA) is restricted to the processing stages before the higher-level logical analysis. One must note that oftentimes the two terms are freely interchanged and generally refer to the entire document processing chain. The starting point of the process is a digital document image, such as a newspaper- or magazine page. Over time, the means for document digitization have continuously evolved and nowadays many different digitization solutions are in use, such as fully automatic scan robots, microfilm scanners, manual book scanners and camera systems.

In their initial form, digitized documents are typically stored on digital storage mediums in compressed raster image formats. Although the compression ratios achieved by lossy algorithms are much higher that those of their lossless counterparts, the trend in digitization is clearly favoring simple, lossless formats. This preference is not just fuelled by the ability of current DIU systems to obtain better results due to the additional information, but much more by the inherent need to make the digital archives *future-proof.* The lower compression ratios allow for a much slower digital decay time with respect to inevitable hardware failures. Moreover, the straightforward, well-known lossless coding algorithms (such as run-length coding, LZW) are more likely to be supported by software products for a much longer time. In the publishing industry and professional photography area TIFF (Tagged Image File Format) [42], followed by JPEG [213] are the most widespread formats for storing (document) images. The versatility of TIFF, as well as its ability to store near-unlimited metadata is highly valuable in the context of large, heterogeneous document archives. Note that TIFF actually represents a container format and is theoretically able to store sequences of images with differing color depths and compression schemes (e.g. tile- or strip-based, uncompressed, lossless, lossy – including JPEG).

The result produced by a document understanding system, given an input document image, is a complete representation of the document's logical structure, ranging from semantically high-level components to the lowest level components. For example, in case of text, the high level components of a document are layout columns and paragraphs, whereas the low level components are lines, words and letters. The logical comprehension of arbitrary documents is a challenging task, which is currently still out of reach for automated systems. However, insular solutions restricted to narrow sub-domains, such as form processing or address recognition are starting to be applied in the industry.

1.2.1 Workflow of a Generic DIU System

We have seen in the previous section that the end goal of a document understanding system is to produce a complete, semantically layered logical model of a given digital document image. In order to accomplish this, the DIU system is usually divided into several modules with a more restricted, but well-defined behavior. This section will provide the reader with a bird's eye view of each module, while more detailed information about each subject can

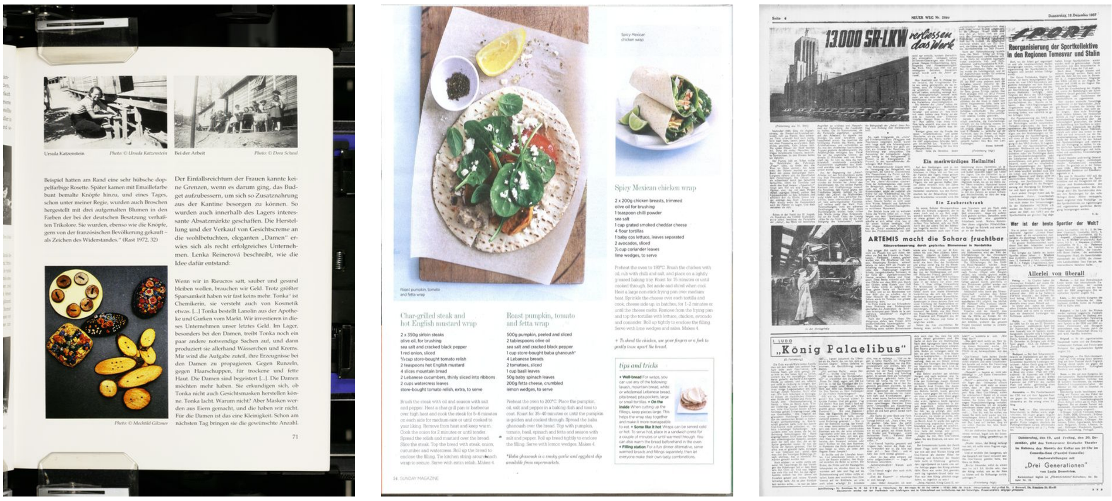


Figure 1.4: Sample scans from professional scan providers. Left to right: book, magazine and newspaper images

be found in the dedicated chapters. Before delving into the overview, it is important to note that in real systems (incl. the Fraunhofer DIU system mentioned in section 1.1) some of the modules described below may be absent or appear in a slightly different order. For professional scans of high-quality printed material for example, some of the page enhancement modules may be entirely skipped.

The first step in a generic DIU system is *quality assessment (QA)*. One must note that quality assessment in general is an process which can be applied to each processing step as an additional safeguard (see [167]). At this early stage, the purpose of the QA module is to ensure that the input document has a high enough visual quality that allows the DIU system to reach a set of pre-specified goals. For (semi-)professional scans a good way of ensuring near-optimal visual quality of the digitized material is by making use of color targets, whereas in amateur scenarios, there exist many uninformed quality measures that have been proposed as features for the sanity check.

After one has made sure that the document is of sufficient quality to warrant the DIU processing, it is customary to *split digital image into single document pages*. Having a single document page per digital image is helpful for ensuring the optimal performance for the subsequent processing steps, as in most situations they implicitly or explicitly assume this fact. The situation in which an acquired digital image contains more than one page is very common in case of books or journals, when the (automatic) scanning process digitizes two pages or one and a half at once. In such cases, one must devise a page splitting algorithm, which recognizes such cases, splits the input image accordingly and outputs the individual page images. To the best of the author's knowledge, there exist no general-purpose page splitting algorithms, all solutions relying on apriori knowledge about a certain restricted application domain. A positive side-effect of this fact is that such highly-tuned algorithms are able to achieve very accurate splitting results. Our proprietary Fraunhofer DIU system also makes use of apriori knowledge, more exactly it assumes the existence of a visible book fold and/or page edges, as specified by a human operator or by an algorithm parameter. By making certain assumptions about the position of such salient features, the directions of the edges and of the book fold can readily be determined via a solid separator detection algorithm. Page splitting and a preliminary per-page deskewing can thus be both

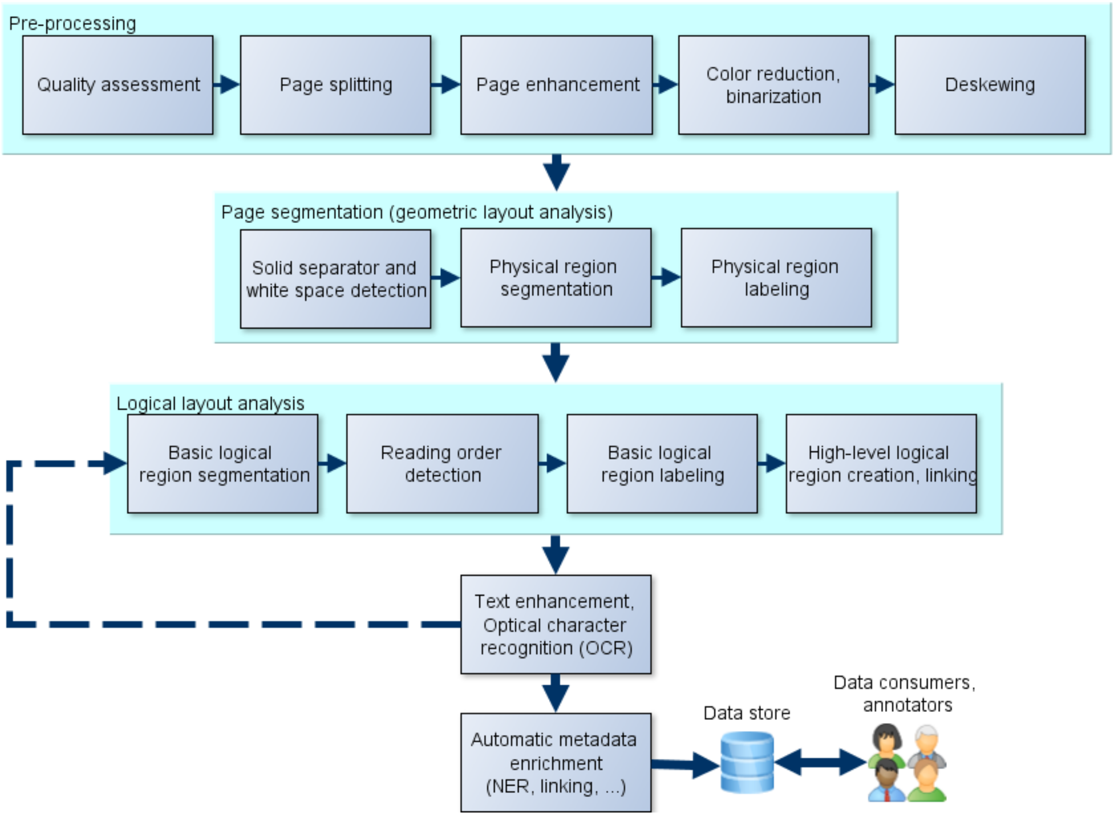

Pre-processing
Quality assessment
Page splitting
Page enhancement
Color reduction, binarization
Deskewing
Page segmentation (geometric layout analysis)
Solid separator and white space detection
Physical region segmentation
Physical region labeling
Logical layout analysis
Basic logical region segmentation
Reading order detection
Basic logical region labeling
High-level logical region creation, linking
Text enhancement, Optical character recognition (OCR)
Automatic metadata enrichment (NER, linking, ...)
Data store
Data consumers, annotators

Figure 1.5: Block diagram of a complete, generic document understanding system. Some modules may be absent or appear in a slightly different order, depending on the target area

accomplished at this point. Because of the mostly heuristic nature of the page splitting procedure and its restricted applicability domain, the exact process will not be detailed any further in the current work. An overview of separating line detection algorithms may be found in section 5.1.1 as a sub-part of page segmentation, and a detailed description of one such algorithm can be consulted in section 5.2.1.

The next step in our generic DIU system is *image improvement*. It is important to note that we shall treat the term "enhancement" exclusively as synonym for the optimization of the results produced by the automatic DIU process, which may or may not also be consistent with the visual assessment of human observers. Document image enhancement is necessary because most document images nowadays are obtained by optical scanning of paper documents, a process which invariably introduces different kinds of unwanted effects or artifacts in the resulting digital image. Usually, these unwanted effects include blurring, poor or greatly varying contrast over the surface of the image, or different kinds of noise patterns (including prominent paper texture and unrelated artifacts near the page margins – e.g. portions of fingers, scanning robot clamps). Even if the scanning process would produce perfect digital replica of the input document pages, the paper documents themselves oftentimes contain signs of paper/ink degradation, due to the printing process itself (e.g. 18^{th}–19^{th} century material), the effects of time and/or imperfect storage. Chapter 2 will present the most effective methods for improving the quality of digital document images in view of further automatic processing.

Another pre-processing module is responsible for the *color quantization* or *thresholding* of

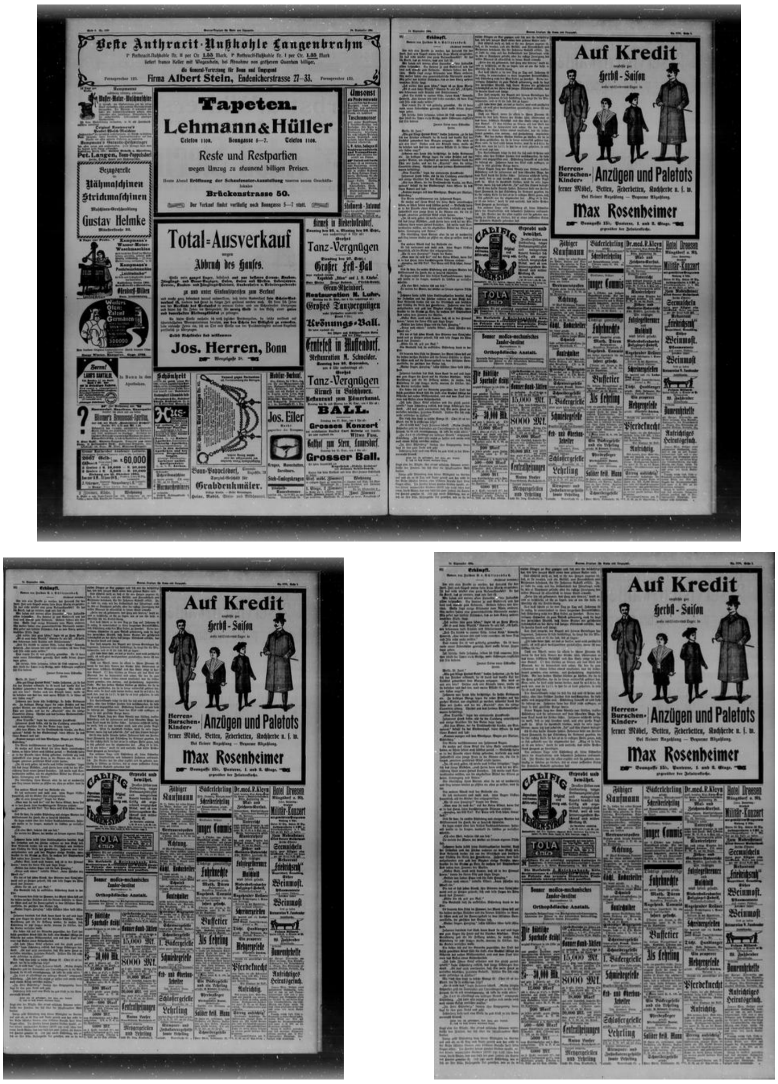


Figure 1.6: Top: Original scan showing two adjacent newspaper pages; Bottom-left: automatic page splitting result for the right-side page; Bottom-right: Border removal and book fold-based deskewing result on the same newspaper page

the document image, in other words the conversion of a color or grayscale image into a paletted- or a binary one, respectively. The current industry trend is leaning towards full-color document scans, in order to capture all potentially relevant information. While the intent is laudable, this greatly increases the burden of DIU systems which now have to deal with significantly larger amounts of data. Simply put, the linear growth in the dimensionality of the search space (for the information of interest) makes the search for a solution exponentially(!) more difficult (a.k.a. the well-known curse of dimensionality). Thus, color reduction/binarization plays a key role in document processing since its success/failure in reducing the dimensionality and quantity of input data is paramount to the performance of all subsequent processing steps [116]. Especially when dealing with degraded document images, color reduction/binarization represents a challenging task. Document color reduction and binarization methods are the subject of chapter 3.2.

Concluding the pre-processing stage is the document *skew detection and correction.* Like document image enhancement, its necessity stems from the fact that most document images have been digitized in a nearly arbitrary manner by a human operator. In fact, a recent study on a sample of around 1 000 000 pages done by a major German scan service provider [9] has revealed that scanning errors caused by humans are more than twice as frequent as errors coming from all other causes taken together(!). Thus, the digital image of a document page may have been rotated or skewed at an arbitrary angle on the platen or, in fewer cases, it was intentionally printed at a different angle to draw attention. For reading/ presentation purposes, a skew of as little as 0.1 degrees is already apparent to a human observer [134]. More importantly, although humans can cope with document images having relatively high skew angles, the majority of the current page segmentation algorithms cannot do so. An overview of several classes of skew detection algorithms is available in chapter 4.

Page segmentation represents the basis on which all high-level "understanding" modules operate. Its task is to segment and label all physical layout structures present in a document image. Physical layout structures are usually categorized into text, graphics, pictures, tables, horizontal and vertical rulers. Note that, in some publications the class of physical layout structures may be either more restricted or wider, depending on the target area. As noted by Cattoni et al. [66], the entire document understanding process up to and including this step is sometimes referred to in the specialized literature as *physical or geometric layout analysis*. Chapter 5 will review in detail the state-of-the-art in page segmentation and introduce new algorithmic solutions for enabling true mass processing for digital document archives.

As hinted in the previous paragraph, *logical layout analysis (LLA)* directly follows page segmentation. Logical layout analysis has the task of refining the already identified page regions, assigning them logical labels, as well as determining their logical sequence (i.e. reading order) and hierarchical structure. In the context of periodicals, this process is usually referred to as *article segmentation*. In the case of non-periodical structured publications LLA is mostly focused on building fine-grained tables of contents, including "hyperlinks" to the chapters, sections and figures. In comparison to page segmentation, relatively few scientific papers deal with logical layout analysis. One possible reason for this situation is the fact that logical layouts vary greatly depending on the document type, publisher and year of apparition of the respective document. It follows that solving the task of logical layout analysis in a more general fashion is an extremely challenging endeavor. An overview of the available techniques for performing logical layout analysis makes the subject of chapter 6, along with new practical solutions and a theoretical framework focused on improving the adaptability of LLA algorithms in the case of large heterogeneous document collections.

The next step of a generic document image understanding system is the recognition of the individual characters from each text region, a process widely known as OCR (optical character recognition). The main difficulties in this stage are the differentiation between fonts (OFR, handled in papers such as [166]) and obtaining reliable recognition results in the presence of noise or broken characters. This work will entirely skip the description of this step, as many commercial products exist on the market offering a reasonably good OCR performance [1, 14].

After the OCR stage has been completed, the final results must be saved in a format which would allow the DIU system to efficiently respond to a diversity of queries regarding the acquired content. Document representation issues and semantic linking problems do not make the subject of the current work, as they are both large research areas even on their own. The storage format and the type of allowable queries depend heavily on the specific application domain and utilization scenarios. However, the current trend goes towards

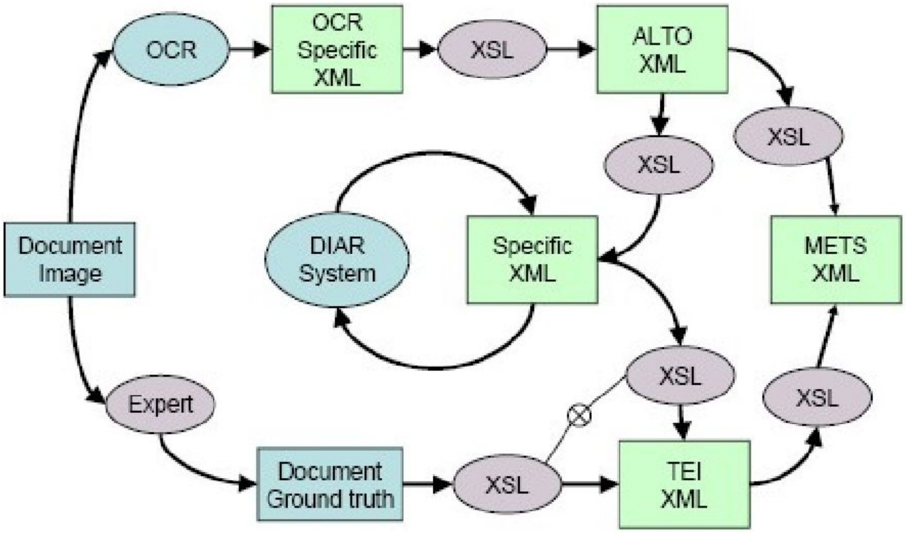


Figure 1.7: Generic, XML-based data representation scheme for document image analysis. Source: [55]

XML-based description schemas specializing on different logical levels. The recent work of Belaïd et al. [55] describes a framework for an efficient and flexible combination of several widespread formats: ALTO (Analyzed Layout and Text Objects) for the physical structures, TEI (Text Encoding Initiative) for encoding the logical regions and METS (Metadata Encoding and Transmission Standard) for describing the complex mapping between physical and logical structures. The PAGE XML format [216] additionally permits the representation of the results produced by each individual stage of a DIU system (e.g. including information about any existing geometric distortions, applied image enhancements). At his point, the automatically extracted data can be corrected and enriched with additional semantic information by human operators or external automated algorithms. A prominent example in the context of the semantic Web is the recognition (named entity recognition – NER) and disambiguation (e.g. using Wikipedia [222]) of entities and of the relations existing between them. The recognized entities and relationships allow the interlinking of digitized material, facilitating complex search queries and allowing human browsing of entire collections in an efficient manner.

1.2.2 Commercial and Open-Source Document Analysis Software

In this section we offer an overview of the most prominent automated solutions for document image analysis currently available to the general public. Note that the vast majority of today's document image analysis software started out as simple optical character recognition (OCR) products.

Tesseract [15], currently at version 3.01, is a well-known cross-platform open source OCR system. It was initially developed at Hewlett-Packard up until 1995, then development was discontinued until 2005, when it was released as open source. Ever since, it has been sponsored by Google and has seen continuos improvements. The OCR module supporting a multitude of scripts and languages is described in detail by Smith [249]. More recently, a fully-fledged page segmentation module was incorporated in order to be able to process more complex layouts. The geometric layout analysis is based on tab-stop detection [250] and a comparison with other current layout analysis modules (including the Fraunhofer

DIU system) is available in section 5.3.2.2.

Another prominent open source project sponsored by Google is *OCRopus* [13], with version 0.5 recently released in June 2012. The project was initially released in 2007 and has seen a gradual transition to a pluggable architecture (comprising mostly Python modules). Up until version 0.4 OCRopus internally used as OCR module the Tesseract engine, afterward also providing its own trainable character recognizer. While OCRopus also incorporates a few document pre-processing modules, its main current focus is on enhancing the OCR performance and architecture. The geometric layout analysis module of OCRopus is also part of the comparison in section 5.3.2.2.

The three main commercial providers of OCR products are ABBYY, Nuance (formerly ScanSoft) and I.R.I.S.. Their respective software products are marketed as all-round solutions for a complete document analysis workflow. All products include:

- OCR for over a hundred languages and different scripts
- pre-processing algorithms for the improvement of scanned document images
- page-level layout analysis capabilities to reproduce digitally the layout of the original printed material
- workflow or automation tools for corporate environments

In addition, there are some product specific features: *ABBYY FineReader* [1], currently at version 11, highlights its ability to consider multi-page documents as a whole. This way, formatting information remains consistent across the document. Logical elements like headlines and footnotes are recognized, capturing some of the logical document structure. A unique selling point of the product is the inclusion of FineReader XIX, a recognition engine capable of recognizing text in old German script, or Fraktur. This engine was developed in the concept of the EU project METAe, which ran from 2000 until 2003. While a few other products can also recognize Fraktur text (most notably Tesseract), the recognition quality of the ABBYY engine is generally superior (e.g. see sections 2.2.2.3 and 2.2.3.3). One may find a comparison between ABBYY FineReader and other state-of-the-art layout analysis engines in section 5.3.2.2. *OmniPage Professional* [14], currently at version 18, includes a page extraction algorithm to deal with double page scans, as well as a specialized form recognition engine. *IRISDocument Server* [11], currently at version 9, is marketed specifically as a solution for high-volume document analysis tasks, promising high-speed OCR conversion.

It is interesting to note that the latest trend and most developments in both ABBYY FineReader and Nuance Omnipage are related to logical layout analysis. In their newest versions, both products claim as main improvements the ability to recognize the structure of tables as well as identify large titles, headers and footers. In addition, FineReader can (partially) handle footnotes, recognize page numbers and detect the layout structure of documents. Interestingly, Omnipage is the first commercial product to employ a machine learning approach for adapting to new document layouts. In contrast, IRISDocument Server does not incorporate any significant logical layout analysis module, but instead focuses on robust mass processing and employs a variety of pre-processing steps (smoothing, despeckling, deskewing, orientation detection).

One must note that none of the software products claims to be capable of higher-level document analysis, e.g. separation of newspaper pages into articles, table of content linking in books or extraction of semantic information like authors or categories. There exist systems that build on the aforementioned OCR engines and provide additional capabilities, such as *docWORKS/METAe Edition* by CCS GmbH [2] that was developed in a corresponding research project. docWORKS/METAe performs structural analysis as well as double page splitting and extracts some semantic information. The results can be accessed hierarchically by document structure. However, just like the OCR engines described above, docWORKS cannot segment a page into individual logical elements (e.g. articles in a newspaper, book chapters).

The other trend for document analysis software goes in the direction of cloud-based services. One chief requirement for "in the cloud" document processing is robustness, as the user base and its needs become increasingly diverse. Thus, all software products are transitioning from their initial assumption of clean or born-digital documents to more generic processing modules. Many other web-based startups offer document analysis solutions via APIs, however under the hood most of them use one of the aforementioned engines.

1.3 Outline

The current work is organized as follows: The following chapter introduces some of the most widespread distortions found in generic digital images as well as in digitized documents. Traditional image enhancement solutions are presented along with novel methods for color reference target detection, text extraction and text improvement in degraded historical documents. Chapter 3 discusses the problems faced by color reduction and thresholding algorithms in the context of digital document images and offers an overview of state-of-the-art algorithms. Subsequently, a new framework allowing constant-time optimal local binarization is proposed. In the second part of the chapter we argue for the necessity of a holistic treatment of color in document images and the need for meaningful evaluation measures for color reduction results. In chapter 4 we address the issue of document skew and review the current research status in the detection of global-, multiple- and non-uniform skew. A new method allowing the seamless skew and orientation detection is then described and tested on a large document dataset. Chapter 5 delves into the problematic of geometric layout analysis and contains a survey of research methods for separator detection, region segmentation and classification. From the viewpoint of mass digitization, we present remedies for current deficiencies in the state of the art via the practical example of the Fraunhofer DIU page segmentation module, tested extensively on real-life, heterogeneous document sets. Chapter 6 introduces logical layout analysis and its challenging task of dealing with noisy and continuously changing input data. The status quo is reviewed and practical solutions for front page detection and complete article segmentation are presented along with a potential learning approach targeted at future large scale digitization projects. Finally, chapter 7 concludes the text with an overview of the main results and mentions several promising research directions.

Chapter 2

Quality Assurance and Document Image Enhancement

We know truth for the cruel instrument it is. Beauty is infinitely preferable to truth.

– G.R.R. Martin *(The Way of Cross and Dragon)*

Digital images used as input for a document understanding system most frequently suffer from degradations of the image quality. The presence of image distortions and artifacts has a negative impact on the quality of the results obtained by all subsequent processing steps. We have seen from the generic architecture of a DIU system (figure 1.5) that document enhancement represents one of the earliest processing modules. Thus, the overall success of the system critically hinges on it. Furthermore, in the context of mass digitization fully automated enhancement methods must be employed so as to satisfy hard processing time constraints.

Along with image enhancement methods, quality assessment forms the main focus of the current chapter. While the two areas are distinct, they share a number of similarities, most notably the heavy reliance on measures for image quality. Quality assurance methods most often use these criteria as basis for acceptance/rejection in contrast to image enhancement methods which use the computed measures for the accurate adjustment of the correction parameters.

It is important to note that the term "enhancement" as used in this section relates to the performance of the automatic DIU modules, and not always also to the visual aspect for human observers. In many cases, the two points of view differ considerably – for example, a smoothing procedure meant to attenuate small noise will inevitably have a detrimental effect on the perceived quality of halftones. Another meaningful example is the existence of visible page texture – for most humans this is a positive trait, since it enhances the perceived authenticity of the material, whereas the performance of automatic algorithms would in most cases take a significant hit. It is interesting to note that there currently exists a significant body of research on the fully automated assessment of perceived aesthetics, although the research is mostly focused on halftones or hand drawn content [289]. Another important observation is that the location of the document enhancement methods can differ greatly within a DIU system. Some enhancement methods, such as generic noise filtering and page border removal are usually applied right at the start of the processing chain, whereas font enhancement can be applied at any point after page segmentation.

The starting section of the current chapter offers a brief overview of quality problems commonly found in (semi-) professional digitized material (not restricted to documents), and discusses the corresponding state-of-the-art solutions. Keeping the focus on (semi-) professional collections of digital images, we introduce a generic algorithm for the detection of color reference targets. By comparing the actual color information from the digitized image with the theoretical target, exact quality measurements are made possible. The second section deals with quality problems exclusively found in digitized documents. We review the state of the art in document image enhancement and subsequently propose two new algorithms targeted at historic documents. The first algorithm targets the extraction of text/graphic information from monochromatic old documents, while the second represents a novel approach for removing character-like artifacts from text areas. Both algorithms are shown to produce significant and generic OCR improvements on heavily-affected, real-life document test sets. We conclude the chapter with a review and a look at future work in the (document) image enhancement area.

2.1 Generic Enhancement and Quality Assessment

Image enhancement and quality assessment algorithms both use as basis for their decisions the same set of quality measures. The quality measures can be obtained via two types of algorithms: informed algorithms, which assume a certain knowledge about the digitized material and blind (uninformed) algorithms which work in a fully generic manner. While the generality of the blind algorithms is certainly a big advantage, it comes at the cost of a limited applicability for the obtained measures.

A categorization and statistical evaluation of 26 uninformed image quality measures was recently done by Avcibas et al. [45]. They use a comparatively large test set of 30 heterogeneous images, artificially subjected to different compression types and rates (JPEG [213], SPIHT [226]), blurring and contamination with Gaussian noise. Other image quality measures may be found in the recent paper of Liu et al. [168]. Once the individual distortions have been detected by making use of one or more quality measures, they can be corrected using standard algorithms, such as those introduced in the following section. According to Avcibas et al. [45], most existing objective image quality measures can be categorized into one of the following categories:

- pixel difference-based measures such as mean square distortion
- correlation-based measures, using the correlation of pixels, or that of the vector angular directions
- edge-based measures, using the displacement of edge positions or their consistency across resolution levels
- spectral distance-based measures, which employ the Fourier magnitude and/or phase spectral discrepancy on a block-by-block basis
- context-based measures, that is, penalties based on various functionals of the multidimensional context probability (e.g. Hellinger distance, generalized Matusita distance)
- human visual system-based methods, which use a partial model of the human visual system for weighting features (e.g. image browsing functions)

The testing results obtained by Avcibas et al. confirm that no single quality measure performs best in all situations and shed light on the measures best suited for specific distortion scenarios.

If the digital images come from professional photographers or digitization service providers, it is typically the case that color reference targets have been included. In this situation, the measurement of the image quality can be performed more accurately, hence the decisions taken by the quality assurance module are also improved. We shall discuss this issue in more detail as part of the proposed automatic color target detection algorithm. Finally, it must be noted that in general every module of a DIU system for which there exists a pre-specified quality standard can be enriched with a corresponding quality assurance module. In the current work we shall only deal with quality assurance regarding the document image quality. More information on the topic of quality assessment can be found in the work of Lin [167], who offers a comprehensive insight into quality assurance issues regarding high volume document digitization. Another practical example is given by Meunier [185], who proposes a fully-automated quality assurance solution for document layout analysis targeted at homogeneous collections of documents.

2.1.1 Image Quality Issues

Some of the most widespread quality problems in digital images are the result of a combination of the following: blurring, noise, poor contrast, lens distortion, color bleeding and vignetting. When dealing with (semi-) professional photographs or scans, the first three issues are by far the most prominent. These issues shall be presented in more detail in the current section. Lens distortions are typically unnoticeable, unless they are used intentionally for their aesthetic aspect. We refer the interested reader to section 4.1.2, where we discussed the closely related topic of document dewarping. Color bleeding is most commonly only encountered as a compression artifact at the higher compression ratios of standard lossy algorithms (e.g. JPEG [213]). In section 3.2 we discuss a holistic color reduction framework, which deals with this problem effectively in the context of document scans. Vignetting is often an undesired effect caused by camera settings or lens limitations resulting in the reduction of the brightness and/or saturation near the borders of the image as compared to the image center. For automated processing purposes however, its effect on the overall result quality is minimal. The most common method of reducing the vignetting effect in document images is the application of adaptive (local) color reduction/ binarization methods (e.g. as presented in section 3.1.2).

2.1.1.1 Blur Correction

Blur is probably the most easily noticeable artifact in a digital image and can affect the entire image or just parts of it. The causes of blur can be manifold: wrong focus settings, camera or environmental movements (e.g. atmospheric turbulence) or washed-out printed material (as commonly encountered in document image analysis). From the feature set investigated by Avcibas et al. [45], it was found that the spectral phase, the edge stability measure and the normalized spectral distortion in a human vision-based coordinate system performed best with respect to their response for blurred regions. Patel et al. [210] recently performed a comparison of different features and feature sets most suited to classifying entire medical images with regard to the presence of an out-of-focus blur effect. The authors found out that a combination of several features performed best on the medical image dataset.

In general, the simplest and most widespread methods for sharpening are accomplished either by applying certain high-pass filters, or via a procedure known as unsharp masking. Such methods are most suited for the removal of out-of-focus blur. Note that depending on the cause of the blur effect, there exist other well-established methods. For example, in case of camera or environmental movements, a powerful framework is given by Wiener filtering-based approaches [119]. The main purpose of sharpening is to enhance the details which have been blurred. In document analysis, the focus falls normally on sharpening the character contours. Especially color reduction algorithms greatly benefit from having as input a sharp image, in contrast to working directly on the original blurred image.

Any grayscale digital image can be seen as a 2-dimensional function $f(x, y)$, taking values from the interval $[0, 255]$. Regarding an image in this way, its *Laplacian* is defined as: $\nabla^2 f = \frac{\partial^2 f}{\partial^2 x} + \frac{\partial^2 f}{\partial^2 y}$. The implementation of the digital Laplacian as a filter mask 2.1 gives a simple, yet powerful high-pass filter usable for sharpening an image. As described by Gonzales and Woods [119], applying the Laplacian masks directly tends to produce images containing grayish lines, superimposed on a dark, featureless background. In order to recover the vanished background features, one usually adds the original image to the one obtained by applying the Laplacian operator. Note that, when using the masks with a negative center coefficient, then the image resulting after applying the Laplacian must be subtracted (not added) from the original image. Normally, these two steps are merged into a single one, by simply adding (or subtracting) the value 1 to the center point in the mask.

$$\begin{array}{cc}
\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} & \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \\
\text{a)} & \text{b)} \\
\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} & \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \\
\text{c)} & \text{d)}
\end{array}$$

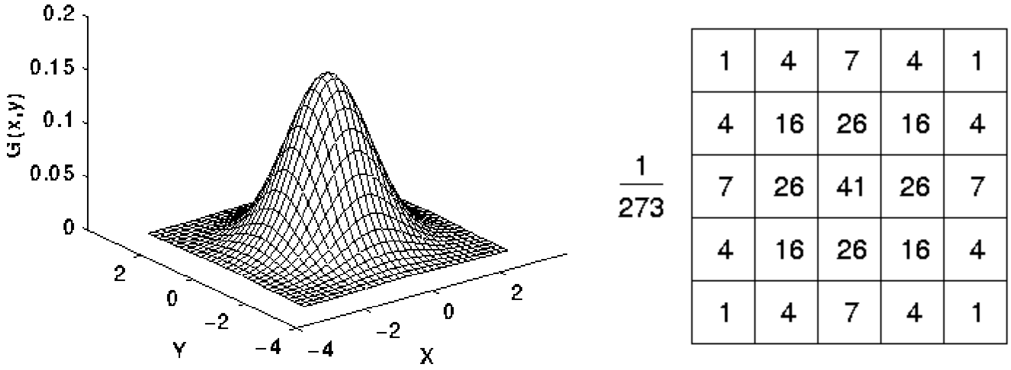
Figure 2.1: (a),(b) Direct implementations of the Laplacian as filter masks; (c),(d) Extensions of these masks, also including the diagonal neighbors

In practice, due to the fact that the application of this procedure tends to significantly amplify noise in the image, it is common to apply a certain smoothing (low-pass) filter on the input image first. In this way, the effect of small noise in the sharpening step becomes much lower, while larger “artifacts” (such as letters) can still be sharpened in a satisfying manner. Smoothing is most usually realized by applying on the image a mask representing a discrete approximation of a (scaled) 2-D *Gaussian* function, with the form:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

In theory, the Gaussian distribution is non-zero everywhere, requiring an infinitely large mask, but in practice it is nearly zero more than about three standard deviations from the mean. Truncating the mask at this point represents common practice in the image processing community. There exists one caveat [119] however: directly applying the Laplacian masks tends to produce images containing grayish lines, superimposed on a dark, featureless background. In order to recover the vanished background features, one must usually add the original image to the one processed using the Laplacian operator. Note that when using the masks with a negative center coefficient, the image resulting after applying the

Laplacian must be subtracted (not added) from the original image. Normally, these two steps are merged into a single one, by simply adding (or subtracting) the value 1 to the center point in the mask.



1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

Figure 2.2: Left: 2-D Gaussian distribution with mean $(0, 0)$ and $\sigma = 1$; Right: Corresponding 5×5 discrete approximation

In fact, since the convolution operation (application of a certain 2-D mask) is associative, we can combine (convolve) the Gaussian smoothing filter with the Laplacian filter first, and then apply the obtained hybrid mask on the image to achieve the required result. This hybrid mask is called the *Laplacian of Gaussian (LoG)* or *Mexican hat* operator. Doing things in such a way has two important advantages [106]:

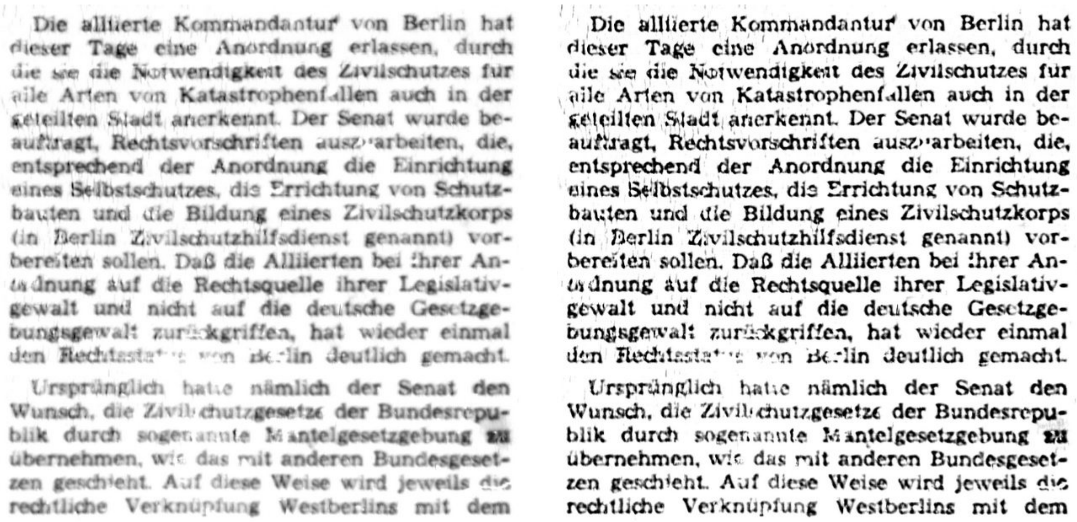
- This method usually requires far fewer arithmetic operations, because both the Gaussian and the Laplacian masks are usually much smaller than the image.
- The LoG mask can be precomputed, so that only one convolution needs to be performed at run-time on the image.

The 2-D LoG function, with mean $(0, 0)$ and standard deviation σ has the form:

$$\text{LoG}(x, y) = -\frac{1}{\pi\sigma^4}(1 - \frac{x^2 + y^2}{2\sigma^2})e^{-\frac{x^2+y^2}{2\sigma^2}}$$

The exact methodology and considerations regarding the computation of a discrete approximation to the LoG function can be consulted from the book of Klette and Zamperoni [152], section 6.2.4. It is interesting to observe that there exists neurophysiological evidence that certain human retinal ganglion cells perform in a very similar manner to the LoG operations ([119], pp. 583).

Another method for out-of-focus blur removal is *unsharp masking.* Most notably, the process has already been used for many years in the photographic and publishing industry, even before the apparition and proliferation of digital photography. It consists of subtracting an unsharp (smoothed) version of the image from the image itself, then adding the obtained result (the unsharp mask) to the original image. More formally, we can express the described process as: $f_{\text{sharp}}(x, y) = f(x, y) + k \cdot g(x, y)$, where $g(x, y) = f(x, y) - f_{\text{smooth}}(x, y)$. Here, $f_{\text{sharp}}(x, y)$ represents the final, sharpened image, $f(x, y)$ is the original image, $f_{\text{smooth}}(x, y)$ denotes a smoothed version of the original image, $g(x, y)$ is the unsharp mask and k is a constant scalar value, controlling the amount of sharpening. Normally, the smoothed version of the original image is obtained by applying a low-pass Gaussian filter (as described before), whereas the values of k commonly fall within the interval $[0.2, 0.9]$.



Die alliierte Kommandantur von Berlin hat dieser Tage eine Anordnung erlassen, durch die sie die Notwendigkeit des Zivilschutzes für alle Arten von Katastrophenfällen auch in der geteilten Stadt anerkennt. Der Senat wurde beauftragt, Rechtsvorschriften auszuarbeiten, die, entsprechend der Anordnung die Einrichtung eines Selbstschutzes, die Errichtung von Schutzbauten und die Bildung eines Zivilschutzkorps (in Berlin Zivilschutzhilfsdienst genannt) vorbereiten sollen. Daß die Alliierten bei ihrer Anordnung auf die Rechtsquelle ihrer Legislativgewalt und nicht auf die deutsche Gesetzgebungsgewalt zurückgriffen, hat wieder einmal den Rechtsstatus von Berlin deutlich gemacht.

Ursprünglich hatte nämlich der Senat den Wunsch, die Zivilschutzgesetze der Bundesrepublik durch sogenannte Mantelgesetzgebung zu übernehmen, wie das mit anderen Bundesgesetzen geschieht. Auf diese Weise wird jeweils die rechtliche Verknüpfung Westberlins mit dem

Die alliierte Kommandantur von Berlin hat dieser Tage eine Anordnung erlassen, durch die sie die Notwendigkeit des Zivilschutzes für alle Arten von Katastrophenfällen auch in der geteilten Stadt anerkennt. Der Senat wurde beauftragt, Rechtsvorschriften auszuarbeiten, die, entsprechend der Anordnung die Einrichtung eines Selbstschutzes, die Errichtung von Schutzbauten und die Bildung eines Zivilschutzkorps (in Berlin Zivilschutzhilfsdienst genannt) vorbereiten sollen. Daß die Alliierten bei ihrer Anordnung auf die Rechtsquelle ihrer Legislativgewalt und nicht auf die deutsche Gesetzgebungsgewalt zurückgriffen, hat wieder einmal den Rechtsstatus von Berlin deutlich gemacht.

Ursprünglich hatte nämlich der Senat den Wunsch, die Zivilschutzgesetze der Bundesrepublik durch sogenannte Mantelgesetzgebung zu übernehmen, wie das mit anderen Bundesgesetzen geschieht. Auf diese Weise wird jeweils die rechtliche Verknüpfung Westberlins mit dem

Figure 2.3: Left: Original, degraded part of a newspaper image (624 × 629); right: result after applying unsharp masking (smoothing mask with a radius of 13 pixels, $k = 0.75$)

2.1.1.2 Noise Reduction

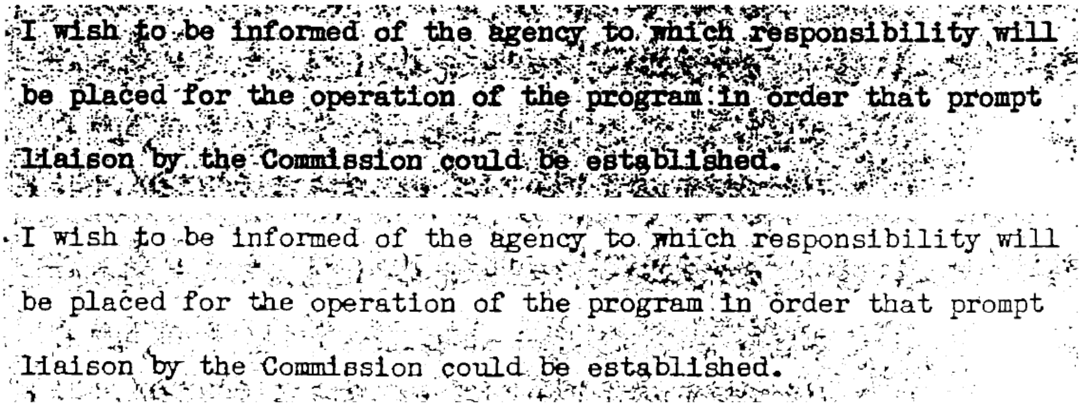
Noise appears in many forms in digital images. In image processing it is most commonly modeled as an additive process described by a certain probability distribution. Pixel- and intensity-independent *Gaussian noise* represents the most widely encountered noise type, usually caused by heat-related sensor issues. Considering the special class of document image scans, *salt-and-pepper noise* (also called *impulse noise*) is the most prevalent noise type [147, 205]. Impulse noise appears as isolated pixels or small pixel regions of black pixels on white background, as white holes within black regions or as rough edges of character or graphic components. Other less prominent types of noise are: *speckle* patterns, *shot noise* (following a Poisson distribution) and *uniform noise* (following a uniform distribution and most commonly encountered as an artifact of color quantization). Avcibas et al. [45] have identified the mean square/ absolute error as as fundamental metrics for noise.

The design objectives for a *noise reduction filter* is that it removes as much of the noise as possible, while retaining the quality of the original signal. As seen in the previous section, a simple and effective method for reducing noise is the application of a *smoothing filter* on the input image (usually followed by or already convolved with a sharpening filter). This method is most effective in case of Gaussian noise and can seamlessly be used on color, grayscale and binary images. The main drawback of straightforward Gaussian smoothing is that edges are also removed ar attenuated in the image. A solution to this problem are the so-called *bilateral filters* [264], which were developed specifically to exhibit both noise-reducing and edge-preserving properties. They accomplish this task by not only weighting pixel values with Gaussian coefficients based on their Euclidean distance from the center, but by also taking radiometric differences (i.e. color differences) into account for the weighting.

Note that for all filtering techniques, one must be aware of the *color bleeding* problem introduced by per-channel filtering. By applying a filter separately on each channel one may produce colors which were not originally present in the image. This happens because virtually all color spaces are not perceptually uniform, thus interpolation produces unexpected effects. Furthermore, the interpolation effects are highly dependent on the color space. The color bleeding effect is most visible at areas featuring sharp color transitions and which after the application of the filter will contain pixels containing “random” colors. Color bleeding can be mitigated or even completely eliminated by employing the more computationally-expensive vector filtering.

For salt-and-pepper-, as well as small speckle noise, *median filters* are commonly the method of choice. They belong to the wider class of order-statistics filters. Order-statistics filters work by ordering (ranking) the pixels contained in the image area covered by the mask and replacing the center pixel value by the value determined by the particular ranking algorithm considered. In case of median filters, the value of an image pixel is substituted by the median of the set of values of the pixels contained within the mask centered at the respective pixel. For example, when using a 3×3 mask the median is the 5^{th} largest value. Note that this definition is usable in general for images containing an arbitrary number of color values, as well as on vectors, the only pre-requisite being the possibility to compute a median value. In case of binary images, the definition becomes much simpler, allowing for efficient implementation techniques to be applied (see the integral image trick described in section 3.1.2).

Larger noise grains are typically removed via *morphological filtering*. Morphological filtering is a powerful tool which can generally be used so as to facilitate the search for/ the removal of objects of interest. This is done by smoothing out object outlines, filling small holes, thinning, detecting object boundaries, and using other similar techniques. All aforementioned operations are realized by applying the standard operators of mathematical morphology. The basic morphological operations are dilation and erosion. *Dilation* has the effect of expanding objects, while *erosion* shrinks objects by etching away (eroding) their boundaries. As most morphological operators, they can be customized by the selection of the structuring element (a binary mask), which determines exactly how the objects will be dilated or eroded. Both erosion and dilation are performed by laying the structuring element on the image and sliding it across the image in a manner similar to convolution, the only difference being the operation performed for computing the value to assign to the image pixel covered by the *focus point* from the structuring element. For erosion, if the current (focus) pixel belongs to the foreground and at least one of the image pixels covered by non-zero structuring element components belongs to the background, then it is switched to background; in all other cases the current pixel remains unchanged. In a similar manner, for dilation if the current (focus) pixel belongs to the background and at least one of the image pixels covered by non-zero structuring element components belongs to the foreground, then it is switched to foreground; in all other cases the current pixel remains unchanged. Thus, by applying erosion with a certain structuring element, objects (or noise) in the image smaller than the respective structuring element can be readily removed from the image. Similarly, dilation allows joining of adjacent objects and filling of small holes within image objects. The main problem with erosion is that it not only removes small noise, but also tends to break thin lines (such as in printed characters), thus producing the unwanted effect that a single logical object (such as a character) consists of several connected components. Likewise, dilation does not only join connected components belonging to the same logical object, but may generate connected components spanning several logical objects. In order to alleviate these problems, the *opening* and *closing* morphological operations are most commonly used. Opening consists of an erosion operation followed by a dilation with the same structuring element, whereas closing consists of a dilation followed by an erosion, again using the same structuring element. Opening has the effect of smoothing the contours of objects, breaking narrow isthmuses and eliminating thin protrusions. Closing also smoothes the object contours, but unlike opening, it usually merges narrow breaks and thin gulfs and eliminates small holes in the objects. Many more standard morphological operators exist, as well as extensions working directly on grayscale and color images and the interested reader is advised to refer to the book of Gonzales and Woods [119] for more details.



I wish to be informed of the agency to which responsibility will
be placed for the operation of the program in order that prompt
liaison by the Commission could be established.

I wish to be informed of the agency to which responsibility will
be placed for the operation of the program in order that prompt
liaison by the Commission could be established.

Figure 2.4: Top: portion of an original document showing background speckle and fattened stroke widths; Bottom: same portion of the document after enhancement using the morphological method of Loce and Dougherty [171]. Source: [64]

Finally, in cases where additional information is available on the specifics of the given digital images, more *specific filters* can be designed to take advantage of them. For example, frequency-domain filters (e.g. Fourier domain) [119] or joint frequency-spatial domain filters (e.g. wavelets) [173] can be tailored to match the noise characteristics exactly while retaining all other information from the original image. More advanced generic noise filters usable in multi-channel images can be found in section 3.2.1.

2.1.1.3 Contrast Enhancement

Low contrast is another common degradation effect found in digitized documents. Such images can be caused by poor illumination, lack of dynamic range in the imaging sensor or even wrong setting of the lens aperture during image acquisition [119]. The identification of digital images suffering from poor contrast can be performed easiest automatically or visually by examining the (per-channel) image histogram.

For dealing with the problem of low contrast, a variety of solutions have been proposed in the specialized literature. Techniques for image contrast enhancement can be classified into global and adaptive. Both classes of contrast enhancement methods have the same goal: to increase the dynamic range of the color channels in the image and increase local contrast. More specifically, the number of discernible gray levels in (a channel of) an image is determined by the *contrast* (the ratio of the difference in the perceived luminance of an object and its immediate surroundings) and the *dynamic range* (difference between the luminance values of the lightest and the darkest pixels) of the image. The subjective quality of an image is judged to be better by a human, when both the contrast and the dynamic range of the image are high [65].

Two of the most widespread *global methods* are power-law transformations and histogram equalization. *Power-law transformations* have the general form: $s = cr^{\gamma}$, where c and γ are positive constants, and s and r represent the output gray level and the input gray level, respectively. In general, a curve of such shape has the effect of spreading/compressing the gray levels present in an image. The well-known process of *gamma correction*, silently applied by most CRT (cathode ray tube) monitors nowadays, uses such a transformation (normally with $c = 1$ and $\gamma = 0.4$). Unfortunately, the values for the two required parameters must in general be determined by hand, property which makes this method less suitable for fully automatic processing. In contrast, *histogram equalization* is a parameter-free method for contrast stretching. The gray level transfer function T for a standard grayscale image with 256 gray levels is given by $s = T(r) = \lceil 255 \sum_{j=0}^{r} \frac{n_j}{n} \rceil$, where n represents the total number of pixels in the image and n_j represents the number of pixels in the image which have the value j. This transfer function has the tendency of spreading the histogram of the input image so that the gray levels of the resulting image will span a wider range of the gray scale and their distribution probability will be more uniform. In general, global contrast enhancement techniques work very well if the contrast in the input image is relatively constant (see 2.5), however they fail to produce good results for images with large spatial variations in contrast.

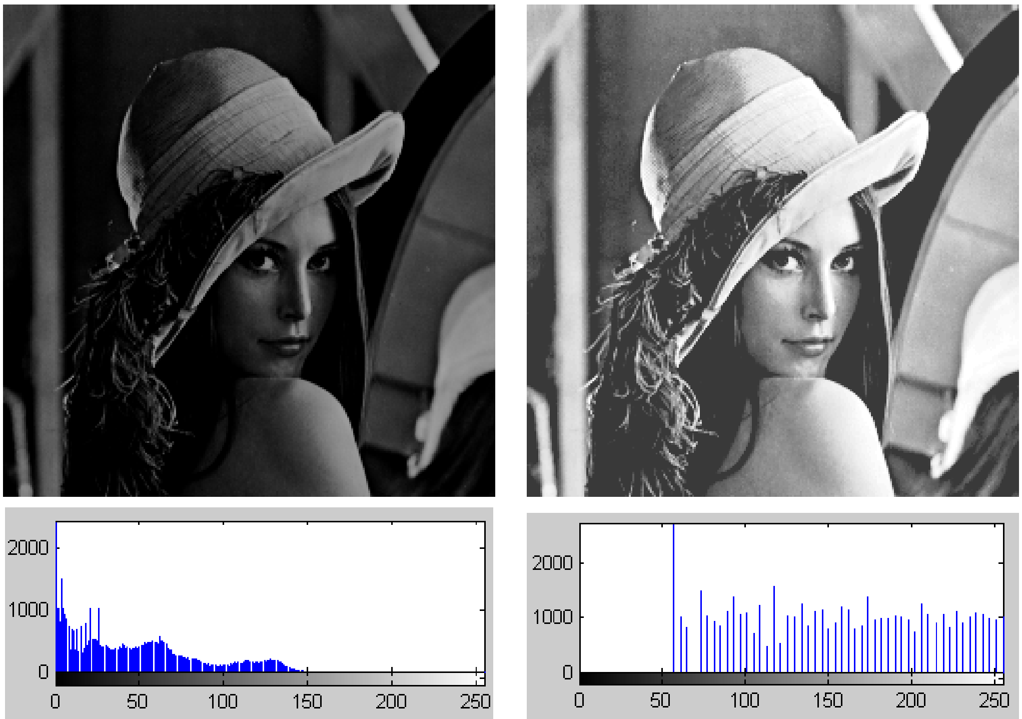


Figure 2.5: Left: Original image with poor, but relatively constant contrast and its histogram; right: result of applying histogram equalization

In order to address this issue, many *adaptive contrast enhancement* methods have been proposed. Most of them explicitly perform image segmentation either in the spatial (multi-scale) or frequency (multi-resolution) domain followed by a contrast enhancement operation on each segment. The approaches differ mainly in the way they choose to generate the multi-scale or multi-resolution image representation (anisotropic diffusion, non-linear pyramidal techniques, multi-scale morphological techniques, multi-resolution splines, mountain clustering, retinex theory) or in the way they enhance contrast after segmentation (morphological operators, wavelet transformations, curvelet transformations, k-sigma clipping, fuzzy logic, genetic algorithms) [255]. In general, adaptive methods are very computationally-intensive and, to the best of the author's knowledge, no rigorous image quality comparisons exist for evaluating the produced results. Short overviews and comparisons, as well as the description of two such algorithms may be found in [65, 255].

2.1.2 Color Target-based Quality Assessment

The employment of reference targets in the process of professional digitization or photography is ubiquitous today. Practically all current digitization guidelines aimed at the preservation of cultural heritage material highly recommend the inclusion of reference targets in each of the originals being scanned. Many agencies go even further, advocating the use of several targets in order to allow for a more accurate quantization of the variables involved in the digitization process. A prominent example are the numerous federal agencies from the United States adhering to the Federal Agencies Digitization Guidelines Initiative (FADGI) [120]. The FADGI suggests as a minimal requirement the use of photographic gray scale as a tone and color reference, as well as the use of an accurate dimensional scale. Color reference targets, also known as color checkers are therefore of central importance in any mass digitization process.

Depending on their type, reference targets allow a precise measurement of many different parameters influencing the digitization. Examples of such parameters are: the scale, rotation and any distortions present in the digitized asset, as well as the color and illumination deviation/uniformity. In the following we briefly present a few of the most popular color reference targets in use today:

- The classic color checker[3], initially commercialized starting from 1976 as the ”Macbeth” color checker [182]. It contains 24 uniformly-sized and -colored patches printed on a 8.5” × 11.5” cardboard. The colors are chosen so that they represent many natural, frequently occuring colors such as human skin, foliage, and blue sky. Nowadays it is still the most common tool employed for color comparison due to its small size and ease of use (cf. fig. 2.6(a)).
- The digital color checker SG [4] contains an extended color palette in the form of 140 quadratic patches. It is tailored to offer a greater accuracy and consistency over a wide variety of skin tones, as well as the provision of more gray scale steps ensuring a finer control of the camera balance and the ability to maintain a neutral aspect regardless of light source (cf. fig. 2.6(b)).
- The Universal Test Target (UTT) [290] is one of the most recent open-source efforts for the development of a single reference target covering a large array of scanning parameters. The development of the UTT is an ongoing process directed by the National Library of the Netherlands as part of Metamorfoze[12], the Dutch national program for the preservation of paper heritage. It is available with various options in the DIN sizes A3 to A0 and has as main purpose a general applicability in all kinds of digitization projects, preservation and access, carried out by libraries, archives and museums (cf. fig. 2.6(c)).

By using the information extracted with the help of the reference targets, a human operator is able to correct any inaccuracies of the scanning procedure on-the-fly. In case of a fully automated digitization process, a computer algorithm has the possibility of performing accurate corrections on the digitized assets for a better reproduction of the original item. Very little research has been done in the area of automatic color reference target detection. To the best of the author’s knowledge there currently exist no fully automatic solutions to this problem. A step in this direction was recently done by Tajbakhsh and Grigat [261], who introduced a semi-automatic method for color target detection and color extraction. Their method focuses on images exhibiting a significant degree of distortion (e.g. as caused by perspective, mechanical processes, camera lens). In the process of mass document digitization however, such pronounced distortions are extremely seldom mainly due to the

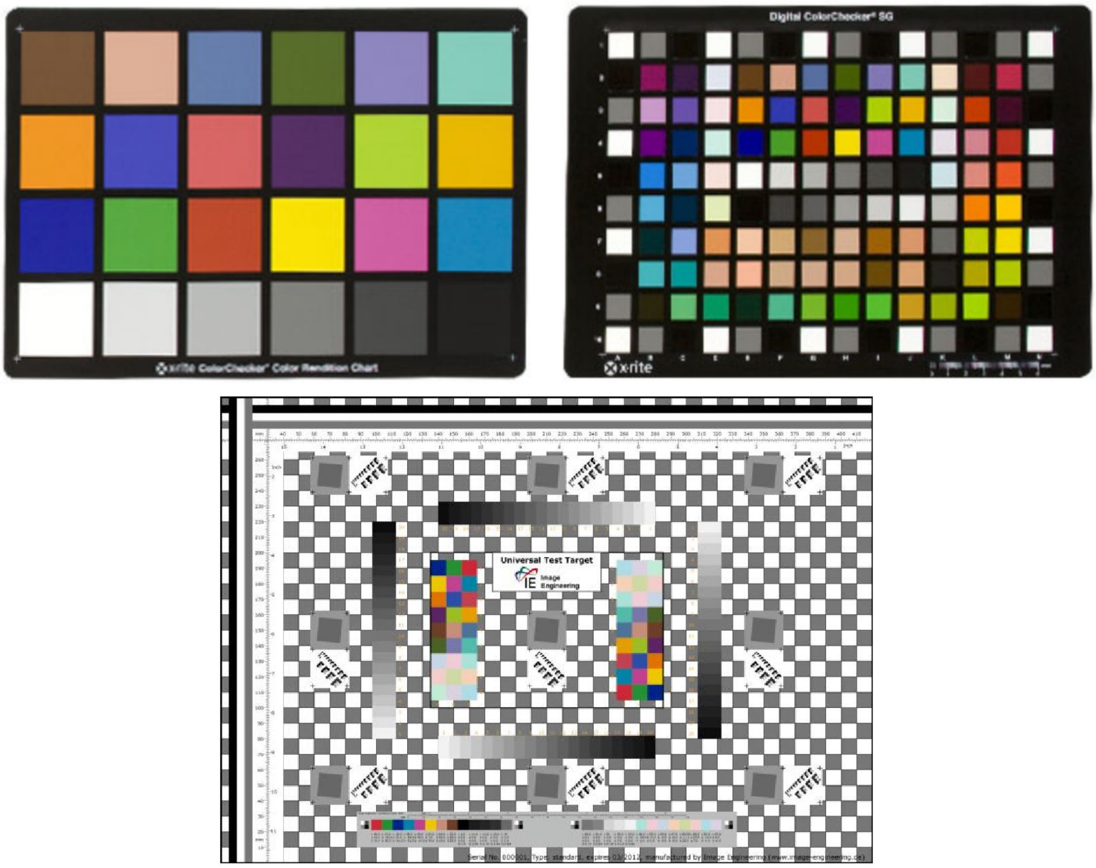


Figure 2.6: Top-left: Classic color checker; top-right: digital color checker SG; bottom: basic Universal Test Target in DIN A3 format

cooperation with professional scan service providers. A commercial software which is also capable of a semi-automatic color target detection is the X-Rite color checker Passport Camera Calibration Software [5]. The X-Rite software ultimately relies on the human operator to manually mark/correct the detected reference target in order to be able to perform any subsequent color correction. Human intervention is of course not practical in any mass digitization process, as it would cause far too large disruptions in the process.

In the following section, we present a fully automatic and robust algorithm for the detection of classic color checker targets in digital document images. The newly-proposed algorithm is an instance of an informed quality assessment, as opposed to the blind methods we have seen so far. The detected color information can be used together with the pre-specified target data to produce precise quality measurements, which in turn enable more accurate decisions about the suitability of the given digital material. Our main focus is its applicability in mass document processing, where robustness and flexibility are of paramount importance. Thus, the introduced technique can be readily extended to other types of color reference targets, including the digital color checker SG as well as the UTT (see fig. 2.6). An evaluation on a set of 239 real-life document and photograph scans is performed so as to investigate the robustness of our algorithm.

2.1.2.1 Color Target Detection

Our algorithm targets professional scans, as normally found in any mass digitization project. As such, in order to ensure a fully automatic and robust operation, we make a few assumptions. The first assumption is that the scans exhibit no or low perspective distortions. In this respect, the method of Tajbakhsh and Grigat [261] complements well the proposed algorithm in case or larger distortions. The second assumption is that the scanning resolution is known (exactly or approximately) for each scanned image, which is virtually always true in case of professional scans. A last but very important requirement is that the lighting is approximately constant on the whole image. Note that the last restriction is not specific to our system, but it applies to all methods employing color targets. In case of uneven lighting conditions (e.g. shadows, multiple light sources possibly having different spectral distributions), it is generally not possible to obtain a meaningful automatic color difference measurement/ adjustment without apriori knowledge of the lighting conditions. One may easily see this by considering the following exemplary basic situations: uneven lighting solely on the color target or uneven lighting restricted to the scanned object. In the former case, an automated color evaluation would match the (possibly correct) colors from the scanned object to the (partially) wrong ones from the color target, thus reporting large color differences and performing wrong color corrections. The latter case would result in no correction being applied to the object (possibly exhibiting large color shifts), due to the perfect lighting in the region of the color target.

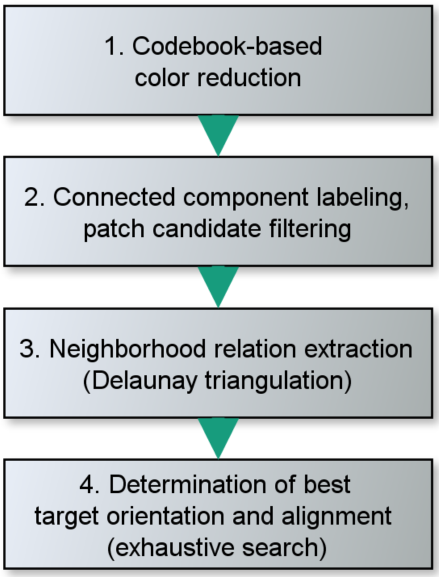

1. Codebook-based color reduction
2. Connected component labeling, patch candidate filtering
3. Neighborhood relation extraction (Delaunay triangulation)
4. Determination of best target orientation and alignment (exhaustive search)

Figure 2.7: Block diagram of the proposed algorithm for color reference target detection

Our algorithm consists of the four main steps presented in detail below, followed by the automatic color quality assessment described in section 2.1.2.3. The first step is the application of a codebook-based color reduction. More specifically, the color C_p of each pixel p is replaced with the value of the nearest codebook color:

$$C_p \leftarrow Codebook_{\operatorname*{argmax}_i D(C_p, Codebook_i)}, \tag{2.1}$$

where $D(C_1, C_2) = \sqrt{(r_1 - r_2)^2 + (g_1 - g_2)^2 + (b_1 - b_2)^2}$. The simple Euclidean distance in the RGB color space has performed well enough in our tests, however, in order to obtain more perceptually accurate color reduction results, one may use any more advanced color measure, such as CIEDE2000 [244]. The codebook consists of the set of colors existing on the color target. Note that all color components for each patch are precisely known, being specified by the reference target manufacturer as both *sRGB* and *CIE* $L^*a^*b^*$ triplets [236]. In case of the classic color checker this step results in a color-reduced image having exactly 24 colors.

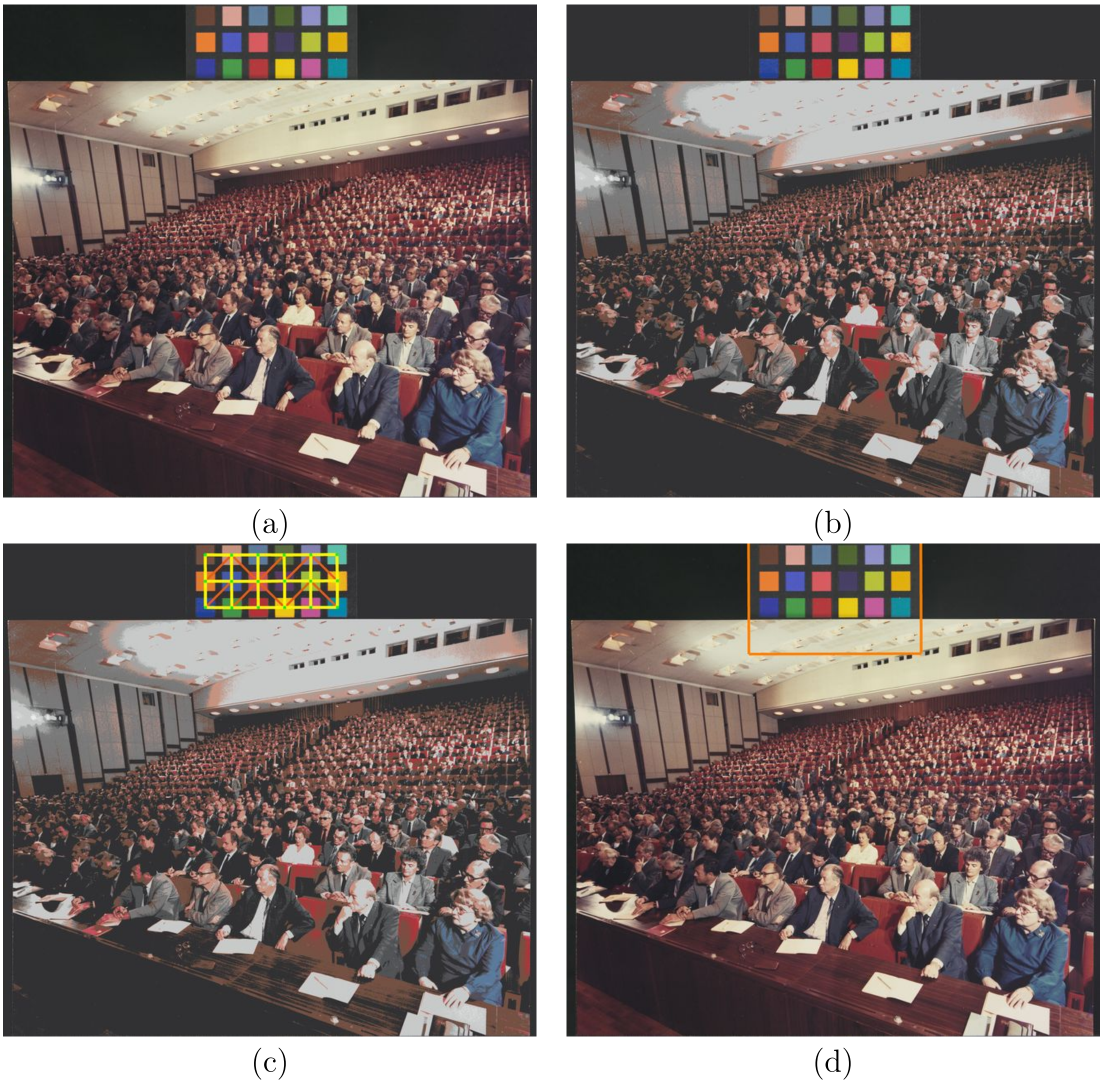


Figure 2.8: (a) Original photograph scan with a completely occluded last color checker line (no grayscale patches visible); (b) after color quantization; (c) with superimposed Delaunay triangulation (orange-painted edges were discarded from the adjacency list used for matching); (d) correctly detected color target

In the next step, a connected component analysis [90] is performed on the color-reduced image. In practice, connected component analysis is extremely fast even for large, high-quality scans because the complexity of the algorithm is constant in the number of pixels in the image. Subsequently we make use of the known scanning resolution to perform a filtering of the potential patch candidates based on their size, namely we discard all connected components having a width or height deviating more than 20% off the expected patch size. Since the shapes as well as the average distances between the color patches on the reference target are also known in advance for each color target model, they are used next as a refinement to the initial filtering. For the classic color checker our algorithm uses the following restrictions:

- the (roughly) square aspect of each color patch, i.e. width and height, are within 20% of each other
- the size uniformity between the patches, i.e. area of each bounding box, deviates less than 30% from the median area
- the average distance between direct horizontally or vertically neighboring patches, i.e. distance to closest patch candidate must lie within 20% of the median minimum distance

All previously mentioned thresholds have been experimentally determined via an independent training image set consisting of a random sample of images with a similar provenience as the images from the test set. The thresholds allow our algorithm to successfully cope with minor perspective distortions, image blur, as well as lens distortions (e.g. chromatic- and spherical aberrations). The remaining connected components constitute the final list of patch candidates. For each candidate we now determine its dominant color as the mean color of the pixels from the original scan located within its connected component. Since the original colors within each patch are relatively close to each other (they were assigned to the same cluster center by the color reduction), such a mean can be computed safely even in the RGB color space. It is important to observe that generally computing a simple mean is not possible because of resulting color interpolation artifacts, which are highly-dependent on the employed color space, such as color bleeding. Another possibility for assigning a single representative color to each patch would be the use of the median, computed either channel-by-channel or by considering each color as a vector.

A third step consists of the determination of all direct neighborhood relations between the final list of patch candidates. This is accomplished via a Delaunay triangulation [122] using as seed points the centers of the patch candidates. Next, the obtained triangulation is pruned of the edges diverging significantly from the horizontal or vertical, as regarded from a coordinate system given by the main axes of the color target. Discarding edges which deviate more than 20% from the median edge length represents an efficient pruning method. Finding one of the axes of the color target can at this point be readily accomplished by determining the median skew angle from the remaining edges. The other axis of the reference target is always considered to be perpendicular to the determined axis.

As a final step, we may now determine the exact orientation of the color target by employing the direct neighborhood relations extracted, as well as the dominant color for each patch candidate. For this purpose, we employ an exhaustive search over all four possible orientations (0°, 90°, 180°, 270°) of the target in order to compute the best matching one. The optimization criterion used is the minimization of the sum of the per-patch color distances under the neighborhood restrictions extracted in the previous step. Note that the search algorithm used in this step has a relatively small importance with respect to the running time in case of the classic color checker, as the size of the candidate list and the number of neighborhood relations is generally low. However, for large/complex color targets one may wish to use a more sophisticated search algorithm such as A^* [128] or one of its variants.

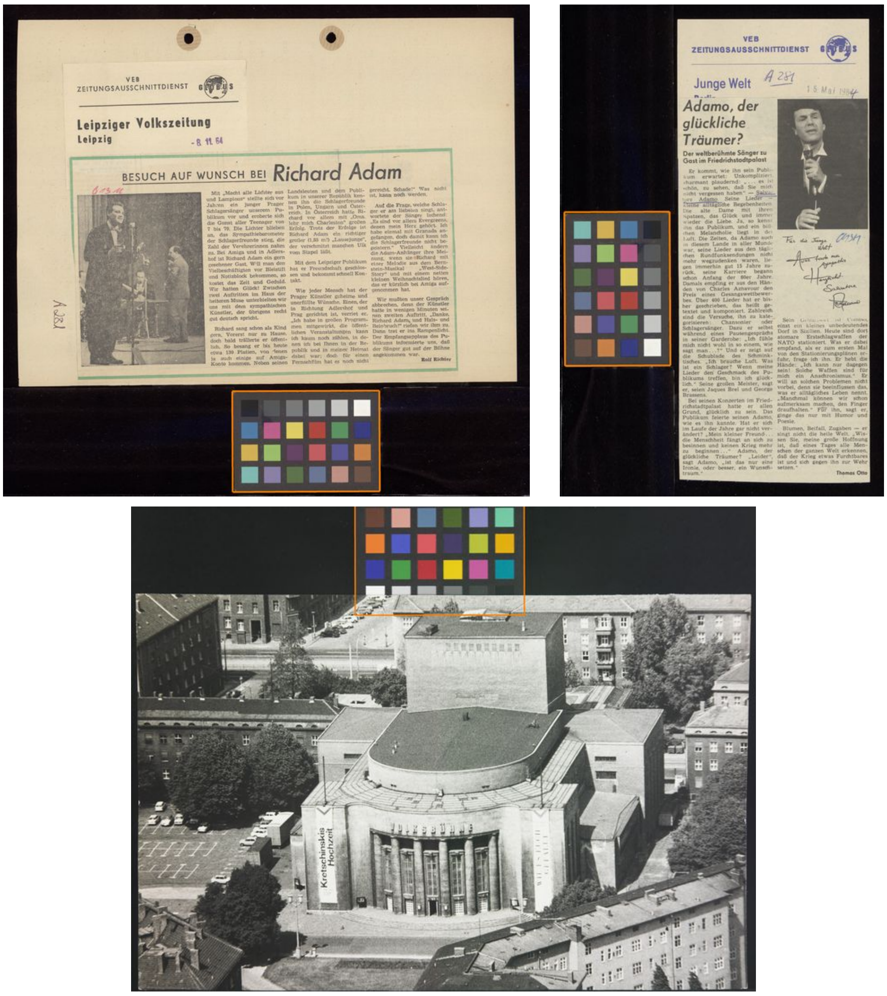

VEB ZEITUNGSAUSSCHNITTDIENST
Leipziger Volkszeitung
Leipzig
BESUCH AUF WUNSCH BEI Richard Adam
VEB ZEITUNGSAUSSCHNITTDIENST
Junge Welt
Adamo, der glückliche Träumer?
Kretschinskis Hochzeit

Figure 2.9: Examples of identified color reference targets, illustrating correct target detection for multiple orientations and robustness to partial occlusion

2.1.2.2 Experimental results

Our test set consists of 239 test images including photographs and various printed documents (newspaper excerpts, book pages) digitized by the German National Library as part of the use case Contentus in the project Theseus [17]. The test images were scanned using different resolutions ranging from 300 dpi to 600 dpi. The position of the color target varied considerably as the human scanning operator was allowed to put the color target anywhere in the vicinity of the item to digitize. Table 2.1 depicts the color target orientations in the dataset. As can be seen, our detection algorithm yields an average recall of 97.1%. We can thus conclude that the proposed algorithm is robust and has a good recall for any color target orientation.

From the analysis of the test data, we have identified two main causes for the detection failures. The majority of the cases were caused by errors in the metadata of the input scan, namely the units for the scanning resolution were incorrectly specified as being dots per centimeter instead of dots per inch. The resulting grossly different scan resolution value caused the candidate patch filtering process to fail and further prevented the correct recognition of the color reference targets. Since all scans have the same provenience and

	Color checker Orientation				**Total**
	Horizontal		**Vertical**		
	0°	**180°**	**90°**	**270°**	
Total Scans	115	13	109	2	239
Failed Detections	2	0	5	0	7
Recall	98.4%		95.5%		97.1%

Table 2.1: Color reference target detection results on a heterogeneous dataset consisting of books, newspaper excerpts and photographs

were taken in the same time interval, it seems most likely that these inaccuracies are simply glitches in the image metadata. Such errors are practically unavoidable when large amounts of data are involved. The other cause of failure was a very high or complete occlusion of the color checker. In case less than a single row of color patches is visible on the scan, our algorithm fails because of its inability to find enough initial patch candidates required for establishing a reliable orientation match. It is interesting to observe that in such extreme situations with very few reference color patches visible, the identification of a color target may not even be desirable because of the inherent inability to perform a subsequent meaningful color quality evaluation and/or correction.

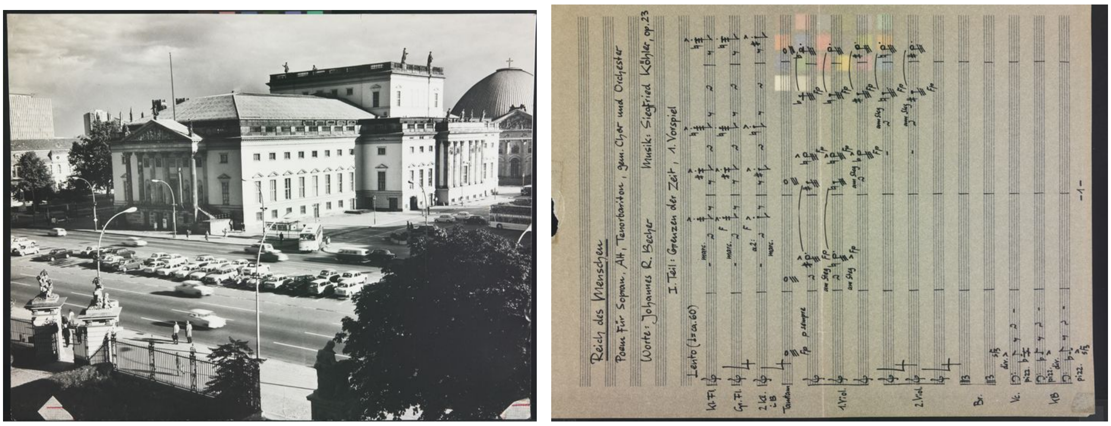

Reich des Menschen
Poem für Sopran, Alt, Tenorbariton, gem. Chor und Orchester
Worte: Johannes R. Becher
Musik: Siegfried Köhler, op. 23
I. Teil: Grenzen der Zeit, 1. Vorspiel
-1-

Figure 2.10: Examples of failed color target identifications caused by high occlusion ratios

2.1.2.3 Quality Evaluation and Color Correction

At this point we have seen how color reference targets can be automatically detected in digital images. Furthermore, since the original colors of each color patch lie relatively close to each other we were able to determine a single color value corresponding to each of the detected patches. The color value can be safely computed even in the *RGB* color space as a simple mean, because the colors within each patch lie sufficiently closed to one another. Commonly used professional scanners, either flatbed scanners or high-resolution DSLR cameras, are specifically calibrated to accurately reproduce the colors of the input physical objects into the digital image [72]. However, even after the calibration step, the color distribution of the scanned document may still significantly deviate from that of the original because of different photosensitive materials, differing refraction and reflection properties or the ambient illumination. The degree of luminance and color shift represents an important image quality measure which can be used by the quality assurance module to make an informed decision about the viability of each scanned document.

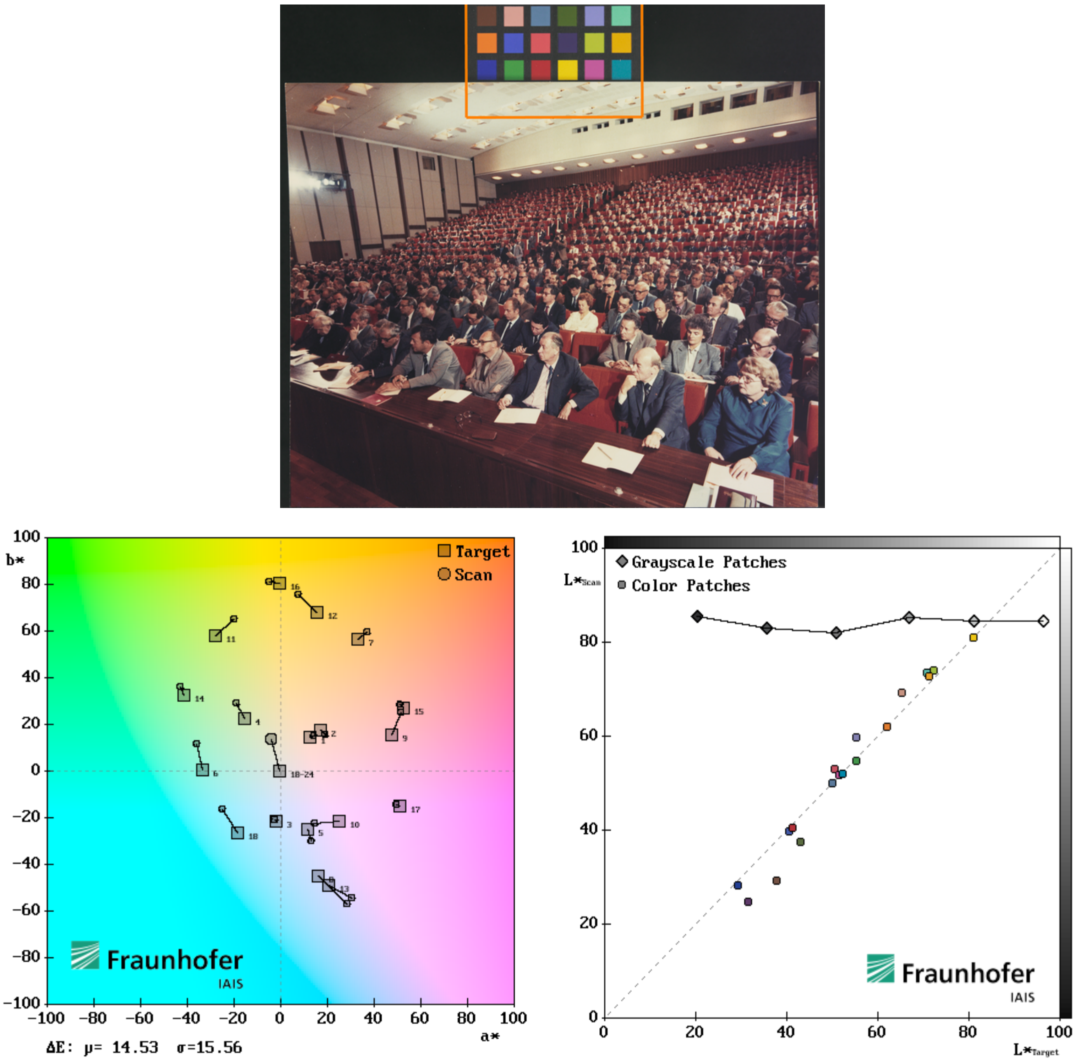
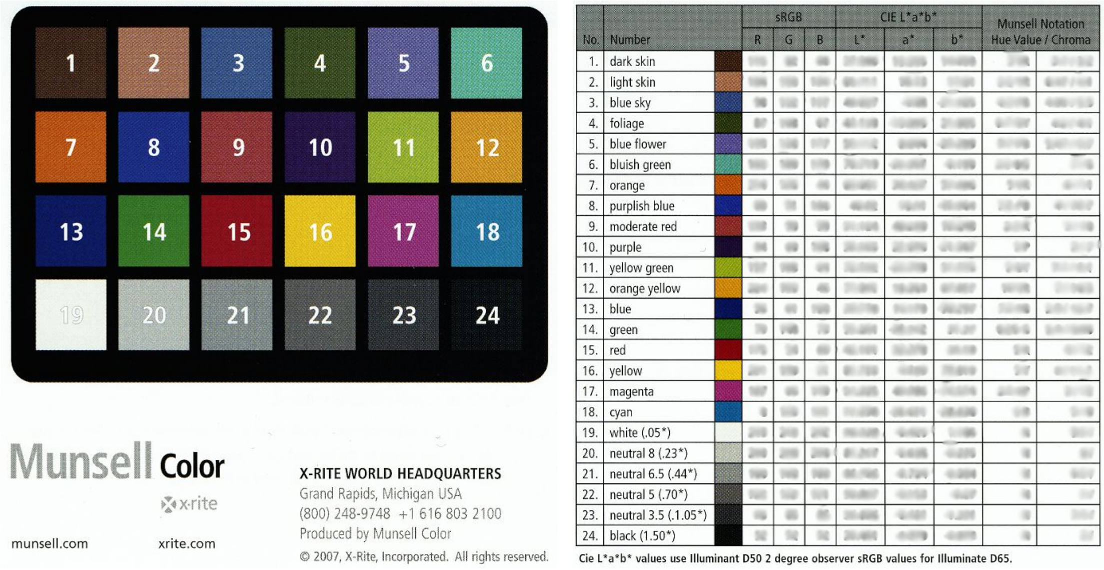


Figure 2.11: Top: Result of color checker detection on a photograph scan with a completely occluded color checker bottom line (grayscale patches not visible); Bottom: Visualizations of the automatically extracted color quality results (chrominance, luminance). Note the large luminance deviations caused by the missing grayscale patches

The Delta E value, defined by the *International Commission on Illumination* (CIE), describes the difference between the original and the scanned color checkers within the CIE L*a*b* color space. Note that all color reference targets include exact CIE L*a*b* triplets describing the reference patch colors. An additional reason for using the CIE L*a*b* color space instead of *sRGB*, is that the former color space is the one applied by default for describing most ICC (International Color Consortium) profiles of color management for scanners and printers [137]. The Delta E value between two color samples $(L^*, a^*, b^*)_1$ and $(L^*, a^*, b^*)_2$ is defined as:

$$\Delta E^*_{ab} = \sqrt{(L^*_1 - L^*_2)^2 + (a^*_1 - a^*_2)^2 + (b^*_1 - b^*_2)^2}. \tag{2.2}$$



X-RITE WORLD HEADQUARTERS
Grand Rapids, Michigan USA
(800) 248-9748 +1 616 803 2100
Produced by Munsell Color
© 2007, X-Rite, Incorporated. All rights reserved.

			sRGB			CIE L*a*b*			Munsell Notation	
No.	Number		R	G	B	L*	a*	b*	Hue Value / Chroma	
1.	dark skin	■	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2.	light skin	■	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
3.	blue sky	■	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
4.	foliage	■	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
5.	blue flower	■	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
6.	bluish green	■	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
7.	orange	■	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
8.	purplish blue	■	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
9.	moderate red	■	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
10.	purple	■	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
11.	yellow green	■	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
12.	orange yellow	■	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
13.	blue	■	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
14.	green	■	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
15.	red	■	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
16.	yellow	■	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
17.	magenta	■	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
18.	cyan	■	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
19.	white (.05*)	■	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
20.	neutral 8 (.23*)	■	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
21.	neutral 6.5 (.44*)	■	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
22.	neutral 5 (.70*)	■	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
23.	neutral 3.5 (1.05*)	■	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
24.	black (1.50*)	■	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Cie L*a*b* values use Illuminant D50 2 degree observer sRGB values for Illuminate D65.

Figure 2.12: Left: Reference X-RITE color checker; Right: table of the corresponding color values as both sRGB and CIE L*a*b* triplets, located on its backside. Note: numerical values were blurred out because of copyright reasons

Based on the CIE76 [138] criteria, if $\Delta E^*_{ab} > 2.3$, the difference is already noticeable, whereas for a $\Delta E^*_{ab} > 5.0$, we may already safely assume that the color samples belong to different colors [243]. The quality assurance module of the Fraunhofer DIU system computes the maximum value of this color metric on the set of detected patches and takes a binary decision about the suitability of the image using a fixed threshold. The threshold represents the targeted quality and varies depending on the application area. Note that the aforementioned color distance represents but the first try in a long series of attempts at providing a true perceptual distance metric. More complex formulas have been devised in order to cope with different scenarios (e.g. graphic arts vs textiles), perceptual uniformity problems around the blue hue region, different weighting factors for each color channel, etc.. While the new formulas and the color spaces for which they were defined do indeed represent a step toward the perceptual distance metric property, they are still far from actually reaching this goal. We discuss this issue in more detail in section 3.2.

2.2 Document-centric Techniques

In addition to the common degradation types found in digital images and addressed in the previous sections, digitized document images exhibit a set of specific degradations. The degradations can be caused by the acquisition source (e.g. uneven feed rates in scanning devices), by human scanning operators (e.g. visible thumbs over document content, broken document parts), by the physical degradation of the ink and paper, as well as by printing-related issues (e.g. interference patterns caused by the dithering grid produced by the original printing device). As examples of typical degradation effects found in digitized documents we mention: non-uniform foreground/background hue and intensity, blocking artifacts, partial smearing/blurring, shadows, poor contrast, specific noise patterns (Moiré, salt-and-pepper, speckle) and backside bleed-through.

It is important to note that a generic quality evaluation methodology for document images

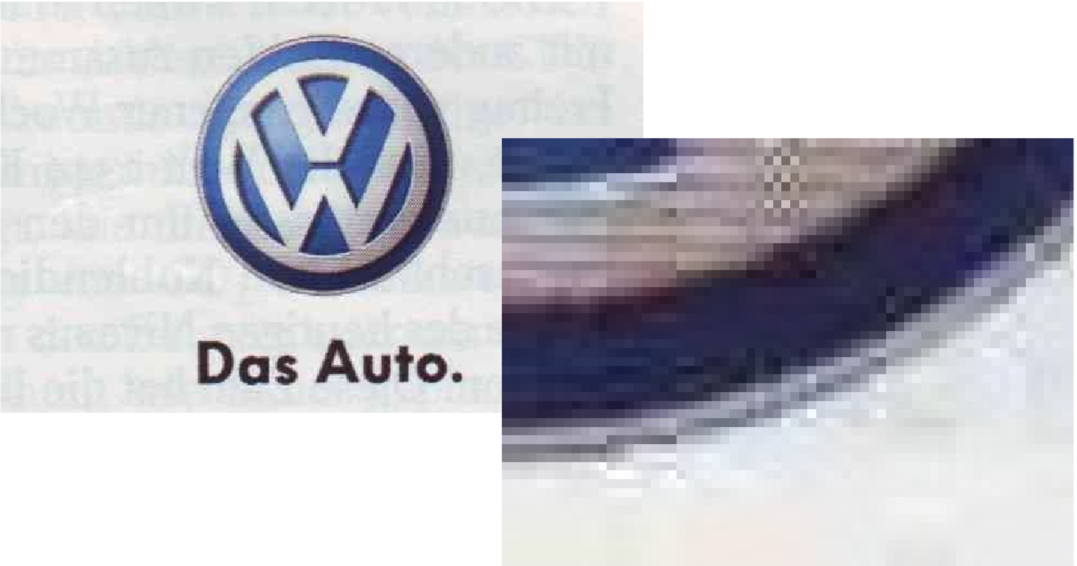


Figure 2.13: Portion of scanned document image suffering from backside bleed-through, Moirée patterns, blurring, light noise and blocking artifacts. Source: [156]

does not currently exist, nor is it likely that such a methodology will be developed in the near future. This is because most of the enhancement methods have a very specific purpose in mind, be it a direct legibility improvement, an overall improved appearance for the document, a better region segmentation performance, a.s.o.. As such, the gains brought by the image enhancement can most times only be ascertained directly by a human viewer, or indirectly by gaging the results of the modules within the DIU system (e.g. color reduction, page segmentation) using the optimized document. The most popular method of assessing enhancement gains is via the OCR results.

2.2.1 Overview

The topic of document image enhancement has seen much research ever since the early days of document image analysis (more than 25 years ago). In this section we present the most important research directions related to document enhancement and the corresponding state of the art. Since digitized documents exhibit all of the generic distortions mentioned in the first section of the chapter, the discussed correction methods typically represent a good first improvement step. Most often however, generic methods must be augmented with document-specific techniques in order to obtain an optimal quality for the enhancement results.

Margin noise- or border removal represents a relatively new area of investigation. In the past, many page segmentation algorithms (e.g. [124, 280]) relied on the implicit and hidden assumption of having as input a digital document image showing only the print area of a page. While newer methods are able to deal with margin noise more robustly, they still produce sub-optimal results in the presence of unwanted (and typically large) artifacts near the document image borders [239, 242]. Common examples of such artifacts are portions of the book cover, visible scanning robot clams or human fingers holding the physical document, dark/textured areas around the page (e.g. scanning platen, hard shadows). Agrawal and Doermann [20] propose a heuristic method for clutter/margin noise removal using features based on a residual image obtained by the analysis of the distance transform and clutter elements. They are able to obtain a 95% pixel-level accuracy for the noise removal on a dataset containing mixed degraded English and Arabic documents.

An indirect method for eliminating margin noise is the detection of the *print area (page frame)* followed by the cropping of the document image. Shafait et al [242] propose a page frame detection algorithm which uses a geometric matching method. They extract the text lines and the homogeneous document regions using well-known robust algorithms and compute the page frame as the smallest area optimizing a set of region alignment criteria (encoded as a quality function). The authors show that the error rate of several widespread page segmentation algorithms on a heterogeneous document dataset is significantly reduced when using as input document images cropped to the detected page frame. Zhou and Liu [303] of Amazon Inc. approach the page frame detection problem with a different purpose: to find suitable areas for embedding contextual advertisements. They note that the print on demand area is steadily and rapidly growing its revenue. In contrast to Shafait et al. [242], their method focuses on book material, generally featuring simpler layouts. The authors detect the layout columns of the document and compute the page frame as the smallest bounding box containing all columns. The layout columns are detected by searching for peaks in the vertical projection of the computed set of approximate text lines. The authors report better results than the Shafait et al. algorithm on the University of Washington UW-III 1600 image dataset [215].

Speckle noise is found in many document scans, especially those produced using low-quality printing methods or as a result of color reduction. A particular challenge is posed by requirement of a relatively strong noise reduction component coupled with retaining text components intact. Via a series of morphological filters most single-pixel islands, protrusions and holes can be found and eliminated (filled), as shown by Shih and Kasturi [246]. For noise patterns larger than one pixel, the kFill filter introduced by O'Gorman [205] has seen successful use. A comprehensive evaluation of the effectiveness of different filters, as well as a fully automated filter selection method was described by Cannon et al. [64] for typewritten document images.

In one of the few papers dealing with color documents, Fan [102] introduces a spatially variant framework for the correction of *uneven illumination and color cast* in digitized books. The main novelty of the algorithm comes from the additional exploitation of the coherency across multiple pages belonging to the same book. As such, the algorithm operates on the premise that each individual set of images belongs to the same book and that within a book the pages possess the same surface properties, thus the parameters describing them fluctuate in a relatively narrow range. In the first step, key algorithm parameters (light surface) are estimated through a sequential scan of a downsampled subset of the book pages (5–30 images). The second step consists of the up-scaling of the computed light map and its subtraction from all book pages. The authors mention an execution time of around 10 seconds per page but only provide visual samples as method of evaluation.

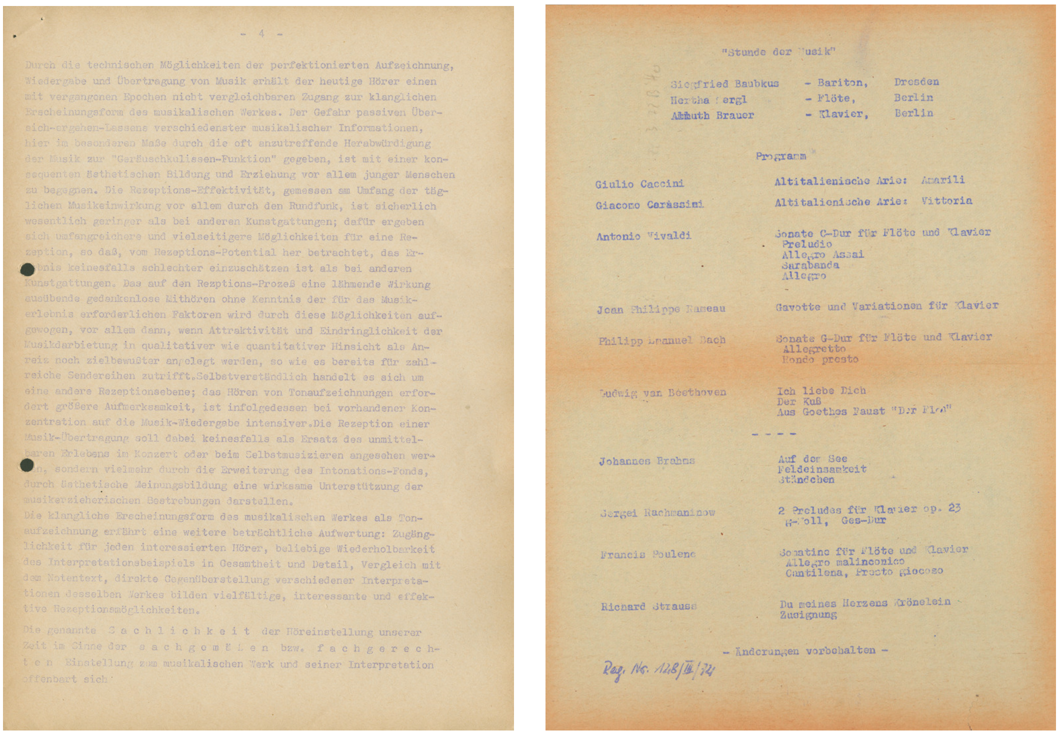
Algorithms for *character enhancement* are continuously being proposed for coping with all types of printing techniques as well as for handwritten documents. Despite the continued research effort, until now no unified algorithm applicable for all printing techniques exists. Because of the sheer variety of artifacts it is indeed doubtful that such a generic improvement method is actually possible. Shi and Govindaraju [245] address the problem of enhancing both typewritten and handwritten characters in binary document images. They reconstruct character strokes via a set of heuristic neighborhood operators for selective region growing. Experimental results on a set of 1500 address block images show an improvement of abound 7% in the OCR rate. Many other algorithms have been proposed for character enhancement in documents exhibiting various kinds of aging- or printing process-related degradation. In this category fall approaches for character restoration using various energy minimization- and stroke models [25], as well algorithms specialized for the enhancement of low-resolution fax images [132] or typewritten documents [32]. These methods restrict themselves to strictly improving the visual aspect and connectedness of individual characters either for presentation purposes or for improving OCR performance in the case of OCR engines working on a connected component basis, such as Google's Tesseract [15].

A step up in difficulty, Gatos et al. [114] address the problem of removing *stroke-like printing artifacts* from letterpress printed historical newspapers. The assumption that the printing artifacts are thinner than the strokes of the characters is used to remove the artifacts by morphological processing. Successive shrinking and swelling operations with a 5×5 window are performed. The main disadvantage of this method is that artifacts are sometimes worsened by merging the artifact with the character. Another disadvantage is that this method only works if the artifacts are thinner than the strokes of the characters. As can be observed in figure 2.21, this is not the case in many situations. Agrawal and Doermann [21] tackle the more difficult problem of stroke-like noise removal from handwritten documents. Stroke-like noise cannot be removed via traditional filters due to its striking similarity to text. Instead, the authors find the text component from the document via a set of script-independent descriptors and subsequently classify the connected components from the textual areas into text and non-text (i.e. noise components) via a RBF-kernel SVM (trained on a set of manually-labeled samples). Finally, stroke-like noise is identified via a 2-dimensional, 2-class K-means clustering [129] using the stroke width and cohesiveness features for the non-text components. Agrawal and Doermann report a precision of 85% and a recall of 90% on a set of 50 handwritten Arabic documents, subject to different deformations.

Finally, one of the best represented research areas within document enhancement is the family of *bleed-through/show-through removal* techniques for double-sided documents. Most generic by nature are the techniques belonging to the blind family, such as blind source separation [267]. These methods attempt to alleviate the bleed-through effects from the document front side without requiring any prior knowledge about its corresponding back side, which is in many cases not available. Blind source separation has the inherent limitation that the spatial distribution and/or the gray levels of the printing artifacts are distinguishable from those of the regular text. In the past a few notable attempts have been done to cope with the important problem of bleed-through (or show-through) effects [265], where the focus was on palimpsests, ancient manuscripts that have been erased and then rewritten again. Tonazzini et al.[266] introduce a Bayesian formulation for a joint blind-source separation and restoration of noisy mixtures of degraded images. The authors employ edge-preserving Markov random field (MRF) image models in order to describe local spatial auto-correlation. Their method involves a linear data model, where multimodal observations of an object are seen as mixtures of all the patterns to be extracted.

2.2.2 Processing of Hectographic Copies

Archives and cultural facilities nowadays contain vast spectra of different document classes, many of which are obsolete by current standards. A prominent document class in this context is hectography, which was a inexpensive printing and duplication method, widely used throughout the 19^{th} and 20^{th} century. The major challenge with hectography is poor contrast on the one side and multiple degradation effects on the other.



- 4 -

Durch die technischen Möglichkeiten der perfektionierten Aufzeichnung, Wiedergabe und Übertragung von Musik erhält der heutige Hörer einen mit vergangenen Epochen nicht vergleichbaren Zugang zur klanglichen Erscheinungsform des musikalischen Werkes. Der Gefahr passiven Übersich-ergehen-Lassens verschiedenster musikalischer Informationen, hier im besonderen Maße durch die oft anzutreffende Herabwürdigung der Musik zur "Geräuschkulissen-Funktion" gegeben, ist mit einer konsequenten ästhetischen Bildung und Erziehung vor allem junger Menschen zu begegnen. Die Rezeptions-Effektivität, gemessen am Umfang der täglichen Musikeinwirkung vor allem durch den Rundfunk, ist sicherlich wesentlich geringer als bei anderen Kunstgattungen; dafür ergeben sich umfangreichere und vielseitigere Möglichkeiten für eine Rezeption, so daß, vom Rezeptions-Potential her betrachtet, das Erlebnis keinesfalls schlechter einzuschätzen ist als bei anderen Kunstgattungen. Das auf den Rezptions-Prozeß eine lähmende Wirkung ausübende gedankenlose Mithören ohne Kenntnis der für das Musikerlebnis erforderlichen Faktoren wird durch diese Möglichkeiten aufgewogen, vor allem dann, wenn Attraktivität und Eindringlichkeit der Musikdarbietung in qualitativer wie quantitativer Hinsicht als Anreiz noch zielbewußter angelegt werden, so wie es bereits für zahlreiche Sendereihen zutrifft. Selbstverständlich handelt es sich um eine andere Rezeptionsebene; das Hören von Tonaufzeichnungen erfordert größere Aufmerksamkeit, ist infolgedessen bei vorhandener Konzentration auf die Musik-Wiedergabe intensiver. Die Rezeption einer Musik-Übertragung soll dabei keinesfalls als Ersatz des unmittelbaren Erlebens im Konzert oder beim Selbstmusizieren angesehen werden, sondern vielmehr durch die Erweiterung des Intonations-Fonds, durch ästhetische Meinungsbildung eine wirksame Unterstützung der musikerzieherischen Bestrebungen darstellen.

Die klangliche Erscheinungsform des musikalischen Werkes als Tonaufzeichnung erfährt eine weitere beträchtliche Aufwertung: Zugänglichkeit für jeden interessierten Hörer, beliebige Wiederholbarkeit des Interpretationsbeispiels in Gesamtheit und Detail, Vergleich mit dem Notentext, direkte Gegenüberstellung verschiedener Interpretationen desselben Werkes bilden vielfältige, interessante und effektive Rezeptionsmöglichkeiten.

Die genannte Sachlichkeit der Höreinstellung unserer Zeit im Sinne der sachgemäßen bzw. fachgerechten Einstellung zum musikalischen Werk und seiner Interpretation offenbart sich

"Stunde der Musik"

Siegfried Baubkus	- Bariton,	Dresden
Hertha Fergl	- Flöte,	Berlin
Almuth Brauer	- Klavier,	Berlin

Programm

Giulio Caccini	Altitalienische Arie: Amarili
Giacomo Caràssimi	Altitalienische Arie: Vittoria
Antonio Vivaldi	Sonate C-Dur für Flöte und Klavier Preludio Allegro Assai Sarabanda Allegro
Jean Philippe Rameau	Gavotte und Variationen für Klavier
Philipp Emanuel Bach	Sonate G-Dur für Flöte und Klavier Allegretto Rondo presto
Ludwig van Beethoven	Ich liebe Dich Der Kuß Aus Goethes Faust "Der Floh"

- - - -

Johannes Brahms	Auf dem See Feldeinsamkeit Ständchen
Sergei Rachmaninow	2 Preludes für Klavier op. 23 g-Moll, Ges-Dur
Francis Poulenc	Sonatine für Flöte und Klavier Allegro malinconico Cantilena, Presto giocoso
Richard Strauss	Du meines Herzens Krönelein Zueignung

- Änderungen vorbehalten -

Reg. Nr. 128/III/74

Figure 2.14: Typical hectography duplicates showing strong degradation effects and prominent folding. Source: German National Library

For simplifying reasons, in contrast to the rest of the current chapter we shall consider that the digitization process in itself has been perfect and did not introduce any unwanted artifacts, or all such artifacts have already been flawlessly removed. Even operating on this premise, a major problem that arises in the context of historical prints is the low-quality of the documents traceable back to two main reasons:

- the outdated printing or duplication process itself which produced only documents of low quality
- time-dependent degradation effects gradually decreasing the quality of the physical documents

As seen in the previous section, traditional document analysis approaches for noise reduction and character enhancement operate almost exclusively on binary images. Yet, the use of thresholding techniques to remove the background is often not effective since the intensities of the unwanted background can be very close to those of the foreground text. Generally speaking, thresholding and color reduction completely discard a large portion of the input data, while also introducing certain discretization artifacts (e.g. speckle noise). While most of the discretization artifacts can indeed be removed using state-of-the-art document enhancement methods, the discarded portion of the data can never be fully recovered afterward (e.g. repairing broken characters, missing parts of halftones). In conclusion, our main goal is to take into account all available image information in order to have at least a theoretical chance of retaining all relevant data. This is in contrast to traditional approaches using binarization, which invariably do not remove the complete background or also discard part of the information of interest within text areas [265].

In this section we propose a novel algorithm which allows for separation of the time-dependent degradation effects from the crucial text component. The proposed technique treats the extraction problem as a classical blind-source separation problem. Independent component analysis (ICA) forms the core of our method, as it is in theory ideally suited in our situation. The problem of unwanted Gaussian noise components is considered as well. Furthermore, straightforward suggestions for an efficient implementation are presented throughout. The proposed preprocessing method for hectography duplicates is experimentally shown to lead to an order of magnitude improvement in the optical character recognition results compared to traditional methods.

2.2.2.1 Hectography – Problem Analysis and Pitfalls

Nowadays hectography is considered to be an obsolete copying/duplication and printing method. This has not always been the case – in fact hectography was a most widespread document multiplication method in use throughout the 19^{th} century up until the third quarter of the 20^{th} century. Today hectography documents are primarily found in archives. The process of hectography uses a mastercopy consisting of paper coated with a special mixture of gelatin. The mastercopy is written or typed with a special aniline ink. Gelatin is employed because its moisture keeps the ink from drying. Duplicates of the mastercopy are then made by means of a roller which presses the blank papers onto the gelatin. With each copy some ink is removed from the gelatin and consequently successive copies are progressively lighter. In practice, between 20 and 80 duplicates can be made from one mastercopy, depending on the skill of the user and the quality of the original. In the past at least eight different colors of hectographic ink existed, but blue/violet was the most popular because of its density and contrast, as can also be seen from the scanned samples in figure 2.14.

One drawback was that even fresh duplicates obtained by hectography exhibited a low quality by current standards. Furthermore, because of the impregnation process the paper used for hectography has the tendency to become yellow over time, causing a poor contrast with regard to both colors and brightness. Strong degradation effects and folds in the used paper are common as well. Therefore without any preprocessing, performing OCR on hectography duplicates leads to very low recognition rates. Combining all above factors an eligible preprocessing method has to incorporate the following minimum goals: reliably separate the background from the text-component and enhance the contrast of the latter.

To summarize the observations so far: the hectography process involves using a single foreground color per document, the paper used is one having a special texture and due to the materials employed in the copying process, the paper will tend to degrade significantly over time. Keeping these facts in mind, it seems natural to model a hectography duplicate as a weighted superposition of several independent components, namely text/drawing content, degradation effects, paper-texture and noise.

The most obvious solution for the problem posed in this manner would be the use of principal component analysis (PCA) [94, 144]. As such, one may extract the principal components which will hopefully correspond to the aforementioned independent components. Indeed our first unsuccessful attempts at solving the hectography separation problem were done using the PCA. In the following we offer a short analysis of the causes of these failures. As it is well known the first principal component contains the most variance of the original random variables. For most documents and hectography copies in special the size of the background clearly exceeds the text area, with the result that the variance of the background noise exceeds the variance of the text/line-art component by far. Therefore the first principal component is inevitably determined by the background noise and it becomes clear that this can not in any way lead to a cleanly separated text/line-art or paper background component. More specifically, this happens because all other principal components must be orthogonal to the first one (and to each other) [93, 200].

2.2.2.2 Proposed Approach

In case the aforementioned simple additive model holds, the independent component analysis (ICA) [135, 136] is a more powerful tool for component extraction. The ICA has the great advantage of not being reliant on the orthogonality of the components and is therefore able to combine the information contained in the luminance component as well as in the color components. Since traditional image files contain three observed signals (RGB), the number of independent components to be extracted from them is limited to three as well [136]. This restriction however does not pose a major problem, since both paper-texture and degradation effects belong to related classes of phenomena, and their distinct extraction is not needed usually, i.e. it is sufficient to extract the sum of both components. In addition, the proposed approach does not consider the spatial information of the image and hence, the RGB components are treated as mere random variables. The RGB-components of each pixel in the image can therefore be seen as single observations.

Using the established ICA notation [136] the problem can be stated as follows:

$$\begin{aligned} R &= \alpha_{11} \cdot Z_{text} + \alpha_{12} \cdot Z_{degrad} + \alpha_{13} \cdot Z_{noise}, \\ G &= \alpha_{21} \cdot Z_{text} + \alpha_{22} \cdot Z_{degrad} + \alpha_{23} \cdot Z_{noise}, \\ B &= \alpha_{31} \cdot Z_{text} + \alpha_{32} \cdot Z_{degrad} + \alpha_{33} \cdot Z_{noise}, \end{aligned} \tag{2.3}$$

where R, G, B are the observed random variables and Z_{text}, Z_{degrad}, Z_{noise} are the unknown generative components. The unknown mixing matrix is denoted by α_{ij}, $i, j \in \{1, 2, 3\}$. In consistence with the restrictions of ICA we can assume that all components are non-Gaussian except for the noise-component Z_{noise}. Particularly the independent component associated with the text-content Z_{text} is non-Gaussian due to a high structure of the corresponding distribution functions. It has to be considered thoroughly how to deal with a possible noise problem. Preliminary results show that the variance of the noise-component Z_{noise} is negligible such that this component can be omitted after whitening. This additional dimension reduction has a significant impact on the time performance of the ICA [78, 136].

In the first processing step an *inverse gamma correction* is performed for all three color channels:

$$v = \begin{cases} V/12.92 & V \leq 0.04045 \\ \left((V + 0.055) / 1.055 \right)^{2.4} & V > 0.04045 \end{cases}$$

where $V \in \{R, G, B\}$. This step is obligatory since otherwise our model in (2.3) would not hold due to the performed gamma correction. From our experiments, we have observed that the impact of the inverse gamma correction on the final binary image is only marginal however.

The next processing step comprises both the *whitening* of the random variables R, G, B and the subsequent *dimensionality reduction*. Since we intend to reduce the dimensionality as well, we achieve this best by the means of the classical PCA [94, 144]. First to simplify matters it is common to subtract the means of all three random variables [136, 144]. Since it is obvious from the context, we will use the same names for the original and the mean-free variables.

As is well known from the theory of PCA [94, 144] in case the variance of one of the random variables R, G, B significantly exceeds the variances of the others, then the first principal component is almost completely determined by this random variable. In such case determining the variance declared by each principal component is distorted and misleading. In order to avoid this phenomenon we have to standardize the variance before applying PCA:

$$\begin{aligned} R &= R/\sqrt{E\{R^2\}}, \\ G &= G/\sqrt{E\{G^2\}}, \\ B &= B/\sqrt{E\{B^2\}}, \end{aligned}$$

such that all random variables R, G, B have the same impact on the principal components.

Subsequently the principal components P_1, P_2, P_3 of the random variables R, G, B have to be obtained. To this end any known method can be applied. We haven chosen the simple and straightforward method of eigenvalue decomposition of the corresponding covariance matrix of R, G, B. In hectography the first two principal components P_1, P_2 declare almost always more than 90% of the original total variance. For this reason the last component P_3 which most likely contains only Gaussian noise is removed.

To conclude the whitening process, we have to standardize the variances of the remaining components:

$$P_i = P_i/\sqrt{\lambda_i},\ i \in \{1, 2, 3\}$$

where $\lambda_1, \lambda_2, \lambda_3$ denote the eigenvalues of the covariance matrix of the scaled R, G, B channels, respectively.

In the last computation step the independent components:

$$\begin{aligned} Z_{text} &= \beta_{11} \cdot P_1 + \beta_{12} \cdot P_2 + \beta_{13} \cdot P_3, \\ Z_{degrad} &= \beta_{21} \cdot P_1 + \beta_{22} \cdot P_2 + \beta_{23} \cdot P_3, \\ Z_{noise} &= \beta_{31} \cdot P_1 + \beta_{32} \cdot P_2 + \beta_{33} \cdot P_3, \end{aligned}$$

are obtained using an iterative procedure described in [135, 136]. As proposed in [136] we apply the symmetric approach and use the tanh non-linearity. Due to the dimension reduction described above P_3 and Z_{noise} are not considered usually.

In order to decrease the number of iterations of the FastICA and hence the computation time an adequate starting value has to be chosen. For this purpose we have determined the coefficients β_{ij}, $i, j \in \{1, 2, 3\}$ of the demixing matrix for a large number of hectography images. We use the arithmetic average as starting value, which due to the similarity of hectography images is a better choice than random values. Additionally, since document scans at 300–400dpi are rather large it is recommended that all steps involved in calculation of the aforementioned demixing matrix be performed on a downsampled version of the original image. A downsampling factor of $M = 2$ or $M = 3$ depending on the original image size has performed well in our experiments. The downsampling has a significant impact on the necessary computational load.

Subsequently all preceding scalings and transformations are combined to a single transformation matrix as described in [135, 136] and applied pixel-wise on the original image.

Please note that the obtained independent components Z_{text}, Z_{degrad}, Z_{noise} are still kept as floating point numbers, only in the last step are rescaled and quantized according to:

$$y = \left\lceil \frac{255}{Y_{max} - Y_{min}} \cdot (Y - Y_{min}) \right\rceil, \tag{2.4}$$

where $Y \in \{P_1, P_2, P_3\}$.

Finally, two ambiguities or indeterminacies with respect to ICA must be dealt with: the ambiguity of the sign and the ambiguity of the order of the independent components.

The ambiguity of the sign has the effect that in some cases some of the images representing the independent components have inverted grayscales values with respect to the original image, i.e. white background in the original image appears black in the component image. Since in text documents the amount of pixels belonging to the background usually exceeds the amount of pixels belonging to the text, this effect can be compensated rather easily by first determining the background color of the original image (after the grayscale transform) and second comparing it with the background color of each component image. The previous scaling in (2.4) allows for using a simple static threshold of $Y_{thres} = 128$ for this purpose. We then count the total amount of pixels in each grayscale-image satisfying $Y < Y_{thres}$ and $Y \geq Y_{thres}$.

The other ambiguity of ICA, as mentioned above, is the fact that the output order of the independent components is arbitrary. As such, there is no guarantee that the first component always will in fact be the text-component. This brings up the question how to automatically identify this component. There are many possible ways of identifying the text component, some more robust than others. The most obvious one would be to count the number of connected components [69, 90] in all component images. Since the text component ideally contains only characters and in contrast to the background and noise components almost no specks, the number of connected components should be much lower here. To render this identification method more stable more sophisticated methods, like determining the variance of the size of the connected components, could be incorporated.

Our approach however is based on two fundamental results from information theory, *1.)* the more "random" (i.e. unpredictable and unstructured) a random variable is, the larger its (differential) entropy [136] becomes:

$$H(X) = -\int p_X(\xi) \cdot \log p_X(\xi)\, d\xi$$

and *2.)* the fact that Gaussian variables have the larges entropy among all random variables with a given covariance matrix. Both fundamental results allow to define a measure that is zero for a Gaussian variable and always non negative. This measure is known as negentropy [136]:

$$J(X) = H(X_{gauss}) - H(X) \tag{2.5}$$

Since the brightness of letters naturally strongly differs from the brightness of the background, text components will usually have a very structured probability density function and hence the negentropy (2.5) assumes a large value. On the other side, extracted background components will have less obvious inherent structure and therefore stronger resemble Gaussian variables. Consequently the negentropy will have a smaller value.

The main problem with negentropy is that it is computationally very difficult. In [136] several different approximations have therefore been proposed. We employ the following one:

$$J(X) = k_1 \cdot \left(E\left\{X \cdot \exp\left(-X^2/2\right)\right\}\right)^2 + k_2 \cdot \left(E\left\{|X|\right\} - \sqrt{2/\pi}\right)^2,$$

with $k_1 = 36/\left(8\sqrt{3} - 9\right)$ and $k_2 = 24/\left(16\sqrt{3} - 27\right)$. In practice the expectation is substituted by the sample average.

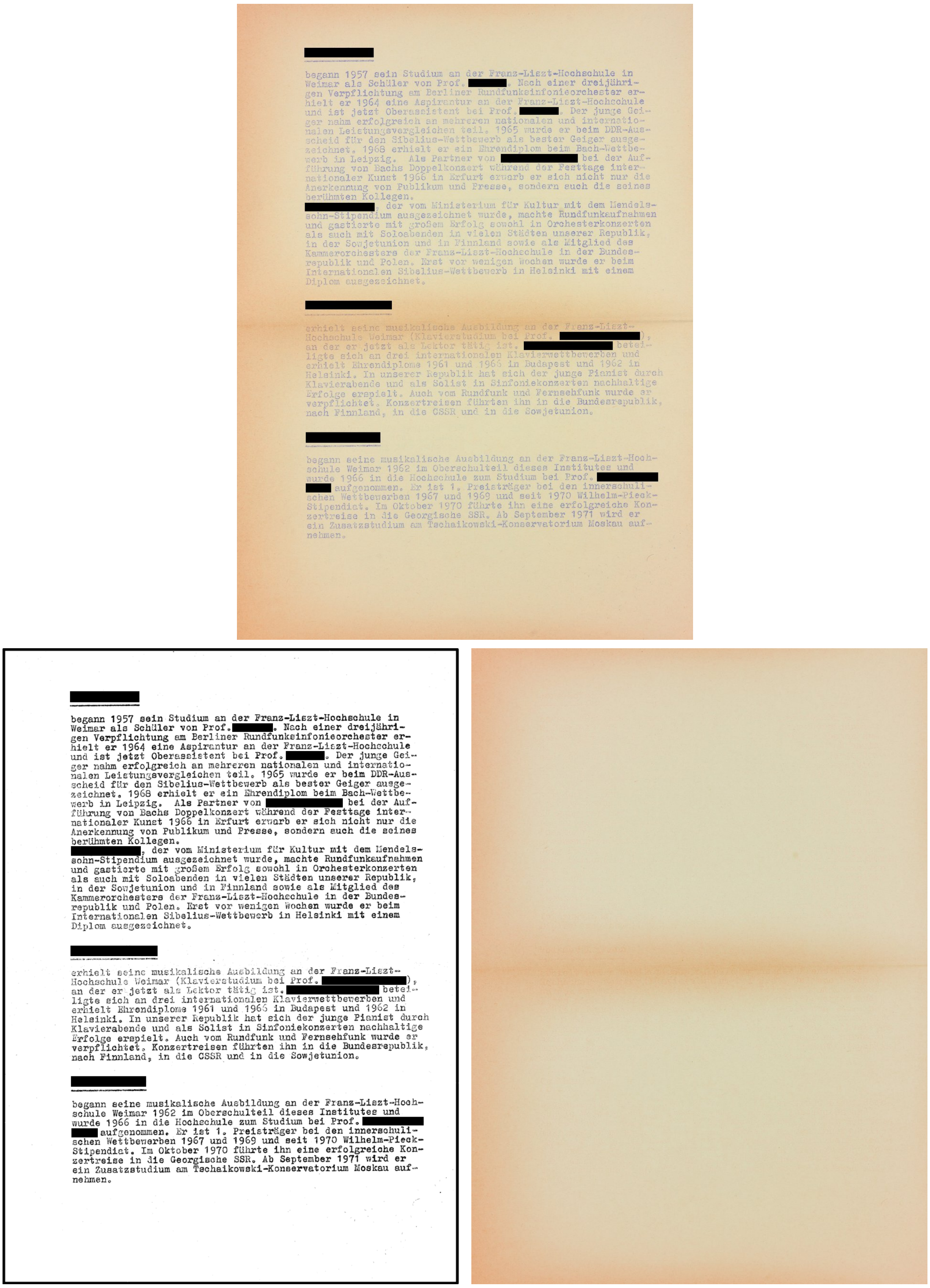
Concluding this section we go over a few practical issues using as reference the hectographic image from figure 2.15 (a). The original document was scanned with 600 dpi resulting in a large 3815 × 5319 image. The corresponding demixing matrix is obtained by performing the FastICA on a downsampled version of the image using a sampling factor of $M = 2$. This is done for computation time issues only. From our experiments on a larger set of images, we have observed that the sampling factor can be increased up to $M = 4$ without incurring a significant loss in the quality of the results. In contrast, the subsequent transformation into the independent components is performed on the full image in order to obtain the components in maximum possible resolution. The first two independent components obtained after variance scaling exhibit more than 98% of the original variance. The last independent component contains (predominantly Gaussian) noise only and is therefore discarded. The 98% threshold was experimentally determined on a disjoint set of hectography scans. The number of FastICA iterations needed was 2 and the running time for the entire process was 10 seconds – a typical value for 600 dpi DIN A4-sized hectography scans. The negentropy (2.5), (2.6) for the text component on this document image was about 10 times higher than that of the background component. A similar magnitude difference was observed throughout our test set, even for documents where the text component was rather faint to the human eye.

2.2.2.3 Evaluation

As can be seen from the zoomed-in document portion in figure 2.16, the text component is generally almost perfectly extracted. In some cases however, the ink itself is so faint that even this is not enough so as to allow for a good subsequent OCR result. Many images from our test set exhibited such areas, which explains the relatively poor OCR results of both engines in table 2.2. In such cases, classic morphological operators were found to improve OCR results by an order of magnitude (e.g. [119], chp. 9). Such target-specific methods do not fall into the scope of the current work, however.

In order to quantify the gains brought by the proposed method we employ the Levenshtein distance [162] to the ground truthed page text. Our choice of the Levenshtein distance as evaluation measure was determined by the fact that it is directly proportional to the amount of human labor necessary for obtaining a perfect text result. Perfect OCR results are in general what every library or content-owner would like to possess before making the digitized material available to the wide public.

Two independent OCR engines internally employing different recognition methods were used for validation, so as to ensure that the improvements are indeed generic. The OCR engines are Abbyy Finereader [1], as the leading commercial product and Google's Tesseract [15], arguably the best-performing free OCR engine available at present. Ground truth text templates for 22 different hectography documents were manually generated. The data set contains altogether 42 825 characters and 5833 words. In order to guarantee an unbiased comparison between the different preprocessing methods employed by the two OCR engines, the exact positions of all text lines were manually marked as well. We



[redacted]

begann 1957 sein Studium an der Franz-Liszt-Hochschule in Weimar als Schüler von Prof. [redacted]. Nach einer dreijährigen Verpflichtung am Berliner Rundfunksinfonieorchester erhielt er 1964 eine Aspirantur an der Franz-Liszt-Hochschule und ist jetzt Oberassistent bei Prof. [redacted]. Der junge Geiger nahm erfolgreich an mehreren nationalen und internationalen Leistungsvergleichen teil. 1965 wurde er beim DDR-Ausscheid für den Sibelius-Wettbewerb als bester Geiger ausgezeichnet. 1968 erhielt er ein Ehrendiplom beim Bach-Wettbewerb in Leipzig. Als Partner von [redacted] bei der Aufführung von Bachs Doppelkonzert während der Festtage internationaler Kunst 1966 in Erfurt erwarb er sich nicht nur die Anerkennung von Publikum und Presse, sondern auch die seines berühmten Kollegen.

[redacted], der vom Ministerium für Kultur mit dem Mendelssohn-Stipendium ausgezeichnet wurde, machte Rundfunkaufnahmen und gastierte mit großem Erfolg sowohl in Orchesterkonzerten als auch mit Soloabenden in vielen Städten unserer Republik, in der Sowjetunion und in Finnland sowie als Mitglied des Kammerorchesters der Franz-Liszt-Hochschule in der Bundesrepublik und Polen. Erst vor wenigen Wochen wurde er beim Internationalen Sibelius-Wettbewerb in Helsinki mit einem Diplom ausgezeichnet.

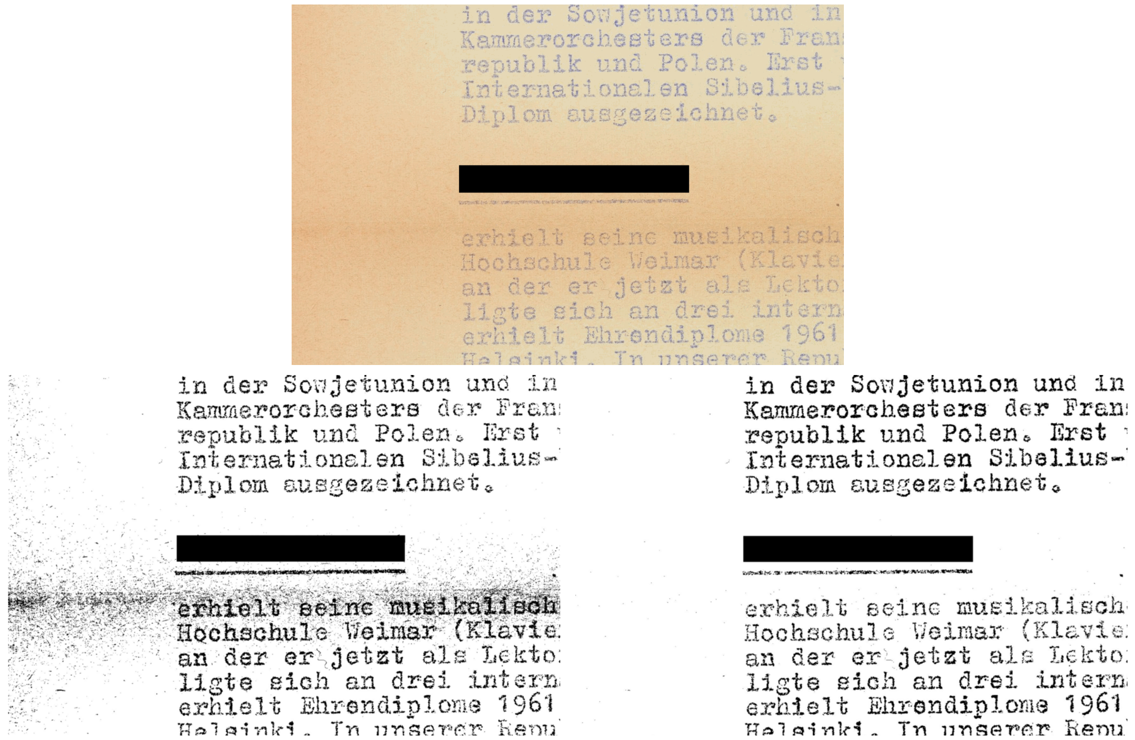
[redacted]

erhielt seine musikalische Ausbildung an der Franz-Liszt-Hochschule Weimar (Klavierstudium bei Prof. [redacted]), an der er jetzt als Lektor tätig ist. [redacted] beteiligte sich an drei internationalen Klavierwettbewerben und erhielt Ehrendiplome 1961 und 1966 in Budapest und 1962 in Helsinki. In unserer Republik hat sich der junge Pianist durch Klavierabende und als Solist in Sinfoniekonzerten nachhaltige Erfolge erspielt. Auch vom Rundfunk und Fernsehfunk wurde er verpflichtet. Konzertreisen führten ihn in die Bundesrepublik, nach Finnland, in die CSSR und in die Sowjetunion.

[redacted]

begann seine musikalische Ausbildung an der Franz-Liszt-Hochschule Weimar 1962 im Oberschulteil dieses Institutes und wurde 1966 in die Hochschule zum Studium bei Prof. [redacted] aufgenommen. Er ist 1. Preisträger bei den innerschulischen Wettbewerben 1967 und 1969 und seit 1970 Wilhelm-Pieck-Stipendiat. Im Oktober 1970 führte ihn eine erfolgreiche Konzertreise in die Georgische SSR. Ab September 1971 wird er ein Zusatzstudium am Tschaikowski-Konservatorium Moskau aufnehmen.

Figure 2.15: Hectography separation result using proposed method: left – original scan; center – extracted text component; right – extracted background and degradation component. Note: the background component was transformed back to RGB for visualization purposes and all person names have subsequently been blacked out due to privacy reasons



in der Sowjetunion und in
Kammerorchesters der Fran
republik und Polen. Erst
Internationalen Sibelius-
Diplom ausgezeichnet.

[redacted]

erhielt seine musikalisch
Hochschule Weimar (Klavie
an der er jetzt als Lekto
ligte sich an drei intern
erhielt Ehrendiplome 1961
Helsinki. In unserer Repu

in der Sowjetunion und in
Kammerorchesters der Fran
republik und Polen. Erst
Internationalen Sibelius-
Diplom ausgezeichnet.

[redacted]

erhielt seine musikalisch
Hochschule Weimar (Klavie
an der er jetzt als Lekto
ligte sich an drei intern
erhielt Ehrendiplome 1961
Helsinki. In unserer Repu

in der Sowjetunion und in
Kammerorchesters der Fran
republik und Polen. Erst
Internationalen Sibelius-
Diplom ausgezeichnet.

[redacted]

erhielt seine musikalisch
Hochschule Weimar (Klavie
an der er jetzt als Lekto
ligte sich an drei intern
erhielt Ehrendiplome 1961
Helsinki. In unserer Repu

Figure 2.16: Top: Portion of scanned hectography duplicate showing poor text quality and significant paper degradation; Bottom: Text component after binarization, using: left – grayscale transformation & Otsu-binarization [204]; right – proposed method. Note: all person names were blacked out due to privacy reasons

compare the proposed method with two classical approaches, each involving a traditional intensity-based grayscale transformation, followed by a subsequent binarization.

OCR Engine	**Otsu's method [204]**	**Sauvola et al. method [229]**	**Proposed method**
Tesseract 3 absolute dist. **relative dist.**	 35207 **82.2%**	 38360 **89.6%**	 17184 **40.1%**
Abbyy FineReader 8 absolute dist. **relative dist.**	 15193 **35.5%**	 13754 **32.1%**	 2182 **5.1%**

Table 2.2: Comparison of OCR results from two independent engines on a dataset of 22 hectography images containing 42 825 characters (5833 words). The Levenshtein distance [162] to manually-generated ground truth was used as evaluation measure.

The two binarization methods used in the comparison are the globally optimal method of N. Otsu [204] and the adaptive binarization method of Sauvola et al. [229]. Both binarization methods are well-known in the document analysis research community and have consistently exhibited a very good performance on document images according to recent comparisons [105, 115]. A more detailed description of both methods may be found in section 3.1. Since the hectography images are overall rather bright, fact which would greatly disadvantage the Sauvola method, we multiply the local thresholds for each pixel by a factor of 1.2. The local window radius used was 35×35, as determined to be optimal in relation to the dominant character size on the test data set.

From the obtained results it is obvious that the proposed method brings about very significant improvements in the OCR rates of both engines. The much higher error rate of Tesseract can be traced back to the fact that it uses connected components labeling for feature extraction. On historic documents where the ink is nearly washed out, most connected components consist of small broken parts of characters and are as such unsuitable as basis for the higher-level recognition task.

2.2.3 Character Enhancement for Hot Metal Typesetted Prints

The quality of the results of the optical character recognition is directly influenced on one side by the quality of the scanning process and by the printing process on the other side. In a digitization workflow the human operator can control the scanning process of the documents and directly take the appropriate measures to deal with scanning errors in the digitization process. By contrast, printing is normally completed a long time before the scanning procedure and is out of the control of the scanning operator performing the retro-digitization. Therefore, the development of algorithms capable of alleviating the problems occurring in the printing process is a highly desirable endeavor.

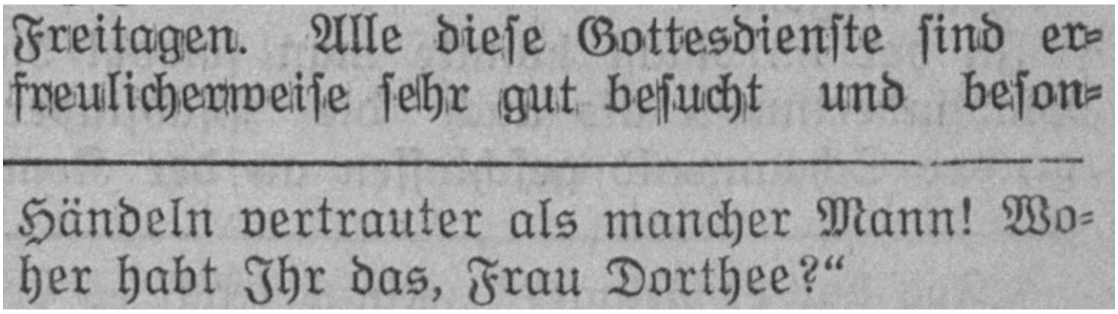
Figure 2.17: Composing stick with a partially finished type directly above a type case. Source: Willi Heidelbach

In this chapter we describe such a method for improving the quality of digitized text documents initially produced using letterpress printing. Specifically, we focus on documents printed during the time period spanning from the beginning until the middle of the 20^{th} century. In this period, a very widespread typesetting technique was hot metal typesetting (also known as hot lead typesetting). This technique represented one of the earliest attempts at mechanizing the printing process, which in turn facilitated its use on an industrial scale. The process consisted of injecting molten type metal (with lead as its main constituent) into molds, each having the shape of a single character or a ligature. The obtained sorts were subsequently used to press ink onto paper and produce the print. A common problem with the procedure was the fact that ink tended to infiltrate between the individual sorts upon imbuing, thus producing specific near-vertical artifacts upon pressing them against the paper. As can be seen in figures 2.18 and 2.21, such artifacts are quite prominent and have a comparable stroke width as well as the exact same gray level as regular characters in the digitized image. This fact makes it virtually impossible for the artifacts to be removed effectively by using any state-of-the-art global or local thresholding algorithms. A wide selection of thresholding algorithms, alongside with a performance comparison on a pixel-accurate ground truth can be found in the paper of Gatos et al. [115].

As such, we follow a different approach, presented in more detail in the following sections. For more examples of closely-related research work in the area of character enhancement one may consult the overview section 2.2.1. The proposed approach is evaluated on a dataset consisting of old German-language newspaper excerpts via the OCR results obtained from two well-known OCR engines, namely ABBYY FineReader [1] and Google's Tesseract [15].

2.2.3.1 Analysis of Letterpress Printing Artifacts

For obtaining an effective text enhancement method it is essential to analyze the nature of the printing artifacts as well as their direct effects beforehand. As mentioned in the introductory section, the targeted artifacts are produced by ink infiltrating between the grouped metal sorts upon imbuing and then leaking onto the paper as the sorts are pressed against the paper sheet during the printing process. Since the traditionally used metal sorts had a rectangular body shape, the ink leaks invariably resulted in near-vertical dashes. More notably because of the fact that hot metal typesetting was designed to automate the printing process, it has found a widespread use starting from the end of the 19^{th} century up until the mid-20^{th} century. As such, most archives containing historical newspapers from the aforementioned period suffer from this kind of artifacts.



Freitagen. Alle dieſe Gottesdienſte ſind er=
freulicherweiſe ſehr gut beſucht und beſon=

Händeln vertrauter als mancher Mann! Wo=
her habt Ihr das, Frau Dorthee?“

Figure 2.18: Portion of newspaper image affected by artifacts caused by hot metal typesetting alongside an unaffected area (as appearing in the original print)

Apart from an aesthetically unpleasant appearance, the affected retro-digitized documents suffer from a more acute problem: a low quality OCR result in the text plane. The artifacts, in the form of vertical line segments located between the glyphs are often recognized as spurious "i"s, "I"s, "l"s and sometimes "t"s in the OCR process. Unfortunately, the OCR errors are more severe and difficult to detect as soon as the artifacts intersect the edges of the glyphs. In such cases, due to the similar stroke width of the characters and that of the redundant vertical segments, the characters where the intersection occurs are wrongly recognized by the OCR. Typical examples include the letter “n” recognized as “m”, “c” recognized as “o”, “r” recognized as “n” and “e” recognized as “o”.

In order to correct such errors automatically, the proposed algorithm makes use of and requires information about the textual/font characteristics of the containing text regions. Such information is usually only available in the later stages of the document image analysis (DIA) process. This does not represent a problem however, since the OCR process is typically the very last step in the DIA chain. In the following section we consider that we have as input a skew-free binary image as well as a complete list of text lines found on the given document page. Note that this implicitly assumes that page segmentation has already been performed and a text line detection algorithm (such as the one proposed by Breuel [59]) has been applied on the document regions identified as containing text.

2.2.3.2 Algorithm Description

Our algorithm works on a line-by-line basis and as such can be applied independently and concurrently on any set of text lines. An initial set of candidate artifacts is extracted from each text line. Then, a set of features is computed for the text line (i.e. x-height, capital height, baseline, dominant stroke width) and each candidate artifact located within it (height, stroke width, height ratio to containing connected component, presence of a diacritic above, adjacency to a counter (bowl)). Using a generic, hand-crafted decision tree classifier and the aforementioned features, the candidate artifacts are categorized as true artifacts or legitimate characters/ character parts. Exchanging the classifier is easily possible, however one must note that training the new classifier would require pixel-accurate training data, which is a non-trivial endeavor. Also, one must keep in mind that the real objective is not a perfect classifier for artifacts, but an overall improved OCR rate. As such, some types of artifacts would influence the OCR result more than others and an “optimal” classifier would certainly need to also take this aspect into account.

A first approximation for the set of candidate printing artifacts is obtained by computing the set of all near-vertical line segments located within the bounding rectangle of the text line. The vertical line detection employed is based on *horizontal directional single-connected chains* (DSCC). The concept of a DSCC was initially proposed in 2001 by Zhang et al. [302] for the detection of horizontal and vertical lines in printed forms. Here we apply the same algorithm for the detection of vertical line candidates corresponding to possible printing artifacts and located both inside as well as in-between the text characters. A very valuable asset of the DSCC-based detection in comparison to the many other existing line detection algorithms is the pixel-accurate detection of the separators. This allows for an exact removal of the artifacts, should the line segments further on be identified as such.

In general, a vertical DSCC C_V consists of an array $R_1 \dots R_N$ of vertically adjacent horizontal black pixel run-lengths. A horizontal black pixel run-length is defined as a sequence of adjacent black pixels bounded on each side by a white pixel. A DSCC is then formed by a maximal-sized group of neighboring horizontal black pixel run-lengths that have exactly one black run-length neighbor both above and below. In other words, the growth process for a DSCC ends in one direction as soon as either its highest/lowest horizontal black run-length has either zero or more than one black run-length neighbor directly above it/below it. Figure 2.19 shows the result of the artifact candidate detection process on a single connected component. For each artifact candidate a midpoint line fit is computed, and merging of DSCCs which lie on the same supporting line is performed in the same manner as proposed by Zhang et al. (including a minimum height threshold of 4 pixels,

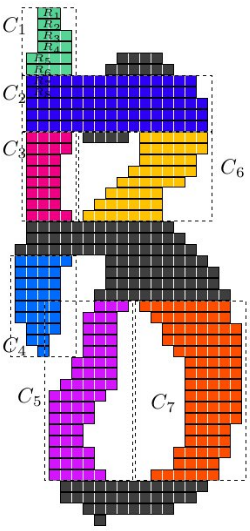


Figure 2.19: Gothic minuscule "z" and all its contained vertical DSCCs with a height greater than 4 pixels superimposed

which can be adjusted according to the document scanning resolution). For example, in figure 2.19 only the DSCCs $1 - 7$ are considered as possible candidates, because all other possible DSCCs are below the height threshold. In addition, we merge all pairs of DSCCs which are located at a larger distance from each other, but having in-between (i.e. on the supporting line) a majority of black pixels. An example of such a situation can be seen in figure 2.19 for DSCCs C_1 and C_4. Finally, based on the observations about the nature of the artifacts from the previous section, we remove from the candidate list all DSCCs deviating from the vertical more than 15° and having a height-to-width aspect lower ratio lower than $1 : 1.5$.

Next, we make use of the *median stroke width* of the characters located on the given text line. The stroke width can be computed exactly and efficiently via a distance transform [181]. For more information about the stroke width computation process, we refer the interested reader to section 6.3.1.1. As an (less accurate) alternative, one may approximate this value by a simple median of the lengths of the horizontal black run-lengths which were already identified as part of the creation of the DSCCs. In the baseline version of our algorithm (tested in the evaluation section) we keep in the candidate list only those vertical segments with a stroke width strictly lower than the median character stroke width. If it is apriori known that the text line contains (severe) printing artifacts, this restriction may be relaxed. However, in the case of unaffected text lines, this filtering is essential for reducing the number of false positives. At this point it is worth noting that by depending only relatively to the stroke width we are independent from both the scanning resolution (e.g. in contrast to [114]) and typeface properties such as boldface or italics.

Another essential feature for restricting the list of potential artifact candidates is the *x-height* of the characters forming the text line. We compute the x-height via a 1-dimensional k-Means clustering [129] of the character heights. The number of classes is fixed to 2 (two) and the median height of the characters belonging to the cluster corresponding to the lower height represents our estimate for the x-height. Note that for text lines containing only majuscules or minuscules this approach will obviously not converge to the actual x-height, but in such cases we found it actually desirable to adapt and allow for taller, respectively shorter artifacts. Subsequently, we simulate the deletion of each

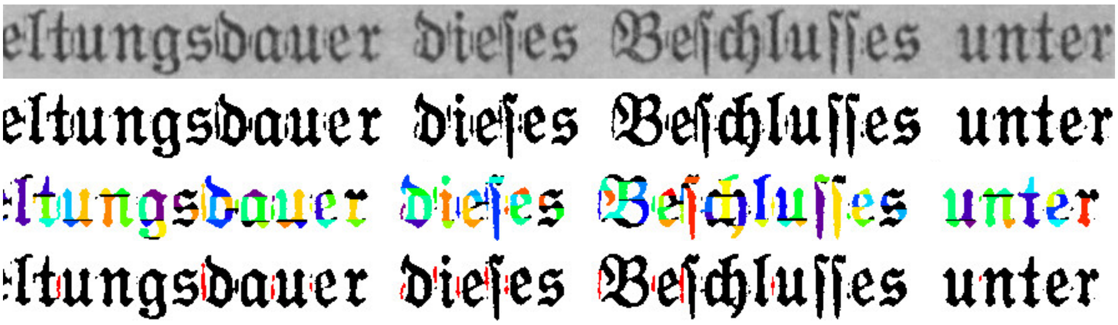

eltungsdauer dieſes Beſchluſſes unter
eltungsdauer dieſes Beſchluſſes unter
ltungsdauer dieſes Beſchluſſes unter
ltungsdauer dieſes Beſchluſſes unter

Figure 2.20: Illustration of the proposed character enhancement algorithm's steps. Top to bottom: original grayscale text line; result of binarization using Otsu's method [204]; superimposed initial vertical DSCCs; determined true artifacts (red)

candidate from its containing connected component. If the height of the resulting component height does not change significantly the candidate is kept, as it likely represents a superfluous protrusion. The other situation when the candidate is kept in the list is when its deletion has cause the connected component to (almost) completely disappear. In the latter case we are very likely dealing with an isolated printing artifact. Note that both cases can be described using the ratio between the height of the artifact candidate and that of its containing connected component (with the artifact part deleted).

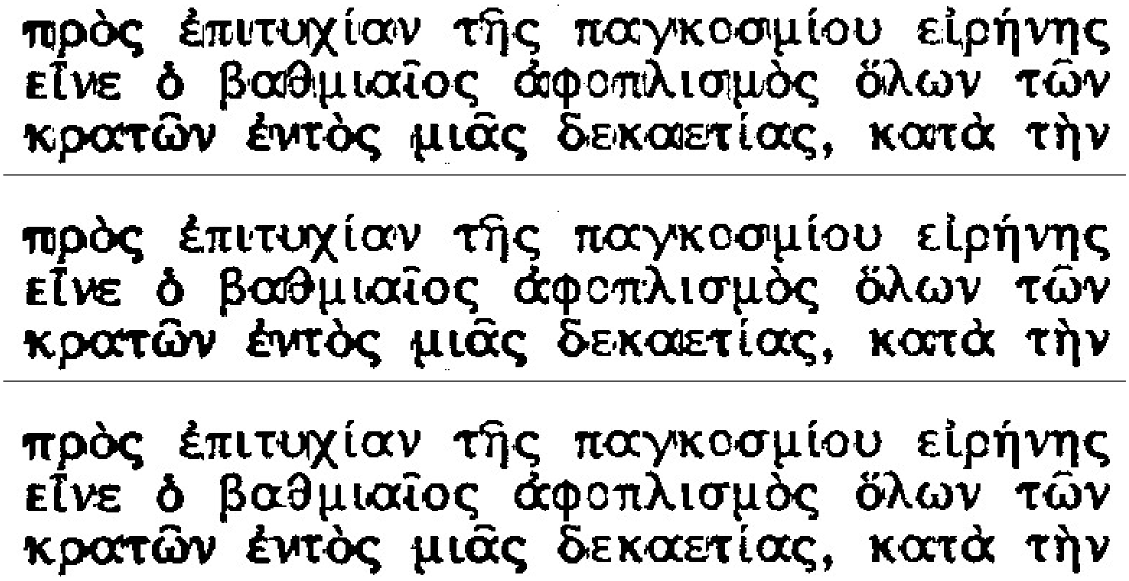

πρὸς ἐπιτυχίαν τῆς παγκοσμίου εἰρήνης
εἶνε ὁ βαθμιαῖος ἀφοπλισμὸς ὅλων τῶν
κρατῶν ἐντὸς μιᾶς δεκαετίας, κατὰ τὴν
πρὸς ἐπιτυχίαν τῆς παγκοσμίου εἰρήνης
εἶνε ὁ βαθμιαῖος ἀφοπλισμὸς ὅλων τῶν
κρατῶν ἐντὸς μιᾶς δεκαετίας, κατὰ τὴν
πρὸς ἐπιτυχίαν τῆς παγκοσμίου εἰρήνης
εἶνε ὁ βαθμιαῖος ἀφοπλισμὸς ὅλων τῶν
κρατῶν ἐντὸς μιᾶς δεκαετίας, κατὰ τὴν

Figure 2.21: Results of font enhancement on Antiqua font and Greek script: top – original image; center – results of morphological processing, as in [114]; bottom – results of proposed method. Top and center image source: [114]

At this point the decision tree classifier already produces satisfactory results with the notable exception of two cases: the small letter "i" on one side and the set of letters containing counters (e.g. "e", "d", "a", "o", "g", "p"). In both situations the tree would identify portions of the respective characters as likely candidates (i.e. either the stem of the "i" or the bowls of the other characters), thus leading to many false positives. The first case can be readily dealt with by searching for a small connected component resembling a dot located right above the candidate and implementing a corresponding rule using the binary feature. The second case can be identified just as easily, with the sole difference that it additionally involves the extraction of all background connected components from the bitmap of the text line and a straightforward adjacency test.

2.2.3.3 Evaluation

The evaluation data set consists of 52 single-column text-only excerpts from grayscale newspaper pages printed between 1920 and 1950 and totaling more than 63 000 characters. The newspaper pages originate from the German-language newspaper “Liechtensteiner Volksblatt” and feature a Fraktur script. We have chosen document images printed using this highly decorative script on purpose so as to assess the robustness of the proposed method. As can be seen in figure 2.21, the proposed technique can readily handle more traditional typefaces, such as Antiqua, as well as non-Romanic scripts. The data set was split about evenly into two groups of images: one containing only text regions clearly affected by printing artifacts and the other containing completely uncorrupted regions. The distinction was made in order to be able to obtain more meaningful evaluation results. This is because a typical newspaper image contains both kinds of regions, irregularly mixed and widely differing in size (number of characters), thus potentially skewing the results greatly in one direction or the other. Also, by having separate evaluation results for affected and unaffected regions one may readily compute a weighted average as an approximation of the expected quality for any image featuring mixed content.

Method	**Affected dataset** **33460 chars** 5034 words 27 images	**Unaffected dataset** **30036 chars** 4730 words 25 images
Tesseract	4793	743
Tess + Enhance	2288	756
Tess + Enhance relative diff.	**52.3%**	**-1.7%**
FineReader	1720	424
FR + Enhance	1074	441
FR + Enhance relative diff.	**37.5%**	**-4%**
Overall relative diff.	**44.9%**	**-2.8%**

Table 2.3: Levenshtein distance [162] from the ground truth without and with the proposed font enhancement method, using Tesseract [15] and ABBYY FineReader [1] as OCR engines

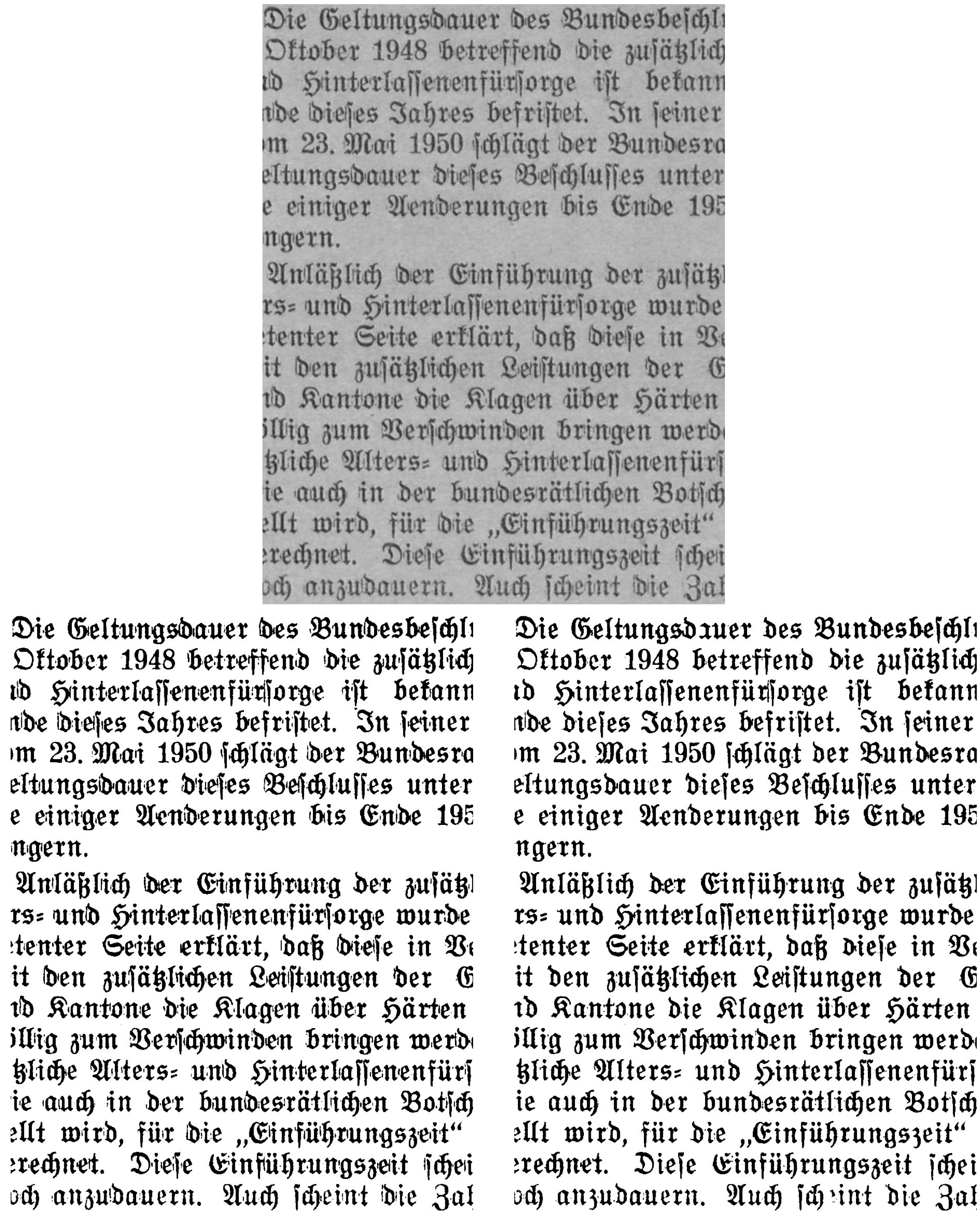
The proposed artifact removal procedure has as primary goal the qualitative improvement of OCR results. Therefore we have chosen as evaluation measure the Levenshtein distance [162] in conjunction with manually corrected OCR ground truth. As already noted in section 2.2.2.3, the choice of the Levenshtein distance as evaluation measure was made because it is directly proportional to the amount of human labor required to reach a perfect text result. Since perfect OCR results are in general the end objective of every library or content-owner, having an estimate of the amount of necessary manual correction work also indirectly provides a highly useful cost estimate. Thus, minimizing the Levenshtein distance translates into minimizing digitization costs.

In order to ensure that the obtained enhancements are indeed generic and not specific to a particular OCR method, we perform the evaluation using two different, well-known OCR engines: the open source Tesseract v3 [15] and the commercial product ABBYY Finereader 7 [15]. The exact same manually corrected rectangular regions (each corresponding to a single text line) were fed into both OCR engines and the Levenshtein distance was computed on a line-by-line basis. All newspaper images were binarized using Otsu's global thresholding method [204] prior to the application of the enhancement and/or the OCR procedure. Note that the primary purpose of our evaluation is to assess the quality of the proposed enhancement method in the absence of any other external factors. Because of this, document images were specifically chosen so that the scan quality was good enough to allow the application of a global binarization method.

As can be seen from table 2.3 our method manages to consistently improve the OCR results of both engines on affected regions. At the same time, the quality loss of the OCR results on uncorrupted text regions is minimal. As such, this is a very encouraging result. In addition, the false positive rate can potentially be reduced even further by observing and appropriately handling the specific situations in which the algorithm still fails. However, we believe that the addition of a (binary) classifier capable of separating corrupted from uncorrupted regions would potentially result in a much more substantial improvement in OCR quality. Note that such a classifier would have a task very similar to that of optical font classification (OFR), area which has lately seen promising developments. For example, Lidke et al. [166] were able to obtain recognition rates of over 93% while only using 5-letter words as query samples. The prior classification on a region-by-region or line-by-line basis would allow for a more aggressive artifact filtering in heavily affected areas. From the newspaper image excerpt in figure 2.21, one may clearly see that there is still room for improvement especially for thick artifacts intersecting characters.

2.3 Conclusions

The current chapter was logically divided into two parts. In the first part we introduced generic quality problems affecting digital images, such as blur, poor contrast, noise. We have seen how image quality measures operate and how these measures can be used for the detection of the aforementioned deteriorations. State-of-the-art removal methods for each type of quality problem have been reviewed. In a professional digitization environment, a generic solution for obtaining accurate image quality measurements is the use of standardized color reference targets. We introduced a fully automatic approach for the detection of classic color checker targets in digital images. Together with the existing reference color target information, the detected color patches can be used to produce precise quality measurements and thus enable more accurate decisions about the suitability of each given digital image. Our main focus is a direct and unrestricted applicability in mass document processing, where robustness and flexibility are of paramount importance. The introduced technique can be readily extended to other types of color reference targets, including the digital color checker SG as well as the UTT (see fig. 2.6). The achieved recall rate of 97.1% on a set of 239 real-life document- and photograph scans featuring partially occluded color targets in different orientations confirms the robustness of our algorithm.



Die Geltungsdauer des Bundesbeſchl
Oktober 1948 betreffend die zuſätzlich
ıd Hinterlaſſenenfürſorge iſt bekann
nde dieſes Jahres befriſtet. In ſeiner
m 23. Mai 1950 ſchlägt der Bundesra
eltungsdauer dieſes Beſchluſſes unter
e einiger Aenderungen bis Ende 195
ngern.

Anläßlich der Einführung der zuſätzl
rs= und Hinterlaſſenenfürſorge wurde
tenter Seite erklärt, daß dieſe in Ve
it den zuſätzlichen Leiſtungen der E
ıd Kantone die Klagen über Härten
illig zum Verſchwinden bringen werd
tzliche Alters= und Hinterlaſſenenfürſ
ie auch in der bundesrätlichen Botſch
ellt wird, für die „Einführungszeit"
rechnet. Dieſe Einführungszeit ſchei
och anzudauern. Auch ſcheint die Zal

Die Geltungsdauer des Bundesbeſchl
Oktober 1948 betreffend die zuſätzlich
ıd Hinterlaſſenenfürſorge iſt bekann
nde dieſes Jahres befriſtet. In ſeiner
m 23. Mai 1950 ſchlägt der Bundesra
eltungsdauer dieſes Beſchluſſes unter
e einiger Aenderungen bis Ende 195
ngern.
Anläßlich der Einführung der zuſätzl
rs= und Hinterlaſſenenfürſorge wurde
tenter Seite erklärt, daß dieſe in Ve
it den zuſätzlichen Leiſtungen der E
ıd Kantone die Klagen über Härten
illig zum Verſchwinden bringen werd
tzliche Alters= und Hinterlaſſenenfürſ
ie auch in der bundesrätlichen Botſch
ellt wird, für die „Einführungszeit"
rechnet. Dieſe Einführungszeit ſchei
och anzudauern. Auch ſcheint die Zal

Die Geltungsdauer des Bundesbeſchl
Oktober 1948 betreffend die zuſätzlich
ıd Hinterlaſſenenfürſorge iſt bekann
nde dieſes Jahres befriſtet. In ſeiner
m 23. Mai 1950 ſchlägt der Bundesra
eltungsdauer dieſes Beſchluſſes unter
e einiger Aenderungen bis Ende 195
ngern.
Anläßlich der Einführung der zuſätzl
rs= und Hinterlaſſenenfürſorge wurde
tenter Seite erklärt, daß dieſe in Ve
it den zuſätzlichen Leiſtungen der E
ıd Kantone die Klagen über Härten
illig zum Verſchwinden bringen werd
tzliche Alters= und Hinterlaſſenenfürſ
ie auch in der bundesrätlichen Botſch
ellt wird, für die „Einführungszeit"
rechnet. Dieſe Einführungszeit ſchei
och anzudauern. Auch ſch·int die Zal

Figure 2.22: Left: portion of original grayscale image; center: binarization result using Otsu's method [204]; right: result obtained using the proposed font enhancement method

The second part of the chapter is focused on image enhancement methods specifically targeted at document images. We gave an overview of the state of the art in document enhancement, broken down into sub-areas: margin noise removal, print area detection, speckle- and stroke-like noise removal, character enhancement, bleed-through removal. An important issue observable from the review is that document enhancement algorithms capable of handling color document images are few and far in between. This represents a significant problem, because a theoretically optimal removal of the deterioration effects can only be accomplished when taking into account the full color information available.

We specifically addressed the aforementioned issue within the proposed method for foreground extraction aimed at low-quality hectographic documents. The introduced additive generative model is not restricted solely to hectographic documents, but can robustly deal with any historical documents using a single ink color. Our approach uses full color information, does not require or rely upon an apriori model for the noise component nor on knowledge of the spatial correlation of close-by pixels. The separation of the additive components is performed by means of independent component analysis. We tested the validity of the proposed algorithm on a dataset of 22 historical documents containing 42 825 characters in 5833 words. The obtained results show that the new method provides an order of magnitude improvement in OCR performance over traditional document analysis approaches. Our current research concentrates on the adaptation of the proposed algorithm to other historical document types, such as old typewriter documents with strong noise components. To this end we are examining the possibilities of introducing additional spatial information into our approach.

Finally, in section 2.2.3 we introduced a new technique for the effective removal of letterpress printing artifacts occurring in historical newspapers. The importance of dealing with this problem is given by the fact that hot metal typesetting was the first non-manual printing technique widely used throughout the late 19^{th} and early 20^{th} century by many publishers around the globe. Such artifacts typically appear as thin lines between single characters or glyphs and are in most cases connected to one of the neighboring characters. The OCR quality is heavily influenced by this kind of artifacts. The nature of the artifacts makes traditional filtering (e.g. frequency domain, morphological processing) approaches unfeasible. The proposed method is based on the detection of near-vertical segments by means of directional single-connected chains. The use of a simple, interchangeable decision tree classifier based on font- and script-independent features allowed us to robustly deal with real-life document scans featuring both simple (e.g. Antiqua family) and complex decorative fonts (e.g. Fraktur). Most importantly, we were able to significantly relax the traditional assumptions in the DIA community about the artifacts and the input image quality and show that a resolution-independent processing of prints exhibiting artifacts with a stroke width even higher than that of most thin characters stems is indeed possible. We evaluated our approach on a 63 000 character dataset consisting of old newspaper excerpts printed using Fraktur fonts and spanning a time period of around 50 years. The recognition results on the enhanced images using two independent OCR engines (ABBYY FineReader and Tesseract) showed significant improvements over the originals in view of further manual correction. Further research shall focus on the automatic detection of affected regions/text lines in order to allow a selective (and consequently much more effective) application of the algorithm, as well as on evaluating different combinations of classifiers and features for a more accurate artifact classification.

Chapter 3

Color Quantization for Document Images

> *It is now known to science that there are many more dimensions than the classical four. Scientists say that these don't normally impinge on the world because the extra dimensions are very small and curve in on themselves, and that since reality is fractal most of it is tucked inside itself. This means either that the universe is more full of wonders than we can hope to understand or, more probably, that scientists make things up as they go along.*
>
> – T. Pratchett *(Pyramids)*

Nowadays the majority of existing digitized material exists in the form of binary or grayscale images. For a long time this artificial restriction was dictated by the lack of sufficient storage space, limitations of the digitization devices and the fact that many historical materials feature only a single foreground color. The situation is changing now, as storage space becomes plentiful and digitization solutions offering good quality are much more affordable. As we have seen in section 2.2.2, even though most historical materials are indeed printed using a single ink color, it is still beneficial to have as input a full-color image in order to be able to effectively separate the strong paper degradations from the foreground. The advantage of color scans becomes even clearer when dealing with modern magazines or journals which feature diverse and colorful layouts. Consequently, increasingly many documents are being digitized into full-color images. While this is indeed a sound approach for mass digitization, it also greatly increases the burden placed upon the automatic DIU systems. The dimensionality of the input (search) space grows at least threefold, making the search for a solution exponentially more difficult. In this context, color quantization/binarization plays the crucial role of intelligently reducing the dimensionality of the data back to a manageable size for the subsequent layout analysis modules. The problem of dimensionality reduction is complicated by the fact that mathematically optimal solutions do only seldom produce satisfactory results, as they cannot take the specific nature of printed material into account.

In its standard definition, *thresholding* is a transformation of a *grayscale* input image f to an output (segmented) image g, by using the following formula: $g(i,j) = \begin{cases} 1, & f(i,j) \geq T \\ 0, & f(i,j) < T \end{cases}$.

There exist many variations of this basic formula, such as $g(i,j) = \begin{cases} f(i,j), \; f(i,j) \geq T \\ 0, \; f(i,j) < T \end{cases}$ (*semi-thresholding*) or $g(i,j) = \begin{cases} 1, \; f(i,j) \in D_1 \\ 2, f(i,j) \in D_2 \\ \quad \ldots \\ n, \; f(i,j) \in D_n \\ 0, \text{ otherwise} \end{cases}$ (*multilevel thresholding*). One may easily notice that all these formulae basically represent just special cases of color quantization. Instead of looking at scalar values, color quantization regards the image pixels as (color) vectors of arbitrary dimensionality and the comparison operations are performed via well-known color distance formulae (e.g. CIE76 delta E [138]). It is known that the problem of K-clustering with variable K is NP-complete [111] and consequently does not allow for a computationally-efficient implementation on today's computers. This is probably an important reason for the proliferation of approximate solutions available in the specialized literature (see section 3.2.1).

The first section of the current chapter is entirely dedicated to thresholding methods using the standard definition. Such approaches are still by far the most widely used in the context of document understanding as they provide the highest amount of data reduction for the further processing stages. Because the use of a single threshold value yields two classes, the process is commonly referred to as *binarization*. We start by reviewing the state of the art in document binarization and pointing out the strengths and weaknesses of each algorithm class. In the second part of the section we propose a novel binarization framework as a way of retaining all the algorithms' strengths while minimizing their weaknesses. Thereby we make it possible to adapt any global algorithm relying on histogram data into a local algorithm, while maintaining a running time constant in the number of image pixels. The proposed framework is evaluated by the adaptation of two effective globally optimal algorithms for document binarization: Otsu's method [204] and the method of Kittler and Illingworth [151].

We start the second section by reviewing the current research status in color image processing, more specifically we address generic color filtering and color reduction, as well as a few methods developed specifically for document images. We identify the most important deficiencies in the state-of-the-art methods for color document processing and discuss the corresponding generic solutions. By looking at the practical example of the system described by Rodriguez Valadez [224] we analyze a few additional, otherwise hard to discover issues regarding the meaningful evaluation of color reduction results. Finally, as a promising research direction for the document image analysis community, we offer a possible holistic solution for the document color reduction problem.

3.1 Binarization

Unless otherwise noted, throughout the current section we shall assume that the input document images consist of gray-level values which can be encoded using at most 8-bit values, i.e. they contain at most 256 shades of gray. Approaches dealing with document image binarization are traditionally differentiated into *global* or *local (adaptive)* approaches. Global methods use a single computed threshold value to classify image pixels, whereas adaptive methods use many different threshold values selected according to the local area information. In both cases however, there are exactly two resulting classes. The output classes (corresponding to the values 0 and 1) are commonly referred to as *background* and *foreground (object)*, respectively.

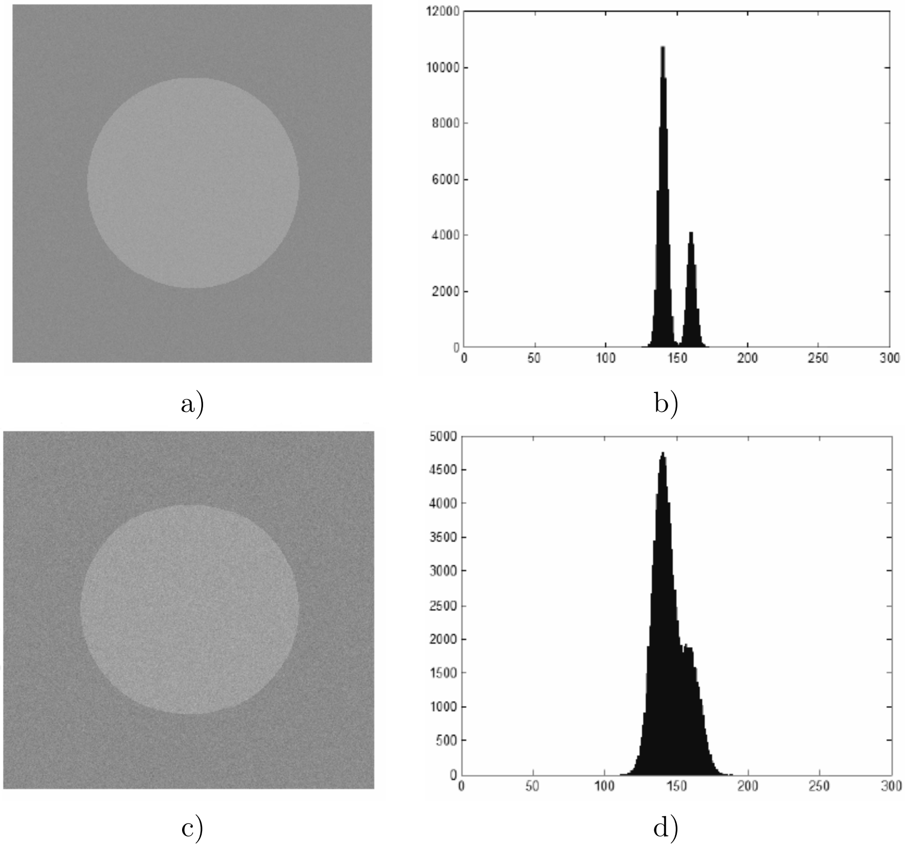
Despite the huge variety of algorithms available, the binarization problem is far from solved. No less than 35 methods took part in the 2009 Document Image Binarization Contest (DIBCO) [115], whereas the 2011 DIBCO [218] comprised 18 different binarization techniques. The competitions have shown that the most effective current binarization techniques employ specialized pre- and post-processing (e.g. the winning algorithms of both DIBCO'09 and DIBCO'11). The pre- and post-processing normally consists of speckle noise removal, hole filling and in some cases the merging of broken characters using heuristic rules. For more information on such enhancement algorithms, the reader may refer to the sections 2.1.1 and 2.2.1. In order to obtain an even better binarization result, one may also resort to region type-specific binarization algorithms combined in a multi-pass segmentation system. For example, the algorithm of Sauvola et al. [229] was found to perform very well for text on darker backgrounds, the Konya et al. algorithm [155] produces good results for overexposed text areas, whereas for halftones a dithering algorithm [107] is more appropriate for human perception.

3.1.1 Global Methods

The choice of the threshold value is crucial for global thresholding methods. However, finding a good global threshold value is not trivial even in seemingly easy situations. For example, figure 3.1 shows two histograms resulting from the same image, corrupted with two different noise levels. For a human observer, the noise in the two images is barely perceptible, despite the fact that the resulting histograms look very different.

The simplest of the global thresholding algorithms assume a bimodal distribution of the histogram values, and choose as threshold value the arithmetic mean of the two maxima. As seen from the previous figure, the bimodality of the histogram (even in a near-perfect case) is greatly influenced by noise. This makes the detection of the two maxima highly unreliable in many cases. In order to deal with this problem more robustly, one may regard grayscale images as containing two main gray-level regions, each having a certain probability density function (PDF). In this case, the image histogram represents the sum of the two probability density functions. If both PDFs are approximately known or assumed to have a certain form (e.g. Raleigh, Gaussian, Poisson), it is possible to determine an optimal threshold between the two classes in terms of minimizing the fitting error. Figure 3.2 illustrates an example of this technique.

One problem with the aforementioned approach is that the estimation of the parameters of the density functions gets complicated, even in case the PDFs are assumed to have a well-known shape. More importantly, in general there exists no hard guarantee that the probability distributions will actually follow the assumed patterns. Without assuming any special form for the two PDFs, it is still possible to define a separability criterion between the object and the background classes. An *optimal threshold* in such case would be located at the histogram position which maximizes/minimizes the chosen criterion. Usually, the separability criterion is related to the gray-level variance between objects and background. Arguably the most well-known such method was introduced by N. Otsu [204]. This algorithm is still frequently used today, due to the fact that it is very fast and works in a wholly unsupervised manner (i.e. requires no parameters). The good accuracy of Otsu's method was confirmed by several comparison papers [105, 115, 269], where it outperformed all other tested global thresholding methods in the case of document images. In the following, we discuss the Otsu thresholding method in more detail, as it represents



a) b)

c) d)

Figure 3.1: a) Image with low noise level and b) its corresponding histogram; c) image with a higher level of noise and d) its corresponding histogram [44]

a good prototype for optimal global methods and it shall also be used to exemplify the adaptation framework introduced in the second part of the chapter. Another noteworthy optimal global algorithm which uses a slightly different criterion was proposed by Kittler and Illingworth [151]. Their algorithm was the best performing technique in the recent comparison by Sezgin and Sankur [237] on a dataset containing mixed document and non-document images.

We shall now shortly describe the Otsu global optimal thresholding algorithm. We have seen that all global optimal methods rely exclusively on information available in the image histogram, and regard the latter as the combination of two PDFs, one corresponding to the background and the other to the foreground (object) pixels. The main difference between optimal binarization methods is the choice of the separability criterion for distinguishing between the object and background classes. Given a separability criterion, an optimal threshold is defined as histogram location which maximizes the chosen criterion. In Otsu's method, the separability criterion is given by the total variance between the pixels belonging to the background and the foreground.

Let us assume that the pixels of a given image can be represented using L gray levels, denoted $[1, 2, \ldots, L]$. Most commonly, $L = 256$ for the classic grayscale images, but the method is generally applicable for any number of gray levels. Considering the total number of pixels in the image to be N, we denote the number of pixels having gray level i by n_i.

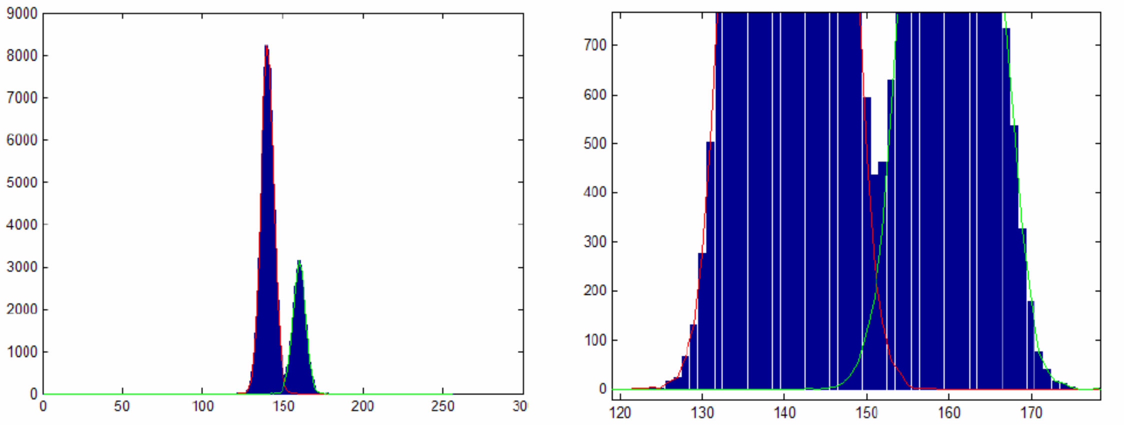


Figure 3.2: Left: Image histogram modeled as a sum of two Gaussian PDFs; Right: Close-up showing the intersection point of the two density functions, which corresponds to the optimal threshold [44]

Thus, the normalized histogram is given by:

$$p_i = \frac{n_i}{N},\ i = \overline{1, L}$$

Next, we assume that the image pixels are divided into two disjoint classes: C_0 and C_1, and that the gray level threshold value k separates the two classes, such that all pixels with gray levels less or equal to k belong to C_0 and all other pixels belong to C_1. With these notations, the two probabilities of class occurrence are:

$$\omega_0 = P(C_0) = \sum_{i=1}^{k} p_i \ ;\ \omega_1 = P(C_1) = \sum_{i=k+1}^{L} p_i = 1 - \omega_0$$

The mean (expected) values for each of the classes can now be computed as:

$$\mu_0 = \sum_{i=1}^{k} i\, P(i|C_0) = \frac{\mu_k}{\omega_k};\ \mu_1 = \sum_{i=k+1}^{L} i\, P(i|C_1) = \frac{\mu_T - \mu_k}{1 - \omega_k},\ \text{where}$$

$$\omega_k = \omega_0 = \sum_{i=1}^{k} p_i,\ \mu_k = \sum_{i=1}^{k} i p_i \text{ and } \mu_T = \mu(L) \sum_{i=1}^{L} i p_i$$

In the usual case of grayscale images, $\mu(L) = \mu(256) = \frac{1}{256}\sum_{i=0}^{L-1} p_i = \frac{255}{2}$. It can be easily verified that $\omega_0\mu_0 + \omega_1\mu_1 = \mu_T$. The variances for the two classes are given by:

$$\sigma_0^2 = \sum_{i=1}^{k} (i - \mu_0)^2 P(i|C_0) = \sum_{i=1}^{k} \frac{(i - \mu_0)^2 p_i}{\omega_0},$$

$$\sigma_1^2 = \sum_{i=k+1}^{L} (i - \mu_1)^2 P(i|C_1) = \sum_{i=k+1}^{L} \frac{(i - \mu_1)^2 p_i}{\omega_1}$$

By making use of the variances of the two classes, there exist several options for evaluating the goodness of a certain threshold k, namely:

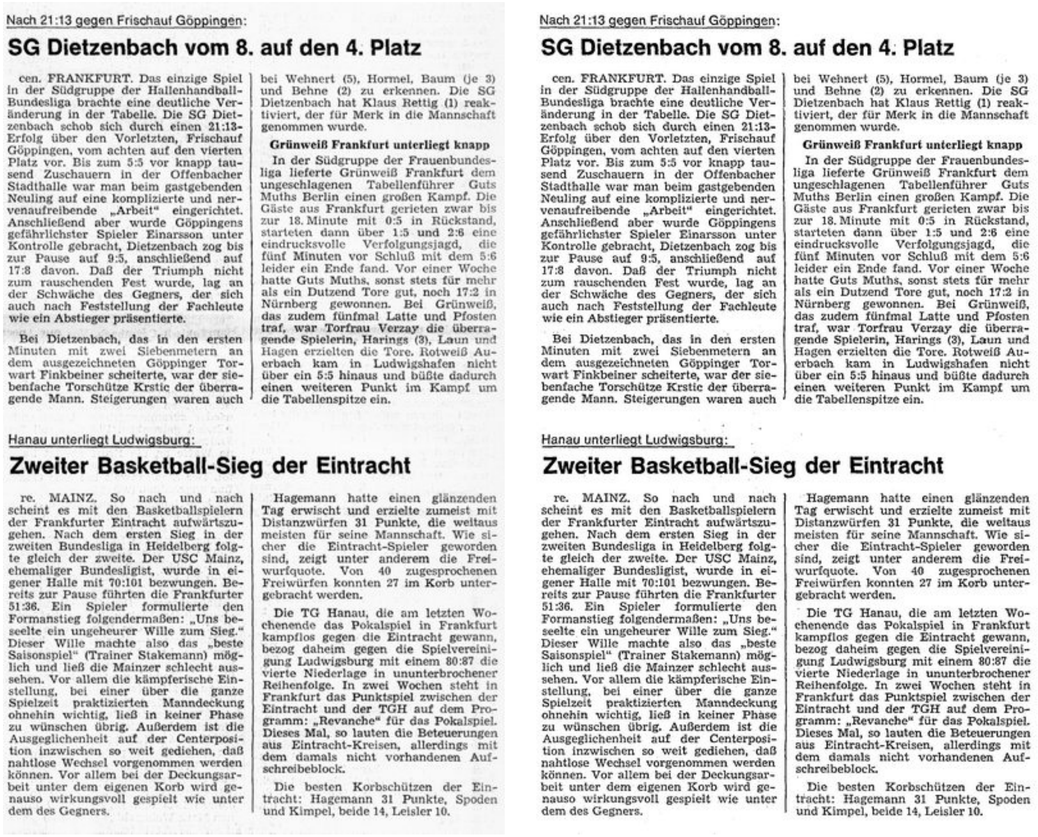
$$\lambda = \frac{\sigma_B^2}{\sigma_W^2}\ ,\ \kappa = \frac{\sigma_T^2}{\sigma_W^2} \text{ and } \eta = \frac{\sigma_B^2}{\sigma_T^2},\ \text{where}$$

$$\sigma_W^2 = \omega_0 \sigma_0^2 + \omega_1 \sigma_1^2$$
$$\sigma_B^2 = \omega_0(\mu_0 - \mu_T)^2 + \omega_1(\mu_1 - \mu_T)^2 = \omega_0\omega_1(\mu_1 - \mu_0)^2$$
$$\sigma_T^2 = \sum_{i=1}^{L}(i - \mu_T)^2$$

The goodness of fit criteria λ, κ and η are related by: $\kappa = \lambda + 1$ and $\eta = \frac{\lambda}{\lambda+1}$, due to the fact that $\sigma_T^2 = \sigma_W^2 + \sigma_B^2$. Typically, the criterion chosen to be optimized is η, meaning that we search for the value k^* as follows:

$$k^* = \arg\max_{1 \le k \le L} \sigma_B^2(k)$$

Given the previous formulae, the practical implementation of Otsu's algorithm is straightforward. It is worth mentioning that, given the histogram of an image, the optimal threshold value can be computed in just one pass over the histogram data. Of course, as the histogram size is independent from the dimension of the input image, the resulting algorithm is extremely fast in practice.



Nach 21:13 gegen Frischauf Göppingen:

SG Dietzenbach vom 8. auf den 4. Platz

cen. FRANKFURT. Das einzige Spiel in der Südgruppe der Hallenhandball-Bundesliga brachte eine deutliche Veränderung in der Tabelle. Die SG Dietzenbach schob sich durch einen 21:13-Erfolg über den Vorletzten, Frischauf Göppingen, vom achten auf den vierten Platz vor. Bis zum 5:5 vor knapp tausend Zuschauern in der Offenbacher Stadthalle war man beim gastgebenden Neuling auf eine komplizierte und nervenaufreibende „Arbeit" eingerichtet. Anschließend aber wurde Göppingens gefährlichster Spieler Einarsson unter Kontrolle gebracht, Dietzenbach zog bis zur Pause auf 9:5, anschließend auf 17:8 davon. Daß der Triumph nicht zum rauschenden Fest wurde, lag an der Schwäche des Gegners, der sich auch nach Feststellung der Fachleute wie ein Absteiger präsentierte.

Bei Dietzenbach, das in den ersten Minuten mit zwei Siebenmetern an dem ausgezeichneten Göppinger Torwart Finkbeiner scheiterte, war der siebenfache Torschütze Krstic der überragende Mann. Steigerungen waren auch bei Wehnert (5), Hormel, Baum (je 3) und Behne (2) zu erkennen. Die SG Dietzenbach hat Klaus Rettig (1) reaktiviert, der für Merk in die Mannschaft genommen wurde.

Grünweiß Frankfurt unterliegt knapp

In der Südgruppe der Frauenbundesliga lieferte Grünweiß Frankfurt dem ungeschlagenen Tabellenführer Guts Muths Berlin einen großen Kampf. Die Gäste aus Frankfurt gerieten zwar bis zur 18. Minute mit 0:5 in Rückstand, starteten dann über 1:5 und 2:6 eine eindrucksvolle Verfolgungsjagd, die fünf Minuten vor Schluß mit dem 5:6 leider ein Ende fand. Vor einer Woche hatte Guts Muths, sonst stets für mehr als ein Dutzend Tore gut, noch 17:2 in Nürnberg gewonnen. Bei Grünweiß, das zudem fünfmal Latte und Pfosten traf, war Torfrau Verzay die überragende Spielerin, Harings (3), Laun und Hagen erzielten die Tore. Rotweiß Auerbach kam in Ludwigshafen nicht über ein 5:5 hinaus und büßte dadurch einen weiteren Punkt im Kampf um die Tabellenspitze ein.

Hanau unterliegt Ludwigsburg:

Zweiter Basketball-Sieg der Eintracht

re. MAINZ. So nach und nach scheint es mit den Basketballspielern der Frankfurter Eintracht aufwärtszugehen. Nach dem ersten Sieg in der zweiten Bundesliga in Heidelberg folgte gleich der zweite. Der USC Mainz, ehemaliger Bundesligist, wurde in eigener Halle mit 70:101 bezwungen. Bereits zur Pause führten die Frankfurter 51:36. Ein Spieler formulierte den Formanstieg folgendermaßen: „Uns beseelte ein ungeheurer Wille zum Sieg." Dieser Wille machte also das „beste Saisonspiel" (Trainer Stakemann) möglich und ließ die Mainzer schlecht aussehen. Vor allem die kämpferische Einstellung, bei einer über die ganze Spielzeit praktizierten Manndeckung ohnehin wichtig, ließ in keiner Phase zu wünschen übrig. Außerdem ist die Ausgeglichenheit auf der Centerposition inzwischen so weit gediehen, daß nahtlose Wechsel vorgenommen werden können. Vor allem bei der Deckungsarbeit unter dem eigenen Korb wird genauso wirkungsvoll gespielt wie unter dem des Gegners.

Hagemann hatte einen glänzenden Tag erwischt und erzielte zumeist mit Distanzwürfen 31 Punkte, die weitaus meisten für seine Mannschaft. Wie sicher die Eintracht-Spieler geworden sind, zeigt unter anderem die Freiwurfquote. Von 40 zugesprochenen Freiwürfen konnten 27 im Korb untergebracht werden.

Die TG Hanau, die am letzten Wochenende das Pokalspiel in Frankfurt kampflos gegen die Eintracht gewann, bezog daheim gegen die Spielvereinigung Ludwigsburg mit einem 80:87 die vierte Niederlage in ununterbrochener Reihenfolge. In zwei Wochen steht in Frankfurt das Punktspiel zwischen der Eintracht und der TGH auf dem Programm: „Revanche" für das Pokalspiel. Dieses Mal, so lauten die Beteuerungen aus Eintracht-Kreisen, allerdings mit dem damals nicht vorhandenen Aufschreibeblock.

Die besten Korbschützen der Eintracht: Hagemann 31 Punkte, Spoden und Kimpel, beide 14, Leisler 10.

Nach 21:13 gegen Frischauf Göppingen:

SG Dietzenbach vom 8. auf den 4. Platz

cen. FRANKFURT. Das einzige Spiel in der Südgruppe der Hallenhandball-Bundesliga brachte eine deutliche Veränderung in der Tabelle. Die SG Dietzenbach schob sich durch einen 21:13-Erfolg über den Vorletzten, Frischauf Göppingen, vom achten auf den vierten Platz vor. Bis zum 5:5 vor knapp tausend Zuschauern in der Offenbacher Stadthalle war man beim gastgebenden Neuling auf eine komplizierte und nervenaufreibende „Arbeit" eingerichtet. Anschließend aber wurde Göppingens gefährlichster Spieler Einarsson unter Kontrolle gebracht, Dietzenbach zog bis zur Pause auf 9:5, anschließend auf 17:8 davon. Daß der Triumph nicht zum rauschenden Fest wurde, lag an der Schwäche des Gegners, der sich auch nach Feststellung der Fachleute wie ein Absteiger präsentierte.

Bei Dietzenbach, das in den ersten Minuten mit zwei Siebenmetern an dem ausgezeichneten Göppinger Torwart Finkbeiner scheiterte, war der siebenfache Torschütze Krstic der überragende Mann. Steigerungen waren auch bei Wehnert (5), Hormel, Baum (je 3) und Behne (2) zu erkennen. Die SG Dietzenbach hat Klaus Rettig (1) reaktiviert, der für Merk in die Mannschaft genommen wurde.

Grünweiß Frankfurt unterliegt knapp

In der Südgruppe der Frauenbundesliga lieferte Grünweiß Frankfurt dem ungeschlagenen Tabellenführer Guts Muths Berlin einen großen Kampf. Die Gäste aus Frankfurt gerieten zwar bis zur 18. Minute mit 0:5 in Rückstand, starteten dann über 1:5 und 2:6 eine eindrucksvolle Verfolgungsjagd, die fünf Minuten vor Schluß mit dem 5:6 leider ein Ende fand. Vor einer Woche hatte Guts Muths, sonst stets für mehr als ein Dutzend Tore gut, noch 17:2 in Nürnberg gewonnen. Bei Grünweiß, das zudem fünfmal Latte und Pfosten traf, war Torfrau Verzay die überragende Spielerin, Harings (3), Laun und Hagen erzielten die Tore. Rotweiß Auerbach kam in Ludwigshafen nicht über ein 5:5 hinaus und büßte dadurch einen weiteren Punkt im Kampf um die Tabellenspitze ein.

Hanau unterliegt Ludwigsburg:

Zweiter Basketball-Sieg der Eintracht

re. MAINZ. So nach und nach scheint es mit den Basketballspielern der Frankfurter Eintracht aufwärtszugehen. Nach dem ersten Sieg in der zweiten Bundesliga in Heidelberg folgte gleich der zweite. Der USC Mainz, ehemaliger Bundesligist, wurde in eigener Halle mit 70:101 bezwungen. Bereits zur Pause führten die Frankfurter 51:36. Ein Spieler formulierte den Formanstieg folgendermaßen: „Uns beseelte ein ungeheurer Wille zum Sieg." Dieser Wille machte also das „beste Saisonspiel" (Trainer Stakemann) möglich und ließ die Mainzer schlecht aussehen. Vor allem die kämpferische Einstellung, bei einer über die ganze Spielzeit praktizierten Manndeckung ohnehin wichtig, ließ in keiner Phase zu wünschen übrig. Außerdem ist die Ausgeglichenheit auf der Centerposition inzwischen so weit gediehen, daß nahtlose Wechsel vorgenommen werden können. Vor allem bei der Deckungsarbeit unter dem eigenen Korb wird genauso wirkungsvoll gespielt wie unter dem des Gegners.

Hagemann hatte einen glänzenden Tag erwischt und erzielte zumeist mit Distanzwürfen 31 Punkte, die weitaus meisten für seine Mannschaft. Wie sicher die Eintracht-Spieler geworden sind, zeigt unter anderem die Freiwurfquote. Von 40 zugesprochenen Freiwürfen konnten 27 im Korb untergebracht werden.

Die TG Hanau, die am letzten Wochenende das Pokalspiel in Frankfurt kampflos gegen die Eintracht gewann, bezog daheim gegen die Spielvereinigung Ludwigsburg mit einem 80:87 die vierte Niederlage in ununterbrochener Reihenfolge. In zwei Wochen steht in Frankfurt das Punktspiel zwischen der Eintracht und der TGH auf dem Programm: „Revanche" für das Pokalspiel. Dieses Mal, so lauten die Beteuerungen aus Eintracht-Kreisen, allerdings mit dem damals nicht vorhandenen Aufschreibeblock.

Die besten Korbschützen der Eintracht: Hagemann 31 Punkte, Spoden und Kimpel, beide 14, Leisler 10.

Figure 3.3: Sample binarization result obtained by using Otsu's algorithm [204] on a clean document image. Left: Original grayscale newspaper scan; Right: the resulting binary version

Finally, it must be noted that optimal global thresholding has a very good performance in documents where there is a clear separation between the foreground and the background. Unfortunately, as we have already seen, document images are usually exposed to degradations that weaken any guarantee for such a separation.

3.1.2 Adaptive Methods

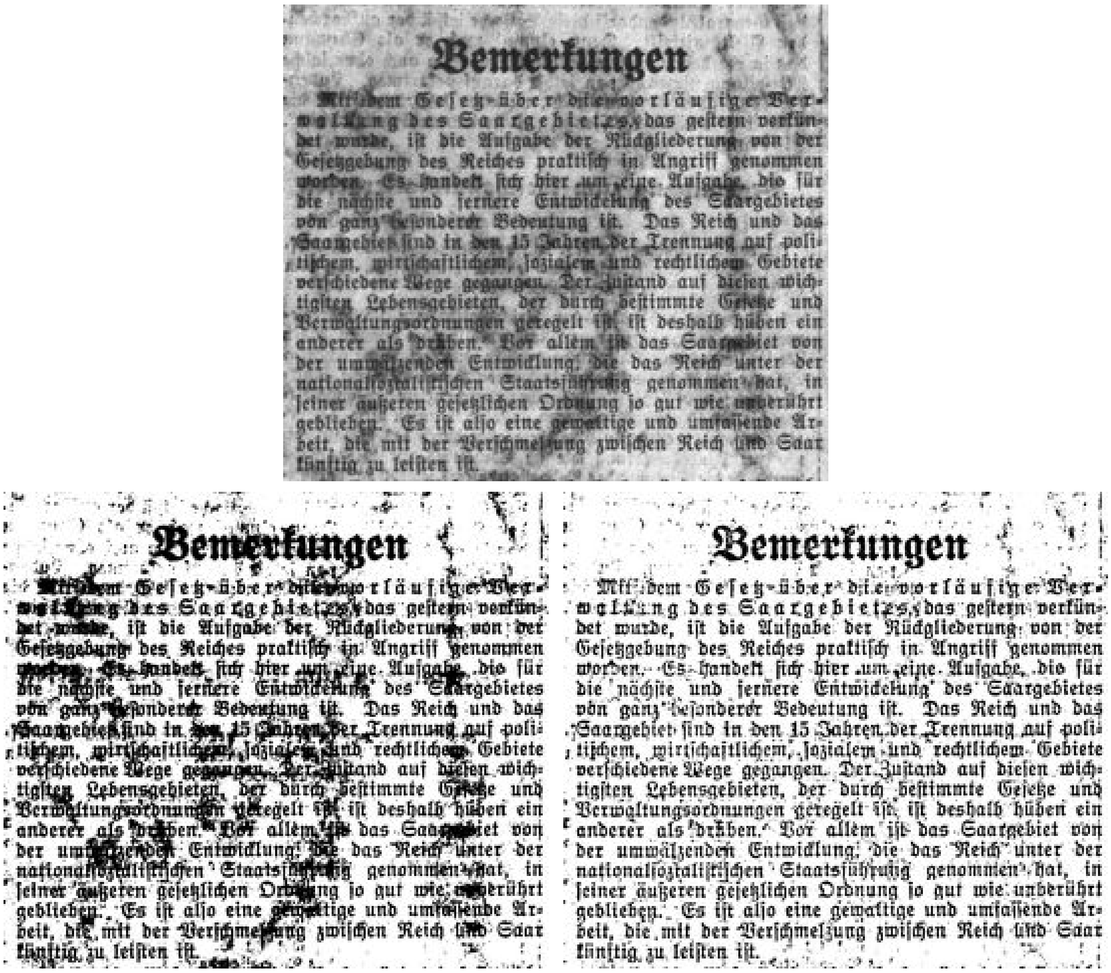
Adaptive binarization techniques have become the standard go-to methods in document image analysis due to their excellent performance on document images exhibiting visible distortions, such as strong lighting changes or noise. As previously mentioned, adaptive (local) binarization methods use different threshold values selected according to the local area information.

A possible approach for obtaining the local thresholds is to divide the input image into subimages and compute one threshold per subimage. Subsequently, the threshold is used to binarize the corresponding subimage. For computing the threshold for each subimage, one may employ any global thresholding method. Two of the earliest adaptive binarization methods are described in the book of Gonzales and Woods [119], pp. 600–607. The first of them divides the page into non-overlapping rectangular subimages and for each of them computes a threshold by assuming the bimodality of the histogram. The second method was introduced by Chow and Kaneko [77] as early as 1972. Their algorithm divides the image into windows having 50% overlap and computes thresholds only for those subimages where the histogram is indeed bimodal. The thresholds for the remaining subimages, where the assumption of histogram bimodality does not hold, are then interpolated from the previously computed thresholds.

Another approach for adaptive thresholding is to choose an arbitrary number of pixels from the input image and compute a threshold corresponding to each of them. The thresholds are computed by using the histograms of the windows of a given radius centered on the respective pixel. Finally, the thresholds for all other pixels in the image are interpolated from the regular or irregular grid formed by the initial set of pixels. Due to the increased processing speed of today's computers, interpolation is seldom needed; instead a threshold is directly computed for each pixel in the input image. The first such method was suggested by Niblack [192]. Later it was generalized and improved specifically for document images by Sauvola et al. [230], and subsequently adapted for use in text extraction from videos by Wolf et al. [285]. The approaches of Niblack and Sauvola et al. are well-known and thoroughly tested in the context of document image analysis, both performing consistently well in several surveys and comparisons [116, 230, 269, 288].

The algorithm of Niblack computes the threshold T for a pixel using the mean m and the variance s of the gray values in the window, as in the following formula: $T = m + ks$. The value k and the dimensions of the local window ($M \times N$) are only necessary parameters. Good results were observed with $k = -0.2$ and $M = N$, selected in such way that the local windows have at least the size of one complete text character. The Niblack method can distinguish objects from background effectively in the areas close to the objects; however it has the tendency to produce lots of small noise in image areas where objects are sparse. As suggested by some authors [269], the remaining noise can be removed in a post-processing step, yielding a visually better result. According to the comparison performed by Trier and Jain [269], the most suitable noise removal algorithm following the application of the Niblack method is that of Yanowitz and Bruckstein [297]. The other possible way of obtaining a noise-free image is not to generate noise in the first place. This can be accomplished by improving the quality of the computed per-pixel threshold values. Following this train of thought, Sauvola et al. [230] and subsequently Wolf et al. [285] proposed such enhancement for the computation of the local thresholds.

Note that in their original paper [230] Sauvola et al. propose a compound binarization algorithm, actually consisting of two different binarization methods. One binarization



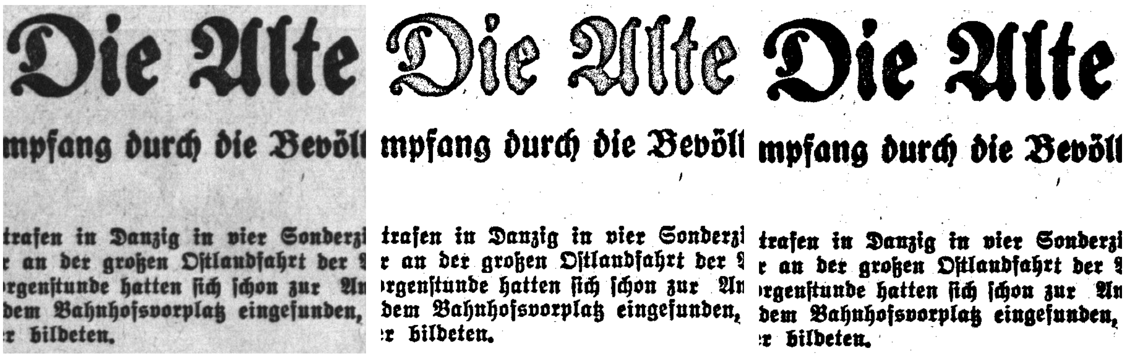
Bemerkungen

Mit dem Geſetz über die vorläufige Verwaltung des Saargebietes, das geſtern verkündet wurde, iſt die Aufgabe der Rückgliederung von der Geſetzgebung des Reiches praktiſch in Angriff genommen worden. Es handelt ſich hier um eine Aufgabe, die für die nächſte und fernere Entwickelung des Saargebietes von ganz beſonderer Bedeutung iſt. Das Reich und das Saargebiet ſind in den 15 Jahren der Trennung auf politiſchem, wirtſchaftlichem, ſozialem und rechtlichem Gebiete verſchiedene Wege gegangen. Der Zuſtand auf dieſen wichtigſten Lebensgebieten, der durch beſtimmte Geſetze und Verwaltungsordnungen geregelt iſt, iſt deshalb hüben ein anderer als drüben. Vor allem iſt das Saargebiet von der umwälzenden Entwicklung, die das Reich unter der nationalſozialiſtiſchen Staatsführung genommen hat, in ſeiner äußeren geſetzlichen Ordnung ſo gut wie unberührt geblieben. Es iſt alſo eine gewaltige und umfaſſende Arbeit, die mit der Verſchmelzung zwiſchen Reich und Saar künftig zu leiſten iſt.

Figure 3.4: Top: Section of a grayscale document image showing significant degradation effects; Bottom: results of applying two different binarization algorithms – Otsu's optimal global method [204] (left) and the Sauvola et al. [230] adaptive method (right)

method is applied in case of text, while the other in case of images/graphics, depending on the output of a switching algorithm. Following the tradition of most papers concerning binarization from document image analysis [116, 285, 288], we shall only be concerned with the binarization algorithm targeted at text regions. The algorithm of Sauvola et al. builds directly on the previous work of Niblack [192] and tries to avoid the necessity of applying costly post-processing procedures by computing a more exact threshold value. The threshold for each pixel is calculated by using the following formula: $T = m(1 - k(1 - \frac{s}{R}))$, where m, s and k have the same significance as in Niblack's algorithm and R represents the dynamic range of the standard deviation within the given image. As noted by Sauvola et al. [230], the use of the local mean value to multiply both terms has the effect of adaptively amplifying the contribution of the standard deviation. This is has a beneficial effect in many situations, such as text regions located within stains. Wolf et al. [285] observe that the aforementioned "trick" is basically equivalent to the assumption that gray values for text are usually lower than those of the background. In the experiments performed by Sauvola et al. on a collection of greyscale document images, the parameters k and R were fixed to the values 0.5 and 128, respectively. More importantly, the algorithm is not overly sensitive to the value of k, fact confirmed by the successful experiments performed with $k = 0.2$ by Gatos et al [116].

The versatility of Sauvola's algorithm was proved by Wolf et al. [285], who employ a slightly improved version of the algorithm for text detection in video documents. Based on the observation that video frames do not always have a full dynamic range of greyscale values, Wolf et al. changed the formula for computing the per-pixel threshold in order to normalize the contrast and the mean gray level of the image. The proposed formula is: $T = m - k\alpha(m - M)$, where $\alpha = 1 - \frac{s}{R}$, and $R = \max(s)$. Here, M represents the minimum gray level in the image and R is set to the maximum of the standard deviations of all windows. Notice that computing the new formula now requires two passes on the image data, as the value R needs to be computed prior to the determination of the local thresholds. A fast method of computing the necessary local means and thresholds by using integral images was introduced in the Master's thesis of Konya [153] in 2006 and later by Shafait et al. [241] in 2008.



Die Alte

mpfang durch die Bevöll

trafen in Danzig in vier Sonderzi
r an der großen Oſtlandfahrt der
rgenſtunde hatten ſich ſchon zur An
dem Bahnhofsvorplatz eingefunden,
r bildeten.

Figure 3.5: A 585 × 561 grayscale image (left side) and results produced by the algorithm of Sauvola et al. with a window diameter of 19 pixels (center) and 85 pixels (right side). Note the hollowed out large characters produced when using a small window radius in comparison to the character size

More recently, a very different algorithm was proposed by Kim et al. [149]. In their adaptive method, the image surface is regarded as a 3-dimensional terrain whose specific local properties are extracted by using a water flow model. As a first step, a certain amount of water is poured uniformly onto the terrain. The water will flow down to the lower regions, thus flooding deep valleys, while higher areas and smooth plain regions remain dry. Finally, the amount of filled water (i.e. the difference between the height of the original and the flooded terrain) is thresholded using a global binarization method, such as that of Otsu [204]. A shortcoming of this method is the necessity of manual selection of two critical parameters, namely the amount of rainfall and the local water-flow mask size. However, in general the resulting images are of good quality [116] and automatic parameter estimation has also been the subject of recent papers [199].

Other notable approaches are that of Papamarkos [209] and Gatos et al. [116]. Papamarkos trains a Kohonen Self-Organized Feature Map neural network classifier, whose outputs, interpreted as fuzzy membership functions are then fed into the Fuzzy C-Means algorithm to generate the final binarization. Gatos et al. present a more complex, 5-step approach specifically tuned for the binarization and enhancement of degraded documents. The steps are, in order: pre-processing using a low-pass Wiener filter, a rough estimation of foreground regions (using the algorithm of Sauvola et al. [230]), background surface calculation, thresholding by combining the calculated background surface with the original image and concludes with a post-processing step in order to improve the quality of text regions and preserve stroke connectivity.

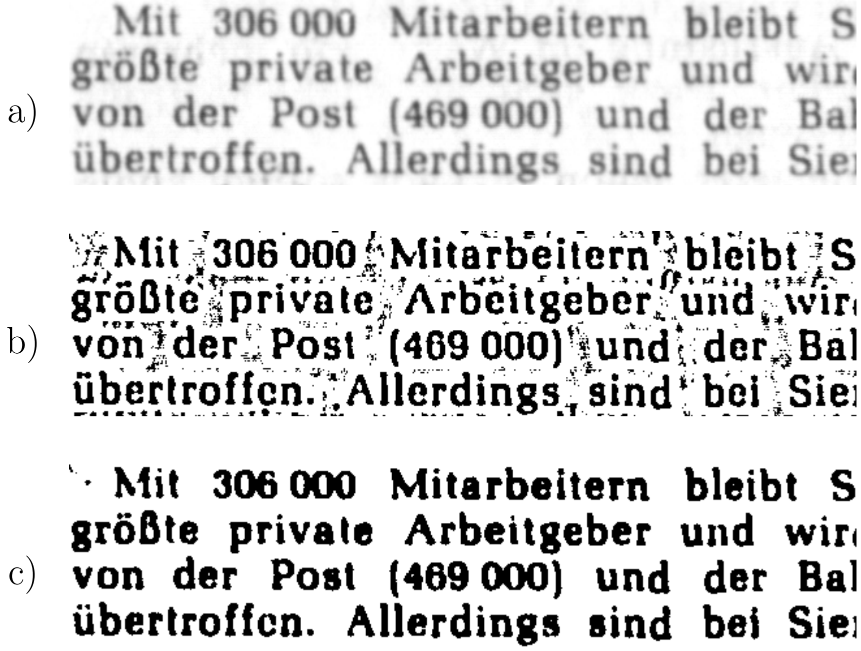
Concluding the overview section, we briefly go over the winning algorithms of the two most recent document image binarization contests. The best performing algorithm in the DIBCO 2009 [115] was proposed by Lu and Tan of the Institute for Infocomm Research, Singapore. Their algorithm consists of four parts: document background extraction, stroke edge detection, local thresholding, and post-processing. Local thresholds are estimated by averaging the detected edge pixels within the local neighborhood windows. In DIBCO 2011 [218], the algorithm of Lelore and Bouchara of the South University of Toulon-Var, France placed at the top of the 18 competing methods. The method starts with a median filtering of the input image, which is then upscaled via linear interpolation. A noisy version of the final result is produced using a local method. The text pixels from this image are then mixed into another temporary three-valued image (classes corresponding to text, background and other), obtained using correct threshold estimation. Noise is then filtered to produce the final image which is downscaled back to the original resolution using bicubic interpolation. The binary result is obtained using a global threshold set to 70. It is interesting to note that both methods rely heavily on noise filtering and are specially tuned to produce good results for black-on-white text and background areas.

3.1.3 Constant-Time Adaptive Binarization Framework

We have seen from the previous sections that there exist global binarization algorithms capable of producing optimal classifications with respect to certain pre-defined separability criteria. The disadvantage of global algorithms is their inherent inability to adapt to local variations in contrast and luminance. In contrast, local binarization algorithms perform well for such variations, but traditionally rely on a wide variety of heuristics for obtaining the local thresholds. The goal we follow in the current section is the combination of optimal thresholding with the local character of the adaptive methods. In the context of local thresholding, the reduction of computational complexity plays a paramount role and as such shall also be deemed as one of our main goals.

In this section we propose a generic binarization framework which effectively transforms any global thresholding algorithm operating on histogram data into an adaptive version of itself. The proposed algorithm runs in a time independent of the selected local window size. Furthermore, it is straightforward to implement, uses little additional memory and requires just two parameters, the sizes of the local windows. In order to show the generality of the proposed method, the framework was used to implement both the globally optimal algorithm of Otsu [204] and an adaptive version of the Kittler and Illingworth [151] global thresholding algorithm as the best performing technique in the recent comparison by Sezgin and Sankur [237]. Subsequently we describe an automatic, constant-time method for estimating the required parameters from a given document image. An evaluation of the run-time performance of the proposed method against a standard implementation, as well as example result images are contained in section 3.1.4.

The basic idea for the proposed algorithm is straightforward: inspired by the success of Otsu's thresholding method on full document images, use the same technique for thresholding local windows. At his point it is worth noting that as seen in section 3.1.1 the global Otsu method computes an optimal threshold, i.e. one which maximizes a criterion related to the grey-level variance between objects and background. Thus, all local thresholds will also be optimal in that sense. As input we consider to have the greyscale image and the radius r of the centered local window to use for each pixel. From the local histogram, one can then readily compute the optimal threshold according to Otsu's criterion in 2



a)

Mit 306 000 Mitarbeitern bleibt S
größte private Arbeitgeber und wir
von der Post (469 000) und der Bal
übertroffen. Allerdings sind bei Sie

b)

Mit 306 000 Mitarbeitern bleibt S
größte private Arbeitgeber und wir
von der Post (469 000) und der Bal
übertroffcn. Allerdings sind bei Sie

c)

Mit 306 000 Mitarbeitern bleibt S
größte private Arbeitgeber und wir
von der Post (469 000) und der Bal
übertroffcn. Allerdings sind bei Sie

Figure 3.6: a) Grayscale document image of size 605 × 142; b), c) its binarization using a window size of 11 and 33 pixels, respectively. Note the significant amount of noise introduced in background areas when using a too small window size.

passes over the histogram. Since the highest number of values investigated is equal to the histogram size (which is constant), it follows that this step can be performed in a time independent of the local window size r.

Unfortunately, the Otsu thresholding method, like all other global methods, is based on the assumption that within the investigated window there always exist exactly two classes (i.e. both background and foreground). It is clear that this does not hold for small, local windows, as they may well be contained entirely within the background or the foreground. Consequently, wrong thresholds and thus unwanted artifacts are produced by the algorithm for all such local windows. An example of this can be seen in figure 3.6. The problem can be solved by using a window size which is large enough to always contain both background and foreground. The results in this case are correct, but the use of large window sizes effectively cancels the “local” character of the method as well as the ability to properly adapt to a non-even illumination/contrast. Therefore, we propose using two centered, local windows – a small one for exploiting local information and a larger one which is guaranteed to always contain both background and foreground pixels. Thus, by applying Otsu’s thresholding method on the weighted sum of the two local histograms one can determine an optimal threshold which makes use of both local and global information. In all our experiments, the weighting factor for the histogram of the small window was set to be equal to $Area(LargeWindow)/Area(SmallWindow)$, whereas the weight for the larger window histogram was set to 1. Note that this particular choice of weights is practically equivalent to (a scaled version of) the sum of the two normalized histograms. The histogram weighting can be adapted in order to seamlessly transition between a highly local character to a global one. An automatic adaptation method for the weighting remains as topic for future work.

In order to dramatically improve the run-time performance of the algorithm we build upon

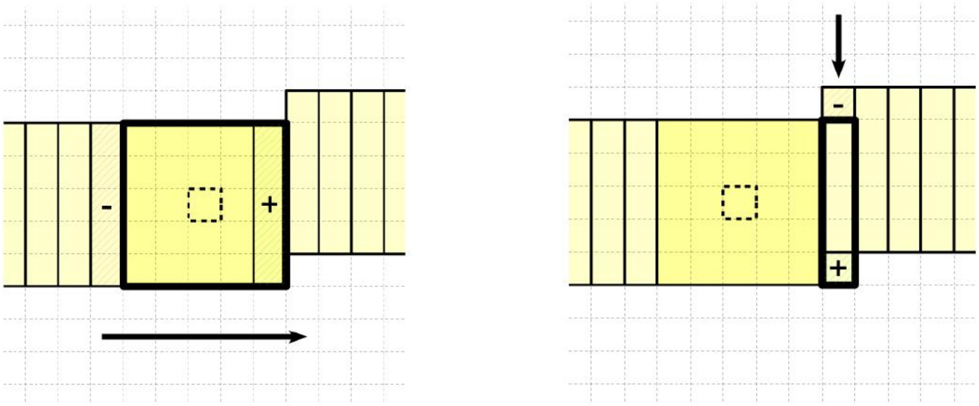


Figure 3.7: Left: local histogram update for horizontal scan; Right: histogram update for vertical scan [214]

the method for performing median filtering in amortized constant time recently introduced by Perreault and Hebert [214]. The median filtering algorithm needs to maintain one histogram for each column in the image. Each column histogram accumulates the vertical stripe of $2r+1$ pixels centered on the image row being processed. The algorithm also needs to maintain the histogram of the local window (i.e. the window centered on the pixel for which the threshold is to be determined). Initially, the local histogram can be computed by summing up the $2r+1$ column histograms centered on the start pixel. Afterward, assuming the image traversal is done left-to-right, the local histogram can be computed for the pixel $i+1$ by subtracting the column histogram $i-r$ and adding the column histogram $i+1+r$. Note that the number of operations required by the addition and subtraction of histograms is independent of the local window size and is constant. When moving from one row to the next one under it, each of the column histograms is updated by subtracting the topmost pixel value inside it and adding the pixel value in the image directly below it. This is obviously also a constant time process. However, as proposed in the original algorithm, at the beginning of each row the local histogram needs to be initialized again with the sum of the nearby column histograms. This means that a penalty of $O(r)$ histogram additions is incurred for each row of the input image. The penalty can be reduced by alternating left-to-right and right-to-left scans of the image. In this way, at the end of each row the local histogram just needs to be moved one row down – all other pixels within it are unchanged. Updating of the local histogram can thus be accomplished with just r pixel additions and r pixel subtractions, a cost significantly lower than the r histogram additions otherwise necessary. A pseudocode description of the final version of the proposed algorithm (using two local windows) can be found at the end of the current section.

Algorithm 1 Proposed binarization algorithm

```
Require: Image $X$ of size $m \times n$, local window radii $r_1 < r_2$
Ensure: Image $Y$ of the same size
  Initialize window histograms $Hs$, $Hl$ and column histograms $hs_{1 \ldots n}$, $hl_{1 \ldots n}$
  $K \leftarrow r_2^2/r_1^2$
  for $i = 1$ to $m$ do
    if $i$ is odd then
      for $j = 1$ to $r_1$ do
        Remove $X_{i-r_1-1,j+r_1}$ from $Hs$
        Add $X_{i+r_1,j+r_1}$ to $Hs$
      end for
      for $j = 1$ to $r_2$ do
        Remove $X_{i-r_2-1,j+r_2}$ from $Hl$
        Add $X_{i+r_2,j+r_2}$ to $Hl$
      end for
      for $j = 1$ to $n$ do
        Remove $X_{i-r_1-1,j+r_1}$ from $hs_{j+r_1}$
        Add $X_{i+r_1,j+r_1}$ to $hs_{j+r_1}$
        Remove $X_{i-r_2-1,j+r_2}$ from $hl_{j+r_2}$
        Add $X_{i+r_2,j+r_2}$ to $hl_{j+r_2}$
        $Hs \leftarrow Hs + hs_{j+r_1} - hs_{j-r_1-1}$
        $Hl \leftarrow Hl + hl_{j+r_2} - hl_{j-r_2-1}$
        $Y_{i,j} \leftarrow OtsuThreshold(Hl + Hs \times K)$
      end for
    else
      same steps, this time traversing and updating from right to left
    end if
  end for
```

We have performed experiments with a version of the above algorithm which uses a single local window of the same width as the document image (dubbed row-adaptive). For the selection of the height of the window one must also ensure that it is always tall enough to contain both background and object pixels. The necessary height can be computed automatically by a simple change in the second algorithm presented in the following section. Furthermore, using the same framework we have been able to transform the Kittler and Illingworth [151] minimum error global thresholding method into a constant-time adaptive algorithm. More details may be found in section 3.1.4.

3.1.3.1 Automatic Parameter Estimation

This section contains the description of two distinct algorithms for automatically detecting the size of the local windows. Throughout this section we will assume that a preliminary binarized version of the input image is already available and that all its connected components have been determined. Note that the considered binary image need not have a good quality, and it is sufficient to create it by global binarization [204] or by employing the fast row-adaptive binarization method mentioned at the end of section 3.1.3. Many connected component labeling algorithms which run in $O(N_{pix})$ time and have negligible additional memory requirements can be found in the literature, e.g. [90].

For estimating the size of the smaller local window we use the observation that local algorithms such as those proposed by Niblack [192] and Sauvola et al. [229] perform best when the local window covers about 1–2 characters. Note that in the specialized literature, there already exist methods for estimating the average character size of a given document image. Gatos et al. [112] estimate the average character height by sampling the image pixels randomly, following the contour of the containing connected component and finally computing an average of the components' heights. Zheng et al. [302] compute the average character width and height in a document by taking the highest peak in the histogram of connected component widths and heights, respectively. The drawback of these methods is that they generally fail on documents which exhibit one or more of the following characteristics: the page contains less text than halftones or drawings, there exist multiple font sizes and there exists significant noise in the image.

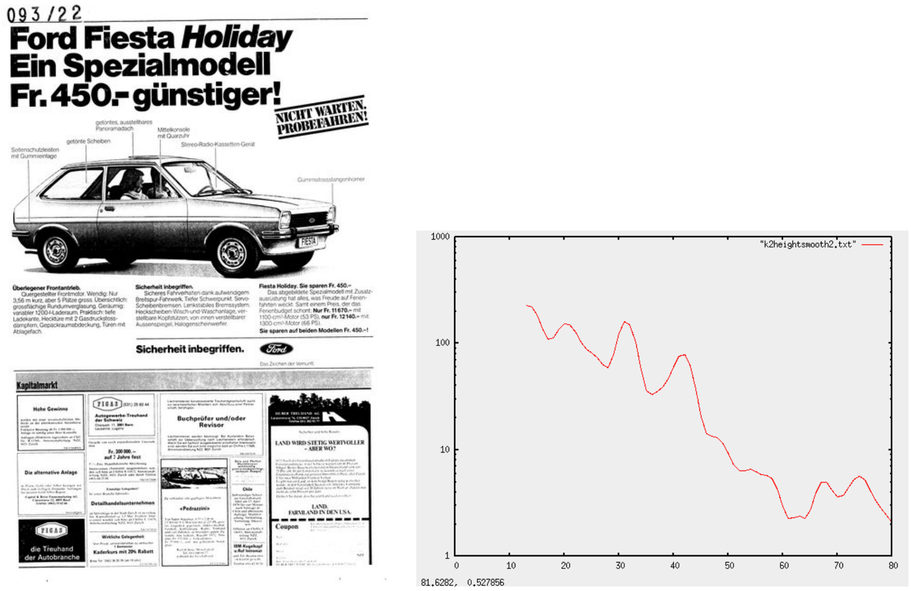


Figure 3.8: Document image with multiple font sizes and its corresponding smoothed histogram of connected component heights

Our method attempts to partially cope with such situations. In order for the local binarization to perform best on text, it is advantageous to use as reference for the dimensions of the window the character size of the dominant font (i.e. the font in which the most characters are typed), not simply the average character size on the page. For pages containing multiple font sizes, the average character size will most likely be different from that of the dominant font, thus using it will lead to sub-optimal binarization results for the dominant font. Another useful observation is that for printed texts, the character height for a specific font size fluctuates much less than its character width, and thus it can generally be estimated more robustly. Finally, since in the majority of cases the scanning resolution of the document is known to the binarization algorithm, one may optionally ignore all connected components with heights smaller than a certain value V. In our experiments, we have used $V = 0.09cm$, motivated by the fact that printed characters smaller than this are generally not legible by humans. In case the scanning resolution is unknown, the value V can be set to e.g. 5 pixels, as any text smaller than this will not be legible either by humans or by optical character recognition systems. As input data, we take a list containing the bounding boxes of all connected components on the document page and optionally, the scanning resolution used. The algorithm operates in two passes: first, a smoothed histogram is created from the heights of all given connected components. Subsequently, a first approximation of the dominant height is computed by searching for a high and flat peak starting from the value V (in pixels). The peak is determined by searching the histogram for the bin i where the following two conditions are met:

- $H_i \geq H_{i-1}$ and $H_i \geq H_{i+1}$ (peak condition)
- $\frac{1+log(H_i - H_{i-1}+1+pen(i))+log(H_i - H_{i+1}+1+pen(i))}{1+\sum_{k=i-1}^{i+1} H_k}$ is minimized (flatness condition)

$pen(i)$ represents a monotonically increasing function having the purpose of penalizing larger values for the dominant font size. The penalty function is necessary due to the fact that if a page features multiple fonts (e.g. advertisements vs. editorial content), the smaller font must be selected as the window size, unless the number of characters in the higher-sized font is significantly larger. Selecting a larger window size is in general undesirable, because it leads to a loss in the locality property (the local thresholds will converge to the global one when using a window size as large as the input image). In our experiments we have used $pen(i) = i$ with good results. After finding the bin index i, a new smoothed histogram is created by using only the connected components having a height greater than or equal to i. The new histogram is again subject to the search for the peak satisfying the aforementioned conditions and the obtained dominant height (i.e. bin index) is taken to be the desired window size. We have experimentally determined that the above method provides a robust estimate for the dominant character height in the majority of cases when there exist at least 1–2 lines of text on the page, regardless of the presence of other non-text content.

The second algorithm of this section computes an estimate for the size of the larger local windows used in the proposed binarization method. As mentioned in section 3.1.3, the larger local window needs to always contain both background and foreground pixels in order for the Otsu thresholding method to work properly. At this point, it is interesting to note that Huang et al. [133] recently proposed the automatic estimation of the local window size based on the Lorentz information measure. This is unfortunately not applicable in our situation, since we need a hard guarantee for the existence of both types of pixels within the window. For reaching our goal, we employ a constant-time approximate Euclidean distance transform (EDT) algorithm [84] on the inverted binary image from which all connected components with height lower than the small window size have been erased. Afterward, the size of the larger local window is set to be equal to one plus the maximum value in the distance transformed image. Note that there exist exact EDT algorithms running in $O(N_{pix})$ time [181], however their hidden constant is generally much higher than the one in approximate algorithms. The approximate algorithm of Danielsson [84], based on vector propagation, has the advantage of being straightforward to implement and runs very fast in practice. For our purpose, the approximation provided by this algorithm is sufficient. Interested readers can find a detailed analysis of the magnitude of the errors in [82]. Note that a straightforward way of skipping the computation of the large window radius would be the use of the whole input image instead (with the proper adjustments for the histogram weights). In this case, our framework degenerates into a pure hybrid method, which would in theory still have the advantage of never performing worse than either local or global methods.

3.1.4 Experiments

We have tested the runtime speed of the proposed method against that of a straightforward implementation of the same algorithm on a typical document image of size 2350 × 3500 (approx. DIN-A4 scanned at 300 dpi). All experiments were performed on a computer equipped with a Core2Duo 2 GHz processor, and for each windows size the algorithms were run 5 times (the average value was plotted). Note that we have tested the run-time performance of the single-window approach, not that of the 2-window method. This is due to the fact that the relative performance would likely remain the same (two histograms would have to be updated instead of a single one) and because we wanted to include in the graph the row-adaptive approach, which only uses a single window size. From the plot one can clearly see the performance gain growing proportionally to the window size. The speed of the row-adaptive binarization is very high and directly comparable to that of a global algorithm. Whereas for a window of size 11 both other algorithms take about 23 seconds, for a window of size 251 the proposed algorithm performs more than 30× faster than the regular implementation. On the same image, the average time for automatically computing the optimal window sizes was under 1 second, whereas the proposed fully automatic 2-window approach took about 48 seconds.

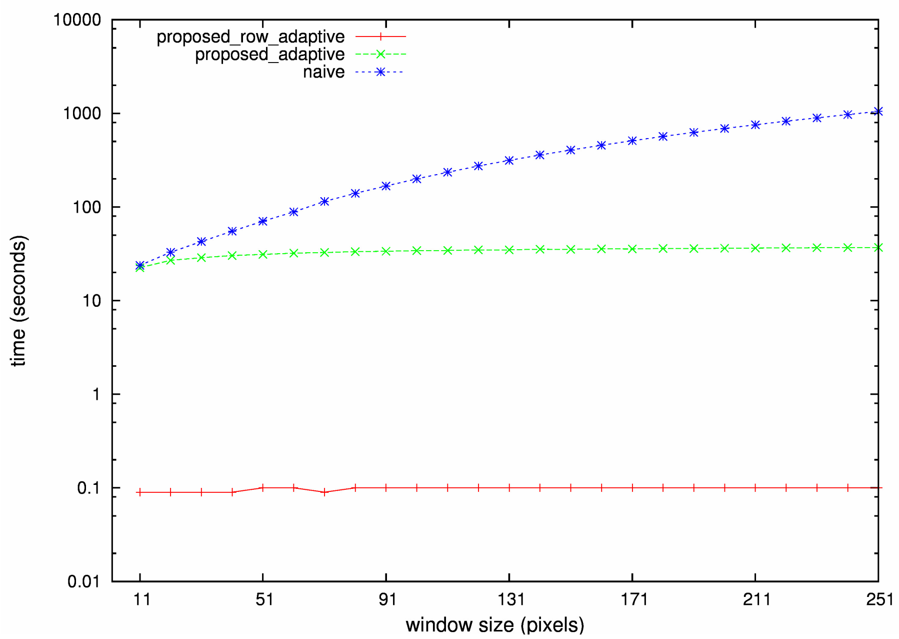
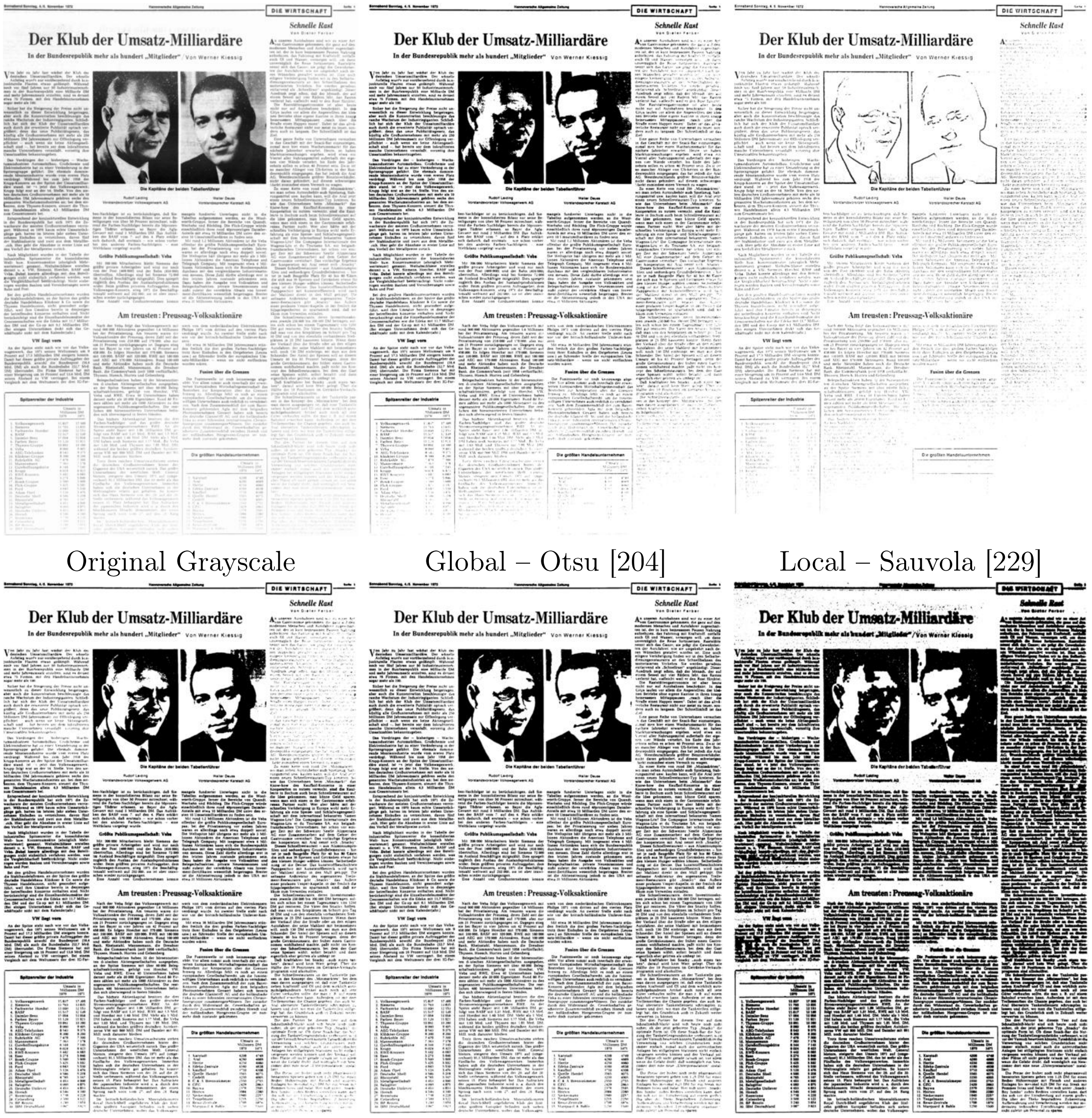


Figure 3.9: Timing comparison for proposed adaptation framework for Otsu's method. Note the logarithmic scale of the Y-axis

An adaptive variant of the Kittler et al. [151] minimum error global thresholding was also implemented using the described fully automatic framework and had its runtime performance tested. It was found that it is about 6 times slower than the adaptive method using Otsu thresholding. A simple, yet effective way of significantly reducing the runtime of both methods would be to use a regular grid to sample the input image, compute the thresholds just for the grid nodes and finally use interpolation (e.g. bilinear) for computing the rest of the thresholds. The size of the grid cells can be experimentally derived in a resolution-independent manner from the already computed dominant character height. In our experiments using a sampling step equal to the dominant character height we were able to reduce the running time of the automatic 2-window approach to under 1 second.



Original Grayscale Global – Otsu [204] Local – Sauvola [229]

Row-adaptive Otsu Adaptive Otsu Adaptive Kittler et. al [151]

Figure 3.10: Binarization results obtained by classical (top row) vs. proposed algorithms with automatically determined window sizes (bottom row)

From figure 3.10 one can see that the 2-window approach with automatically determined parameters produces very good results in comparison with the global binarization algorithm of Otsu and the Sauvola et al. algorithm. For the Sauvola et al. algorithm we have used the same automatically computed small window size, since other radii perform even worse. The poor performance of the Sauvola et al. algorithm is due to their implicit assumption that text is nearly black, which does not hold in the case of overexposed areas. We have also investigated the results on stained and badly degraded images where on certain areas the background and text values were also very close to each other, but had much lower grey values. In this situation, the Sauvola et al. algorithm produced better results than the proposed algorithm, which in turn performed better than global binarization. Note that the Sauvola et al. method works generally well on text, but has the undesirable effect of completely destroying halftones and hollowing out characters much larger than the window size (see figure 3.5). Our algorithm does not exhibit such problems, due to the mixture of local and global information utilized for selecting the thresholds. It is also worth noting that while the row-adaptive method is certainly less versatile than the proposed baseline method, it is much faster and works equally well on documents for which the contrast changes occur from top to bottom (i.e. for a light source located directly above or below the document image).

3.2 Color Reduction

In the previous section we have assumed that the input document scans are always single-channel grayscale images. We shall now relax this restriction and allow for color documents as input data. This is in accordance to the current digitization trend, which is clearly heading in the direction of full color scans. In the vast majority of cases, color images are stored using a gamma corrected (s)RGB space (see section 2.1.1.3 for more details on gamma correction). Without any loss of generality we shall assume this to be the default, unless stated otherwise.

The approaches for generic color reduction from the specialized literature can be categorized as follows:

- *indirect methods*, which first obtain a good approximation of the initial image via a grayscale document image, then apply some of the (multi-)thresholding methods previously described
- *direct methods*, which attempt to use the (subsampled) initial image to directly extract a suitable palette with a low number of colors (i.e. max. 10–20) and quantize the image accordingly

Note that typically the desired number of resulting colors is at most 10–20 in the context of document analysis. This number is given by the limitations imposed by layout and composition styles as used in the publishing industry [197]. Even today, the most common way of obtaining a binary image in the document analysis community is via a thresholding of the grayscale image obtained as the intensity map of the original color image. The intensity value for each pixel usually corresponds to the intensity component in the HSI or YUV color space, normalized to the discrete range $[0, 255]$. Formulae for computing the intensity channel from virtually any color space are readily available in the literature (e.g. [119], pp. 295–300). This is a highly simplistic approach, especially in the case of color documents which have been specially optimized for the human visual system. In the following section we shall give a brief overview of better, generic state-of-the-art techniques for color reduction from each category, together with several methods developed specifically for document images. Since most real-life color images exhibit some kind of degradations (see sections 2.1.1 and 2.2.1), it is customary to apply a pre-processing step so as to reduce the effects of such degradations on the results of the color reduction. Several of the most popular types of filters for color images are described at the beginning of the following section.

3.2.1 State of the Art

The objective of pre-processing color images in view of color reduction is to minimize the effects of typical distortions on the color reduction results. In other words, they ensure that the color reduction procedure produces stable results. This is generally accomplished by removing distortions and combined with some degree of color smoothing. We have mentioned in section 2.1.1 that the most common degradations found in digital images are blur, noise and contrast problems and have seen several basic methods for detecting and mitigating their effect. A detailed overview of many issues introduced by digital color halftoning (incl. Moiré patterns, error diffusion artifacts) can be found in the recent paper of Baqai et al [54]. These issues are of special interest for scanned materials, due to the use of halftoning in practically all printing devices coupled with the limited resolution of scanning devices.

We shall now mention several more advanced techniques specifically targeted at small *noise filtering*. Small noise is a very frequent occurrence in color images and in many cases represents the result of heat-related sensor issues or long exposure times coupled with small sensor sizes. Perreault and Hebert [214] introduce a median filtering technique which is able to construct the local histograms in constant per-pixel time. For median filters, the central pixel in each local window is replaced by the median of the gray values within the window. Another non-linear filter was the mode filter introduced by Davies [86]. Instead of a simple median, the mode filter uses the most probable intensity value (i.e. the mode) within the distribution given by the local histogram. Computing the mode of the distribution is a more challenging problem than the median (mode is not guaranteed to be unique), but may also be accomplished in constant time per pixel. The previous two filters have a tendency to shift the edges of objects in an image. This is generally an undesirable effect and other approaches have been proposed to deal with this problem. Two examples of such edge-preserving de-noising filters are the Kuwahara filter [158] and the class of bilateral filters [264]. Both the Kuwahara and the bilateral filters accomplish their task by additionally taking into account both pixel color values and their location within the local windows.

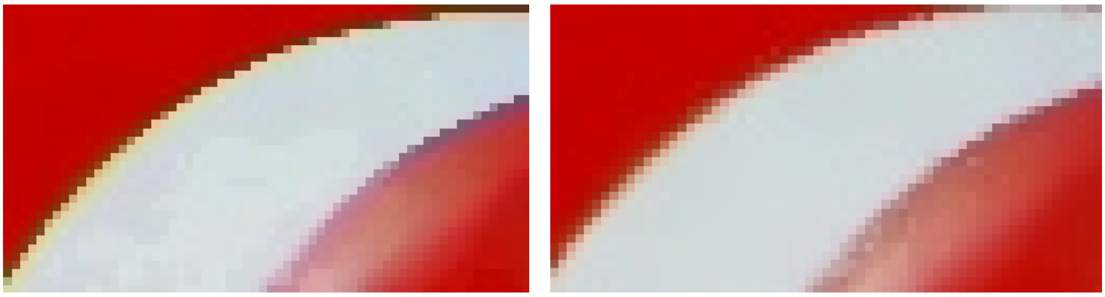


Figure 3.11: Results of mode filtering on (a part of) the Vodafone logo affected by color bleeding. Left – channel-wise filtering; right – vector space filtering. Note the significant reduction of color bleeding in the latter case. Source: [156]

As discussed in section 2.1.1, the application of filters on a per-channel basis leads to the phenomenon of color bleeding, i.e. the introduction of colors not originally present in the image. This effect is most visible at edges located between objects of significantly differing colors (e.g. black text on white/yellow background) – see figure 3.11. Note that some common lossy compression techniques, such as JPEG, have the same tendency to introduce visible color bleeding. Thus, even in case the input document images are considered to have been flawlessly acquired, the use of a lossy compression mandates the need for a proper handling of color bleeding. A solution to this problem was introduced in the form of *vector filters*. All vector filters operate by considering colors as N-dimensional vectors (defined by a direction and magnitude) and computing an ordering of the vectors in the local window via a similarity/distance metric. The result is then a (weighted) linear combination of the other vectors in the sense of the considered metric. In this way, interpolation between vectors produces far less visual artifacts than a per-channel interpolation. An even better solution is to avoid color interpolation in the first place, for example by using median vales instead of means. In this case, the vector with the highest total similarity/lowest sum of distances is taken as the result of the local window filtering. A review of existing vector filters is presented by Trahanias et al. [268] and more recently by Lukac et al. [175]. We discuss more on the issues regarding color interpolation in the following section.

At this point, the input image is relatively noise- and distortion free and color reduction algorithms can be safely applied. *Indirect color reduction* algorithms produce a grayscale image containing a full range of 256 gray values as an intermediary step. Two recent examples of such algorithms were introduced by Rasche et al. [221] in 2005 and Grundland and Dodgson [121] in 2007. It is interesting to note that Rasche et al. initially proposed their method for the purpose of re-coloring images for color deficient viewers. The authors incorporate the objectives of contrast preservation and maintenance of luminance consistency into an objective function which is then used to cast and solve the color reduction problem as a constrained, multi-variate optimization problem (a.k.a. multidimensional scaling). Grundland and Dodgson select as their design objectives a relatively constant contrast magnitude and polarity, as well as directly proportional dynamic range. They convert the image to the YPQ color space and randomly construct a set of sample pixel pairs. They introduce a dimensionality reduction technique dubbed predominant component analysis in order to compute the color axis that best represents the chromatic contrasts which are lost upon the grayscale mapping of the color image. The luminance and chrominance channels are then linearly combined and the resulting dynamic range is adjusted depending on the input image saturation as a final step. Figure 3.12 shows a visual comparison between the results produced by the Grundland and Dodgson algorithm and the traditional approach of using the luminance channel.

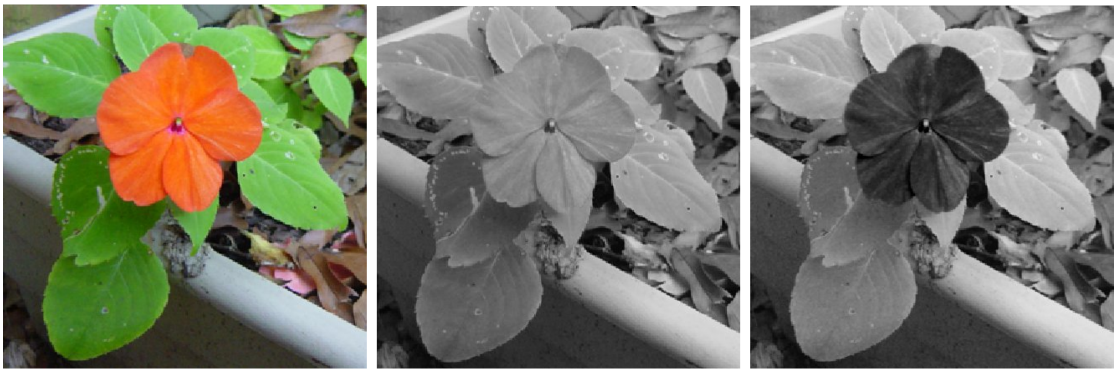


Figure 3.12: Color to grayscale mapping. Left – original image; Center – result of traditional approach using the luminance channel; Right – result of the contrast enhancing color reduction of Grundland and Dodgson [121]. Source: [121]

Many *direct methods for color reduction* have been proposed in the literature. One of the most well-known techniques is the *K-means* clustering [129] algorithm. The K-means algorithm requires the apriori specification of K, the number of (color) clusters in the result. It is also highly sensitive to cluster center seeding techniques, an issue analyzed in detail by Arthur and Vassilvitskii [41]. Other well-known approaches for color reduction which require the specification of the number of resulting cluster are the *median cut* [131] and *octree* [117] techniques. A more powerful clustering technique which can also compute the number of clusters by itself is the *mean shift* algorithm [80]. At its core, mean shift uses a gradient ascent approach for detecting the modes in the (color) density function. While mean shift does in principle allow the use of spatial proximity information along with color information, it has the disadvantage of a high computational cost and is sensitive to input parameters controlling the gradient ascent step size. *Statistical region merging* (SRM), introduced by Nock and Nielsen in 2004 [196] allows a more elegant integration of spatial relationships by employing a region merging approach. A new statistical local merging predicate is proposed, along with an ordering criterion for merging tests. Most significantly, the authors prove that (with high probability) their algorithm suffers from only a single source of error: overmerging, and the overmerging error is provably small. This is in contrast to other approaches, where undermerging and hybrid merging cases are just as likely to occur. Another method capable of incorporating spatial information along with color, is the approach using *maximally stable color regions* (MSCR) of Forssén [108]. Note that the method represents a generalization of the maximally stable extremal region (MSER) concept for grayscale images [180]. MSERs (and MSCRs by extension) are defined as the parts of an image where local binarization is stable over a wide range of thresholds. They were shown to be one of the most reliable affine-invariant region detectors in recent comparisons [186]. Note that MSCR is not per-se a color reduction method, but can be used for example to significantly and intelligently reduce the number of clusters for a subsequent k-means or mean shift procedure. Since the K-clustering problem with variable K is NP complete [111], many other approximate color reduction methods have been proposed. A recent comprehensive survey of clustering algorithms was done by Xu and Wunsch [292].

In the *area of document image analysis*, color reduction methods are relatively sparse and in many times incorporated as integral part of the geometric layout analysis algorithms. As a recent example of such color document segmentation algorithm, we mention the approach of Tsai and Lee [271]. They propose an efficient method for document segmentation using a set of pre-defined segmentation and classification rules. While the authors report visual results superior to those of the commercial products ABBYY FineReader and PenPower OCR on a heterogeneous set of color document scans with simple layouts (books, magazines, name cards, receipts), it is unclear whether the presented methodology generalizes well to documents featuring complex multi-column layouts.

Nikolaou and Papamarkos [193] specifically target complex color documents in their approach. As a first step, they use an edge preserving filter to remove small noise components. A set of color samples is then extracted from the interior of the objects in the input image (detected using an edge map). Finally, a mean-shift procedure is applied on the extracted 3-dimensional color samples to quantize the input image. The authors compare the proposed method with other color reduction techniques on a set of 50 manually ground truthed color documents using as measure the correspondences between the connected components located in the text areas. A further step up in difficulty is made by Lacroix [159] in the work on automatic palette detection for color graphics containing large non-text regions, such as photographs, maps, artwork, logos. The author introduces a novel clustering algorithm, named median shift, which operates in an incremental fashion similarly to mean shift. In addition, for making the approach more robust to outliers, the authors require the specification of an expected radius for the clusters and apply an iterative post-processing step for discarding small clusters. A specialized color bleeding removal procedure is applied prior to the clustering step in the CIE L*a*b* color space. The algorithm is shown to generate good visual color reduction results on a set of 6 printed maps and the Macbeth color reference chart.

3.2.2 Guidelines for Holistic Color Reduction in Digitized Documents

This section is dedicated to presenting several important guidelines concerning the implementation of a truly generic color reduction framework for digitized documents. We present these ideas because we strongly believe that the very nature of document images requires a holistic approach combining and in some cases going beyond the current state-of-the-art methods. We make the aforementioned claim on the grounds that the coloring and composition rules based on which documents are created have been and continue to be specifically adapted for human readers.

Firstly, in the previous section we mentioned the necessity of using vector filtering for effectively removing both small noise and color bleeding artifacts. The color reduction step must also make full use of *vector operations* and requires in many cases a non-trivial reformulation of the color reduction algorithms. While this certainly represents a step in the right direction, we have also seen that even vector-based approaches must in most cases interpolate between colors. This will invariably cause problems in any color space (including CIE L*a*b*) simply because at present there does not exist any color space that is even close to perceptually uniform. The lack of perceptual uniformity is easy to see from the ever increasing amount of color distance measures proposed to partially solve the color comparison problem [100, 138, 188]. Note that the use of more accurate color distance measures does not affect the interpolation issue at all – the latter is still prone to generate the same perceptual inconsistencies. An interesting, but not entirely successful attempt to transform color spaces in such way that the simple Euclidean metric reflects the more complex color distance metrics (e.g. as part of IPT or CIECAM02 [100, 188]) was recently subject to the work of Xue [294]. The performance of the Euclidean distance in the transformed color space for the CIECAM02 [188] distance was nearly as good as using the non-Euclidean distance metric in the original color space. However, for the IPT [101] distance metric the errors were significantly higher in the transformed color space. The transformed color spaces are of special interest because of the possibility of making unrestricted use of the Euclidean metric for color interpolation while generating less color artifacts.

This observation leads us to our second guideline. Since even in the transformed Euclidean color spaces the interpolation is still error prone, a successful vector-based approach must *minimize the amount of color interpolations performed.* This can be accomplished for example via vector median filters or by more complex methods minimizing the sum of spatial volumes over which the interpolations are performed. Note that the magnitude of the potential maximum color error is directly proportional to the distance between the colors, thus minimizing the overall distances considered will minimize the color error.

Thirdly, all methods we have seen so far in our review do not take into account many important *particularities of the human visual system* (HVS). The HVS detects patterns of light rather than being sensitive to the absolute magnitude of the light energy ([247], pp. 479). Several examples of effects specific to the HVS commonly encountered in (static) digital images are:

- *lightness constancy* – the perceived brightness of a surface is invariant to large changes in the actual illumination
- *simultaneous contrast* – the modification of the perceived brightness of a surface with respect to the contrasting brightness of the surrounding region
- *subjective contours* – the HVS has a tendency to create edges for cases where the resulting form looks more complete. Note that the artificial edges can be generated

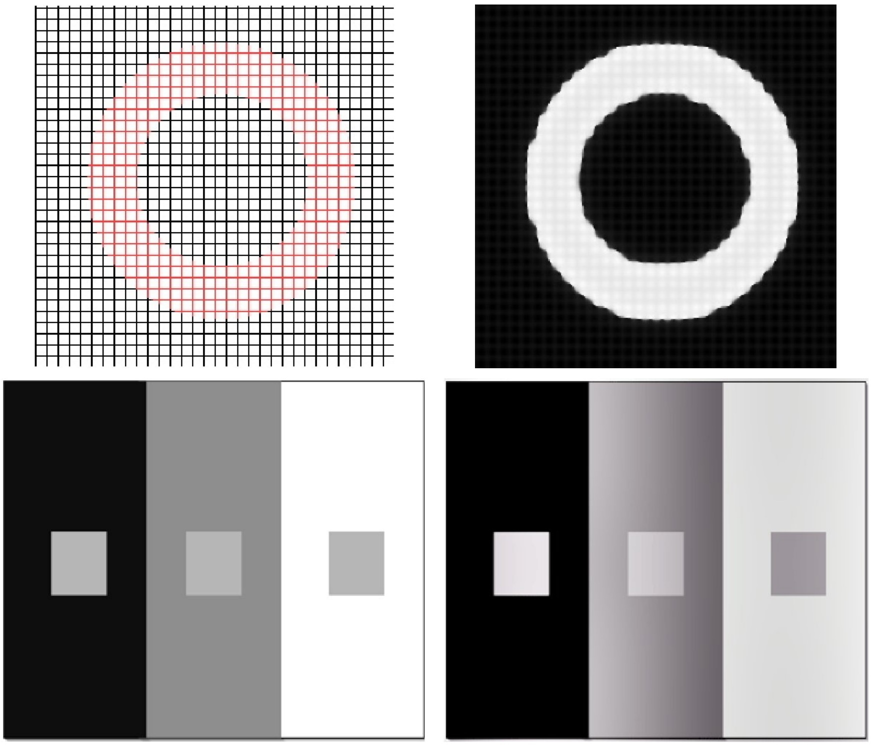


Figure 3.13: Prediction of hue spreading (top row) and simultaneous contrast in the iCAM framework. Source: [101]

both by color (*hue spreading*) and by the spatial arrangement of objects/ patterns, irrespective of color

- *Hunt effect* – for constant cromaticity, the perceived colorfulness increases with luminance
- *Stevens effect* – perceived lightness contrast increases with increasing luminance - i.e. when luminance increases dark colors look darker and light colors look lighter
- *Bartleson-Breneman effect* – perceived contrast increases with increasing luminance
- *light, dark and chromatic adaptation* – automatic visual acuity/ color balancing

The layouts and colors of document images have been specifically optimized to make use of these effects. For example the logical grouping of a set of paragraphs into an article can be inferred easily by humans via hue spreading even for a very sparse (e.g. halftoned), gradient background coloring. If the background color is wrongly discarded in the color reduction step, the aforementioned logical grouping will become much harder to detect correctly in the later processing stages. Traditional color spaces completely disregard these effects. More recently, color appearance models were introduced as holistic frameworks for the simulation of the human visual system. They manage to partially address the aforementioned issues. The most complete color model currently in large-scale use is CIECAM02 [188], introduced in 2002. In the meantime, some of its deficiencies were addressed by iCAM [101], proposed in the scientific literature by Fairchild and Johnson right after the standardization of CIECAM02 [100].

We note that while these issues are generally known to color specialists, until now there exist no works in document image analysis where they have been taken into account and combined as part of a consequent approach. A first step in this direction was the recent work of Rodriguez Valadez [224], done under the supervision of the author of the current

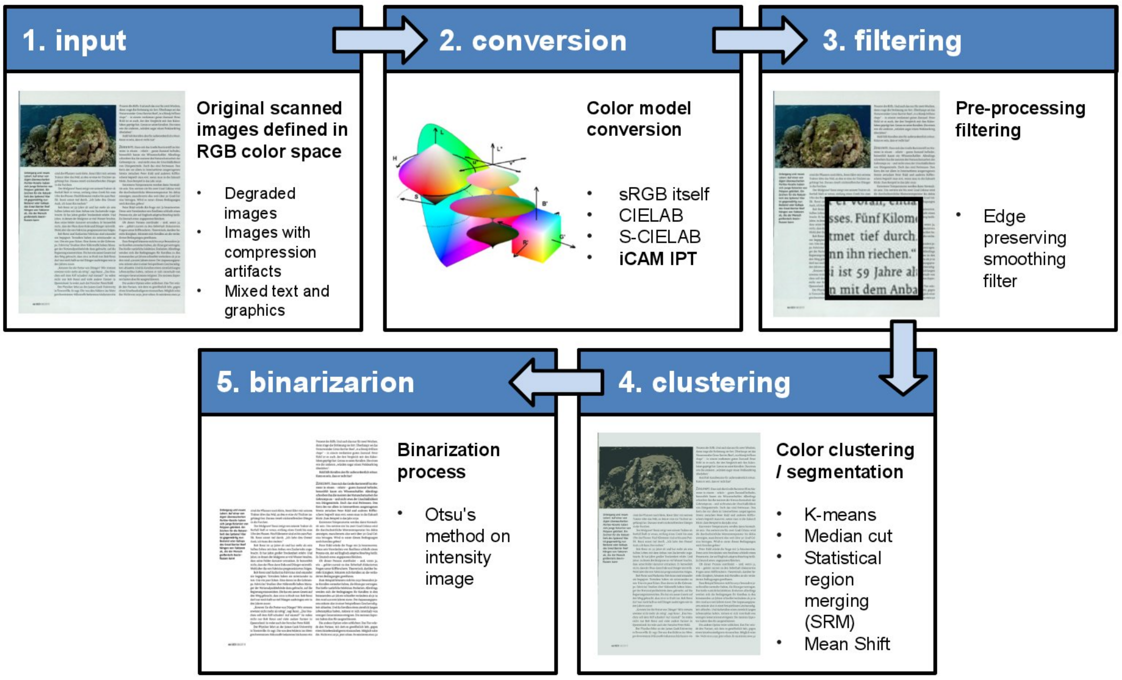


Figure 3.14: Overview of the methodology used by Rodriguez Valadez [224] for evaluating different combinations of color appearance models and color segmentation algorithms for the purpose of text extraction from document images. Source: [224]

thesis. Rodriguez Valadez used the framework depicted in figure 3.14 to compare the viability of different color spaces (RGB, CIE c) and color appearance models (S-CIE L*a*b*, IPT) for the purpose of color reduction in document images. The comparison comprised several popular methods of color reduction, including K-means, median cut, mean shift and statistical region merging. Unfortunately, all approaches were tested using their scalar version, as opposed to the preferable vector version. One must note that the vectorization of the color reduction approaches is in many cases a non-trivial endeavor, especially due to the additional computational complexity entailed by vector operations.

3.2.3 Evaluating Color Reduction in Document Images – a Discussion

In this section we shall go over the evaluation performed by Rodriguez Valadez and discuss some forthcoming issues relevant for future, meaningful evaluation methodologies. In order to be able to obtain a meaningful quantification of the results produced by each combination of color model and color reduction technique, the author restricted the scope of the evaluation to textual regions. Textual regions can be described more precisely than non-text with respect to the expected result. Three data sets consisting of real-life documents were used for testing. Each set exhibited different distortions: old text-only documents showing paper and ink degradations; text-only documents showing different degrees of compression artifacts; partially degraded documents containing both text and non-text content. The ground truth was generated by manual pixel-accurate labeling of the text content, as can be observed in figure 3.15. The reader is advised to note that practically all previous scientific papers containing an assessment of the results of color reduction have done so either just by visual inspection or via generic matching measures. In the work of

Rodriguez Valadez [224] the F-measure was selected for quantifying the closeness of the results to the ground truth.

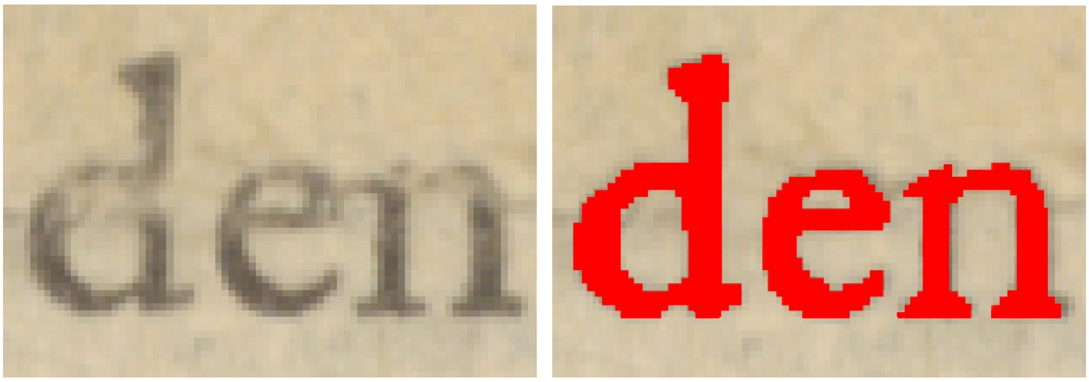


Figure 3.15: Example of manual pixel-accurate ground truth segmentation. While for clearly defined contours the limits can be easily defined, the segmentation of degraded portions require a subjective decision. Source: [224]

It is clear that the pixel-accurate labeling results produced as ground truth by any human are inherently subjective. So why choose them as ground truth instead of the more traditional OCR results? In the past, OCR-based measures have indeed been the norm among the document image analysis community. Over the years, the community has realized that OCR-based measures also contain a heavy bias due to the different engines and the corresponding (oftentimes opaque) parameter settings. The opaque bias of OCR systems was generally found to be even less desirable as ground for an objective evaluation than the fully transparent pixel-accurate ground truth produced by human annotators. One possibility of reducing the subjectiveness of the labeling is the combination of multiple annotations for the same content. Sadly, in most situations, this approach is not feasible due to the high cost of human labor. We shall return to the labeling issue shortly.

Before doing so, we describe the state-of-the-art evaluation measures working on pixel-accurate ground truth in document image analysis. In the most recent and largest comparisons of document binarization methods to date, the DIBCO 2009 [115] and 2011 [218] two highly similar sets of evaluation measures were used for ranking the participating methods. In both cases, the F-measure was used as a primary method. The F-measure is defined as follows:

$$FMeasure = \frac{2 \times Recall \times Precision}{Precision}\text{, where}$$
$$Recall = \frac{TP}{TP + FN}, Precision = \frac{TP}{TP + FP}$$

TP, FP and FN denote the true positive, false positive and false negative values, respectively.

The other two evaluation measures common to both competitions were the peak signal-to-noise ratio (PSNR) and the misclassification penalty metric (MPM). The former represents a similarity measure between two images, whereas the latter encodes the ability of an algorithm to produce an accurate representation of the borders of a set of objects (here:

connected components). The exact definitions of the two measures are:

$$PSNR = 10 \log \frac{C^2}{MSE} \text{ , where}$$
$$MSE = \frac{\sum_{x=1}^{M} \sum_{y=1}^{N} (I(x,y) - I'(x,y))}{M \times N}$$

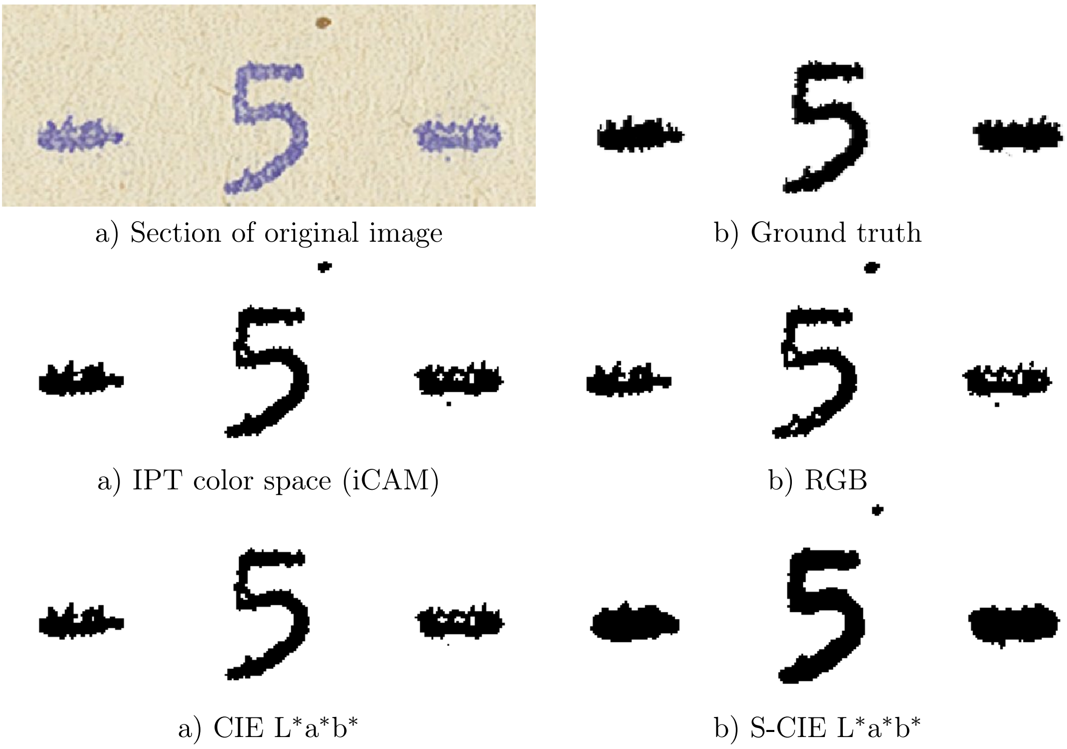
M,N represent the width and height of the input image I, respectively, whereas I' represents the result binary image being evaluated.

$$MPM = \frac{MP_{FN} + MP_{FP}}{2} \text{ , where}$$
$$MP_{FN} = \frac{\sum_{i=1}^{N_{FN}} d_{FN}^{i}}{D} \text{ , } MP_{FP} = \frac{\sum_{j=1}^{N_{FP}} d_{FP}^{i}}{D}$$

Here d_{FN}^{i} and d_{FP}^{j} denote the distance of the i^{th} false negative and j^{th} false positive from the contour of the ground truth, respectively.

In addition to the three aforementioned measures, DIBCO 2009 employed the negative rate metric (NRM), defined as a combination of the false negative and the false positive rates. DIBCO 2011 replaced the NRM with the distance reciprocal distortion metric, a measure proposed by Lu et al. [172] for the comparison of binary image in order to better correlate to the human visual perception. Upon closer inspection, one may see that practically none of these metrics fare well when confronted to the more subjective ground truth portions. This is because they do not allow for any kind of (small) deviation from the ground truth without immediately incurring a penalty. This issue has been analyzed in more detail by Smith [248]. The author employs multiple human annotators on the same ground truth data as used in the DIBCO 2009. Interestingly, it was found that the variability among the human-generated ground truth was actually larger than that among the algorithms taking part in the competition. Even more significantly, the human annotations were comparable in closeness to the top DIBCO results. As Smith notes, this is likely an indication that in a contest, a differentiation between algorithms above a certain level is bound to be arbitrary. Thus the best method will actually be the one best fitting the preferences of the contest organizer.

As we can indeed see from figure 3.16, the S-CIE L*a*b* result subjectively appears to be better than the binarization results produced using the other color spaces. Yet, the F-measure and all other measures presented here will invariably rank such results as significantly worse. The ranking using the F-measure can be observed in figure 3.17. Note that an OCR-based evaluation would in this case most likely not be able to differentiate in any way between the different methods. Beside the previously discussed issues regarding reproducibility, the poor discriminative power is one of the main reasons why we advise against returning to OCR-based evaluation methods. Instead we propose as possible solution a weighted average of two measures: a contour-based measure and a skeleton-based measure. Reliable and accurate algorithms for extracting contours and skeletons exist in the specialized literature [69, 234]. A resolution-independent smoothing/pruning of both representations is also possible in order to partially cope with the invariably subjective ground truth data. The contour and skeleton can then be approximated via polygonal shapes which can efficiently be matched against the ground truth [88, 232, 299].



a) Section of original image

b) Ground truth

a) IPT color space (iCAM)

b) RGB

a) CIE L*a*b*

b) S-CIE L*a*b*

Figure 3.16: Comparison of results obtained in a section of an image using different color models. Source: [224]

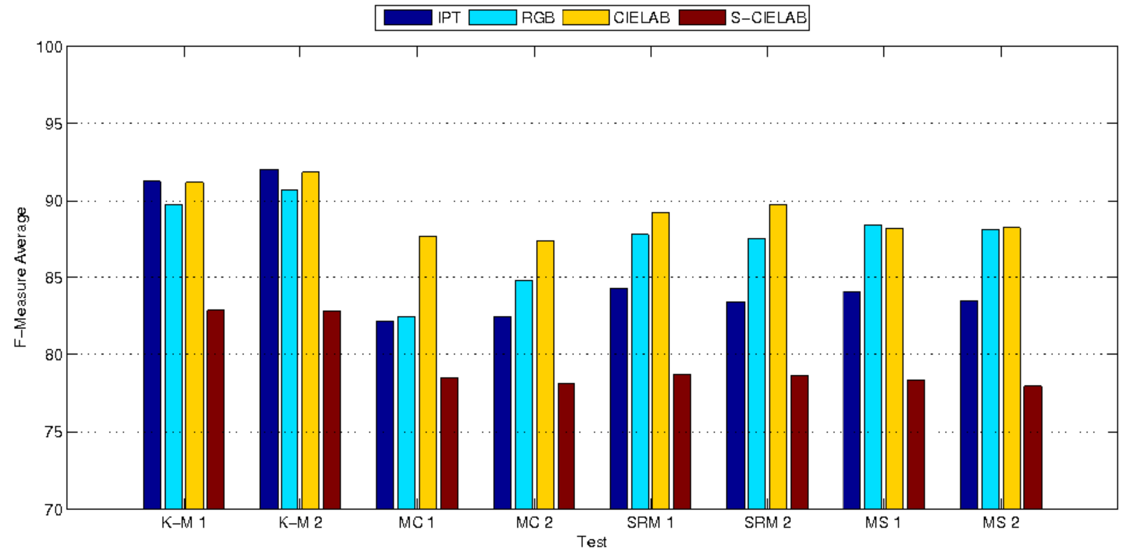


Figure 3.17: Test results obtained by Rodriguez Valadez [224] on a data set containing 5 highly degraded, text-only document scans using different color reduction methods: K-Means, median cut, statistical region merging [196] and mean shift [80]. Source: [224]

3.3 Conclusions

In the first part of the current section we have introduced the problematic of thresholding in the context of document analysis. From a review of the state of the art global and adaptive methods, we have identified the main deficiencies of each class of algorithms. A theoretically optimal solution from both computational complexity- and threshold selection point of view was introduced as original work. The proposed framework allows the adaptation of global algorithms into local ones, while maintaining the lowest possible computation cost. The framework was validated by the implementation of two optimal global algorithms, well known in the document analysis community for their excellent performance on document materials. Our tests show that traditional implementations have a running time significantly higher than that using our framework. For example, when using local windows of radius 125, our implementations are around 30 times faster. Further speed gains can be obtained by using a sampling grid with a period proportional to the automatically detected character height in the dominant font.

The second part of the chapter introduced the complex set of issues posed by color reduction both in a generic context and specifics related to document scans. We reviewed the current stand of science in the areas of color filtering, vector filtering and color reduction/segmentation. We saw that while the future will invariably belong to color digitized documents, the state of the art in document analysis still contains very few methods processing such materials effectively. A set of guidelines were offered as possible solution to the generic document color reduction problem. The guidelines refer to the use of vector-based approaches for both filtering and color reduction, the necessity of minimizing the color interpolation errors and the use of color models which more accurately reproduce the human vision system. Additional practical problems regarding the objective and meaningful evaluation of such color reduction frameworks were discussed using the practical example of the system proposed by Rodriguez Valadez [224]. We conclude the chapter with the observation that even a holistic color reduction approach as discussed in the previous sections may still identify wrong palettes for highly-degraded documents or for pages containing large halftones. This is why future algorithmic solutions in this area must combine the individual per-page results with assumptions regarding style consistency among sets of periodical issues or within whole books. Thus, apriori knowledge about the logical borders of the issues/books must either be available from human annotators or (semi-)automated approaches such as the one proposed in section 6.2 can be exploited as part of an extended, iterative document image understanding workflow.

Chapter 4

Skew and Orientation Detection

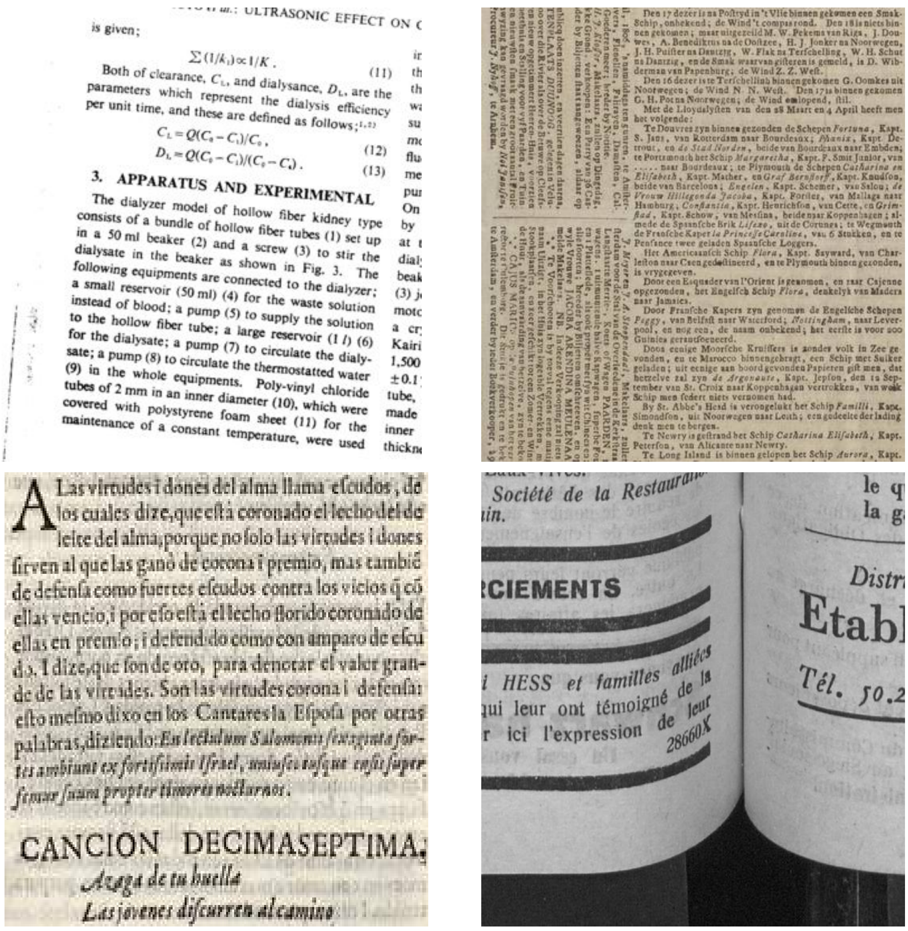
The way you see things depends a great deal on where you look at them from.

– N. Juster *(The Phantom Tollbooth)*

Document skew is a very common distortion found in document images as a result of the digitization process, or as a feature of the document's layout. In most cases, even small skew angles have an obvious detrimental effect on the accuracy of the subsequent geometric and logical layout analysis steps. This is due to the fact that most existing algorithms require a proper document alignment before application, although there exist a few methods which do not require any previous skew correction [39, 98, 150]. One must however take note that the skew-independent layout analysis methods either pose certain restrictions on the possible angle range, or are considered in isolation from the subsequent processing operations, such as region classification or text line extraction. As a result, the apparent simplification leads to the necessity of applying more sophisticated techniques for other tasks.

In general, the skew within a document page can fall into one of the following three categories (see figure 4.1): *global skew*, assuming that (almost) all page blocks have the same slant; *multiple skew*, when certain blocks have a different slant than the others and *non-uniform skew*, when the slant fluctuates (such as lines having a wavy shape). It is worth noting that whereas multiple skew is most commonly an intentional layout feature, non-uniform skew is usually the result of an imperfect digitization process possibly combined with paper degradation/crumpling. As such, non-uniform skew is most pronounced on historical documents or on document photographs taken with a regular digital camera or mobile phone. Perhaps the most common occurrence of non-uniform skew can be observed in book scans in close proximity to the book fold, where the text lines are visibly curled. In case the non-uniform skew is highly pronounced (i.e. incl. perspective distortions and page folds/curls), the skew detection and correction process is commonly referred to as document *dewarping*.

In the current work, we focus mainly on global skew estimation, as it represents by far the most prevalent type of skew found in (semi-) professionally digitized documents. In contrast, non-uniform skew is almost absent from such scans – the only notable exception being the area close to the book fold in case of book scans. The first section of the current chapter contains a holistic overview of the document skew detection area, including the



ULTRASONIC EFFECT ON

is given;

$$\Sigma(1/k_1) \propto 1/K\,. \qquad (11)$$

Both of clearance, C_L, and dialysance, D_L, are the parameters which represent the dialysis efficiency per unit time, and these are defined as follows;[1,2]

$$C_L = Q(C_0 - C_1)/C_0\,, \qquad (12)$$

$$D_L = Q(C_0 - C_1)/(C_0 - C_d)\,. \qquad (13)$$

3. APPARATUS AND EXPERIMENTAL

The dialyzer model of hollow fiber kidney type consists of a bundle of hollow fiber tubes (1) set up in a 50 ml beaker (2) and a screw (3) to stir the dialysate in the beaker as shown in Fig. 3. The following equipments are connected to the dialyzer; a small reservoir (50 ml) (4) for the waste solution instead of blood; a pump (5) to supply the solution to the hollow fiber tube; a large reservoir (1 *l*) (6) for the dialysate; a pump (7) to circulate the dialysate; a pump (8) to circulate the thermostatted water (9) in the whole equipments. Poly-vinyl chloride tubes of 2 mm in an inner diameter (10), which were covered with polystyrene foam sheet (11) for the maintenance of a constant temperature, were used

Den 17 dezer is na Poſttyd in 't Vlie binnen gekomen een Smak-Schip, onbekend; de Wind 't compasrond. Den 18 is niets binnen gekomen; maar uitgezeild M. W. Pekema van Riga, J. Douwes, A. Benedikrus na de Ooſtzee, H. J. Jonker na Noorwegen, J. H. Puiſter na Dantzig, W. Flak na Terſchelling, W. H. Schut na Dantzig, en de Smak waarvan giſteren is gemeld, is D. Wibderman van Papenburg; de Wind Z. Z. Weſt.

Den 16 dezer is te Terſchelling binnen gekomen G. Oomkes uit Noorwegen; de Wind N. N. Weſt. Den 17 is binnen gekomen G. H. Pot na Noorwegen; de Wind omlopend, ſtil.

Met de Lloydslyſten van den 28 Maart en 4 April heeft men het volgende:

Te Douvres zyn binnen gezonden de Schepen *Fortuna*, Kapt. S. Jans, van Rotterdam naar Bourdeaux; *Phœnix*, Kapt. Detrout, en *de Stad Norden*, beide van Bourdeaux naar Embden; te Portsmouth het Schip *Margaretha*, Kapt. F. Smit Junior, van ..... naar Bourdeaux; te Plymouth de Schepen *Catharina en Eliſabeth*, Kapt. Mather, en *Graf Bernſtorf*, Kapt. Knudſon, beide van Barcelona; *Engelen*, Kapt. Schemer, van Salou; *de Vrouw Hillegonda Jacoba*, Kapt. Fortier, van Mallaga naar Hamburg; *Conſtantia*, Kapt. Heurichſon, van Cette, en *Grimſtad*, Kapt. Schow, van Meſſina, beide naar Koppenhagen; almede de Spaanſche Brik *Liſano*, uit de Corunes; te Wegmouth de Franſche Kaper *la Princeſſe Caroline*, van 6 Stukken, en te Penſance twee geladen Spaanſche Loggers.

Het Americaanſch Schip *Flora*, Kapt. Sayward, van Charleſton naar Caen gedeſtineerd, en te Plymouth binnen gezonden, is vrygegeven.

Door een Esquader van l'Orient is genomen, en naar Cajenne opgezonden, het Engelſch Schip *Flora*, denkelyk van Madera naar Jamaica.

Door Franſche Kapers zyn genomen de Engelſche Schepen *Peggy*, van Belfaſt naar Waterford; *Nottingham*, naar Leverpool, en nog een, de naam onbekend; het eerſte is voor 200 Guinies gerantſoeneerd.

Door eenige Moorſche Kruiſſers is zonder volk in Zee gevonden, en te Marocco binnengebragt, een Schip met Suiker geladen; uit eenige aan boord gevonden Papieren giſt men, dat hetzelve zal zyn *de Argonaute*, Kapt. Jepſon, den 12 September van St. Croix naar Koppenhagen vertrokken, van welk Schip men ſedert niets vernomen had.

By St. Abbe's Head is verongelukt het Schip *Familli*, Kapt. Simondſon, uit Noorwegen naar Leith; een gedeelte der lading denk men te bergen.

Te Newry is geſtrand het Schip *Catharina Eliſabeth*, Kapt. Peterſon, van Alicante naar Newry.

Te Long Island is binnen gelopen het Schip *Aurora*, Kapt.

A Las virtudes i dones del alma llama eſcudos, de los cuales dize, que eſtà coronado el lecho del deleite del alma, porque no ſolo las virtudes i dones ſirven al que las ganò de corona i premio, mas tambiē de defenſa como fuertes eſcudos contra los vicios q̄ cō ellas vencio, i por eſo eſtà el lecho florido coronado de ellas en premio; i defendido como con amparo de eſcudos. I dize, que ſon de oro, para denotar el valor grande de las virtudes. Son las virtudes corona i defenſa: eſto meſmo dixo en los Cantares la Eſpoſa por otras palabras, diziendo: *En lectulum Salomonis ſexaginta fortes ambiunt ex fortiſſimis Iſrael, uniuſcuiuſque enſis ſuper femur ſuum propter timores nocturnos.*

CANCION DECIMASEPTIMA.

Azaga de tu huella

Las jovenes diſcurren al camino

Société de la Restaurant

RCIEMENTS

HESS et familles alliées

qui leur ont témoigné de la

ici l'expression de leur

28660X

le

la

Distr

Etabl

Tél. 50.2

Figure 4.1: Different types of skew found in document scans: global skew (top left), multiple skew (top right) and two common instances showing non-uniform skew (bottom)

current state of the art in multiple- and non-uniform skew detection. The second section offers solutions to several pressing issues in global skew detection by means of a generic framework built upon a solid theoretical basis. The framework represents an extension of the work presented in [154]. Despite posing only weak assumptions on the input document images, the proposed framework is experimentally shown to be able to accurately identify the document skew on a heterogeneous data set consisting of around 110 000 document scans. By making use of the experimental observations, we are further able to devise a confidence measure for each obtained result. We conclude the chapter with a short review and promising directions for future research.

4.1 Overview

4.1.1 Global Skew and Orientation Detection

Given a single digital document page, the task of a *global skew detection* algorithm is straightforward: determine the angle by which it was skewed as accurately as possible. Having computed the skew angle, a classic image processing algorithm or one specialized for documents [47] can afterward be applied for rotating the image in the reverse direction.

Subsequently the deskewed image is fed into the next processing module, along with the information regarding the original skew angle. This information is in many cases useful, e.g. for highlighting or visualization purposes on the original scan.

Perhaps the simplest skew detection solution which springs to mind is to determine the location of at least two corners of the original document and subsequently use them to compute the skew angle. After all, algorithms for corner detection such as the Harris detector [127] have been thoroughly investigated and are invariant to rotation, scaling and illumination variations [231]. But what happens when the digitized pages did not lie completely flat on the platen or had folded sections, when the page edges are completely invisible (i.e. white digitization background) or when close to the page edge there are other prominent corners to be found (e.g. from photographs)? Situations such as these are commonplace in mass document processing and they show just how error-prone traditional computer vision-oriented approaches can be when directly applied to document processing. This is also the reason why virtually all techniques for document skew detection employ specialized algorithms and much more robust features (often extracted from the page text) in order to compute the skew angle.

The largest classes of methods for skew detection are based on projection profile analysis, Hough transforms and nearest neighbor clustering. Recently the focus of the research community has shifted more towards the nearest neighbor approaches, as they seem to offer the greatest flexibility and accuracy. Detailed surveys of algorithms capable of dealing with both global and multiple skew may be found in [46, 48, 66, 134].

A common characteristic of most skew detection algorithms is their assumption that the input document contains some amount of text. In such case, the detection of the document skew is reduced to detecting the skew of the text lines, as text lines are usually aligned with the horizontal (or vertical) axis of the page. Techniques described in the scientific and patent literature are commonly categorized into the following groups:

- Projection profile analysis methods [49, 50, 217, 251]
- Methods using the Hough transform [253, 298]
- Methods based on nearest neighbor clustering [46, 174, 198, 277]
- Methods based on cross-correlation [208, 295]
- Other methods – such as those based on morphological transformations [74], gradient direction analysis [228], Fourier spectrum analysis [217], subspace line detection [19], particle swarm optimization [225] a.s.o.

As one of the earliest class of global skew detection algorithms, *projection profile analysis* methods make very strong assumptions on the input documents. More specifically, they require documents to consist exclusively of text regions, which must be arranged along parallel straight lines. When this restriction holds, the (smoothed) projection profile along the correct skew angle will naturally lead to much higher histogram peaks. This property of the histogram is encoded via an objective function, and as such the determination of the document skew angle reduces to finding the angle with a maximum value for the objective function. Repeatedly computing the projection profile along each skew angle for each object pixel in the image is a very computationally expensive process, and because of this many variants of the basic method have been proposed. Their aim is to reduce the amount of data involved in the computation of the projection profile and/or to improve the search strategy (e.g. coarse-to-fine search).

Postl [217], in one of the earliest such methods, uses only points on a coarse grid to compute the projection profile, whereas the objective function to maximize is the sum of squared

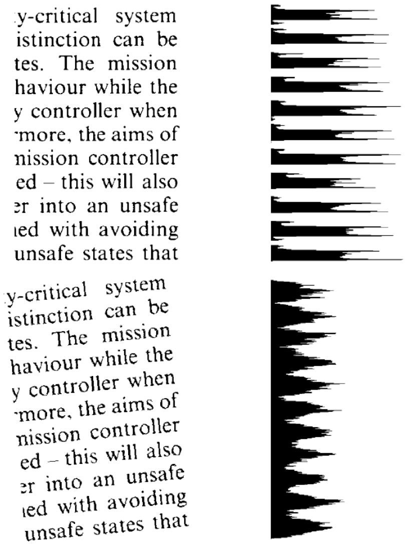


Figure 4.2: Top row: ideal image with no skew and its corresponding horizontal projection profile; Bottom row: same image, skewed by 5° along with its corresponding horizontal projection. Note the more pronounced higher peaks and deeper valleys in the skew-free histogram [134]

differences between successive bins in the projection. Baird [50] selects for computing the projection only the midpoints of the bottom side of the bounding boxes surrounding each connected component. Also, in order to speed up the search for the optimum skew angle, an iterative search method is proposed. In the first step, the whole angular range is searched at a coarse angular resolution. Afterward, in each iteration, the angular range is restricted to a neighborhood of the current best angle and the angular resolution is increased. A thorough comparison and evaluation of the Postl and Baird methods, along with two other algorithms, may be found in [48]. Such techniques were also adapted for direct application on compressed document images [49]. A significant disadvantage of projection profile methods is their brittleness on documents containing significant non-text regions (e.g. line-art, halftones) and/or having multiple layout columns with differing text line placement. As we shall see in section 6.3.1.2 however, in cases where the restrictions hold (e.g. for small textual areas), the objective functions for projection profile assessment offer very powerful discriminative features.

Techniques which use the *Hough transform* have been relatively well explored in many scientific papers [253, 298]) and are based on two main assumptions: characters are aligned in straight text lines and text lines are parallel to each other. The Hough transform maps each object pixel *(x, y)* of the input image into a curve in the parameter space (ρ, θ) by using the transform: $\rho = x \cos\theta + y \sin\theta$. Here, ρ denotes the distance from the origin to the line and θ is the angle between the normal to the line makes and the horizontal axis. In practice, a so-called accumulator array is parameterized by a range of discrete values for ρ and θ. The values for θ and ρ which result from applying the given equation for each input image pixel *(x, y)* are used as indices in the accumulator array, and the value of the accumulator array at the respective location is incremented. Aligned pixels will produce peaks in the Hough space, from which the dominant skew angle can easily be determined. The angular resolution of the method depends on chosen resolution of the θ axis. The complexity is linear with respect to the number of input points and the required angular resolution. Also, the necessary amount of memory is directly proportional to the product of these two terms. Consequently, the different proposed methods try to minimize one or both of these values using certain computational tricks, while still obtaining a high enough angular accuracy. Such tricks commonly include the down-sampling of the input image, the utilization of just the centroids of character-like connected components and the application of the Hough transform in a hierarchical manner. While Hough transform-based methods are mathematically well-founded, they generalize very poorly – i.e. an exact parametric representation of the expected shape is required, and as such are unsuited for application on documents exhibiting non-uniform skew.

The common denominator of all *nearest neighbor-based skew detection* algorithms is the requirement that the input document image has already had its connected components labeled. The individual connected components are then grouped into lines based on the distances between them, and the skew of each line is computed using a certain method. Finally, the computed skew angles are used to compute a histogram (or are simply sorted) and the page skew is estimated to be the highest peak in the histogram (or the mean/-median text line angle). Nearest-neighbor methods are in general more computationally expensive than the other classes of methods, entailing connected component labeling and repeated proximity searches. For obtaining a higher accuracy and/or improving the run-time performance, some methods may also employ a coarse-to-fine search (as described for projection profile-based algorithms). Nearest neighbor methods are the most appropriate to deal with multiple skew angles [66], make fewer restrictions with regard to the maximum possible skew angle and, unlike most other approaches, generally perform well on complex pages containing relatively little text (see examples in [51, 174]). Furthermore, connected components can be extracted directly from grayscale or color images by using suitable color distance measures.

Although in the early years of document image analysis, nearest-neighbor approaches were generally avoided because of their low speed relative to other methods, this does not hold today anymore. Optimal algorithms for connected component labeling [69, 90], as well as spatial search structures have increased their speed substantially – current approaches run in much less than 1 second per document page. This reason coupled with their ability of handling multiple- and non-uniform skew has made nearest neighbor approaches the most popular class of algorithms for skew detection in the more recent years. Van Beusekom et al. [277] use the RAST algorithm for computing the baseline, ascender- and descender line for each text line on the page in descending order of quality for the linear fit. The skew angle of the document is taken to be equal to that of the text line best fitting the linear model. The document orientation is detected by making indirect use of the ascender-to-descender ratio, reflected in a lower fit quality for the top N ascender lines versus the top N descender lines. Ávila et al. [46] compute the set of text lines on the document page by employing a heuristic in-memory traversal technique using as starting point the noise-filtered set of connected components. Afterward, the skew of each text line is extracted from a baseline and added to a 2-level histogram of skew angles (bin resolution: 1, respectively 0.1 degrees). A coarse-to-fine search for the histogram position of the maximum then determines the document skew angle. The document orientation explicitly uses the total number of ascenders and descenders computed over the set of text lines. It is interesting to note that the novel framework to be introduced later in the current chapter also represents an example of nearest neighbor approach.

Skew detection *methods using cross-correlation* aim at estimating the document skew by measuring vertical deviations along the image. They make the assumption that deskewed text regions have a homogeneous horizontal structure and that the document image consists almost entirely of text. In such case, the algorithms will select from the image several vertical strips, distributed equidistantly [295] or randomly [208] along the horizontal direction. Subsequently, for each pair of vertical stripes, the algorithm tries to maximize their cross-correlation by shifting one of them along the vertical direction. The obtained vertical shift s is then used to compute an estimate for the document skew angle θ via the formula: $\theta = \arctan(s/d)$, where d represents the horizontal distance between the two stripes considered. Finally, the global skew will be given by the mean or median of the estimated skew angles. The advantages of cross-correlation methods are their direct applicability on grayscale or color images, as well as the fact that they require no prior connected component identification.

Morphology-based skew detection algorithms try to extract text lines by repeated application of the standard morphological operators. For example, the algorithm of Chen et al. [74] recursively applies the closing operation on the document image. The structuring element of the morphological operator depends on the expected skew angle range. Afterwards, the opening operation is applied recursively on the resulting image, until a stopping criterion is met. At this point, ideally the text lines are represented by elongated connected components whose principal direction can be estimated. In general, care must be taken to avoid taking into consideration too many spurious connected components, resulting from images, graphics or noise. As a final step, the text lines (connected components) whose directions are near the median direction are selected and the global skew is estimated from the selected components. Morphology-based algorithms have the advantage that they can be applied directly to grayscale images, however the computational cost entailed by the repeated application of morphological operators is generally quite high.

New skew detection methods still continue to appear every year in the specialized literature. As observed by Cattoni et al. [66], there are two main causes for the continued research work, namely the need for accurate and computationally efficient algorithms on one hand and the relaxation of the requirements on the input documents on the other. At this point, it is important to note that although the number of publications in this area is indeed significant, virtually all the aforementioned algorithms only work reliably for a very limited skew angle range. Most of them are limited to the interval [-45°, 45°], while a few are capable of coping with the larger (-90°, 90°) interval. For the processing of large document collections however, such limited skew angle detection as offered by the classical algorithms is insufficient, as documents may come in any possible orientation.

In order to cope with this issue, *orientation detection* algorithms have been developed as a complement to classical skew detection. In contrast to skew detection techniques, methods for orientation detection rely heavily upon script-specific features to determine the most likely orientation. Thus most orientation detection algorithms either specifically require an apriori knowledge of the script of the document, or they also detect the script as part of the process. The specialized literature contains much fewer document orientation detection algorithms, presumably because of the difficulty of finding robust, yet simple and at least script-generic features.

One of the earliest methods capable of distinguishing between portrait and landscape orientations was proposed by Akiyama et al. [24] in 1990 for Japanese documents. About the same time, algorithms using the ascender-to-descender ratio for Roman script documents were introduced for detecting the up/down orientation [58, 253]. In the recent years a few algorithms capable of detecting both skew and orientation for Roman script document images have appeared [46, 277]. Their skew detection part falls in the class of nearest neighbor approaches and the orientation is detected based on the ascender-to-descender statistic. For documents making heavy use of majuscules (i.e. no ascenders or descenders are present) or written in scripts other than Roman (e.g. Pashto, Hindi, Arabic) this statistic is not applicable. Thus, a new decision criterion for orientation detection was recently introduced by Aradhye [40]: the opening direction of the characters. Even with the new criterion, one special situation remains where all proposed orientation detection methods still fail, namely pages consisting (almost) exclusively of tables containing numerical data, e.g. financial data. In such cases, with all characters having the same height as well as no significant trend regading the direction of the openings it remains an unanswered question as how to efficiently determine the page orientation (without resorting to the computationally expensive OCR).

4.1.2 Multiple- and Non-uniform Skew

In the previous section we have described the main classes of algorithms for global skew detection and have seen that there exist many algorithms capable of performing this task with various degrees of success, while exhibiting specific limitations. For real-life documents however, computing a single skew angle per document page is in some situations an overly strong restriction (see figure 4.1). Both historical documents as well as modern magazines exist featuring a layout with whole text paragraphs having a different skew or even orientation. This layout feature is intentional and meant to better focus the reader's attention on the respective regions or in some cases, to provide additional optional information. A few of the surveys previously mentioned also briefly cover *multiple skew detection* [46, 48, 66, 134].

In one of the earliest methods on multiple skew detection, Antonacopoulos [28] addresses the problem of converting a generic global skew detection algorithm into a local one (thus making it suitable for documents exhibiting multiple text skew angles). Without a loss of generality, the author considers that a page segmentation step has already been applied (with or without a global skew detection before it). Local skew detection is then applied independently for the individual text regions. Using the aforementioned idea, practically all global skew detection algorithms, including all those mentioned in the previous section, can be adapted to work on multiple skew documents. It is worth noting that the skew adaptation framework assumes that the region segmentation step is by itself capable of differentiating between regions featuring different skews/orientations. As we will see in section 5.1.2, there indeed exist segmentation algorithms which can successfully accomplish this task, most notably those relying on texture analysis or Voronoi diagrams. Because of the adaptation framework the number of algorithms proposed specifically for multiple skew detection is very low.

Most notably, Antonacoupoulos [28] introduces an innovative approach using white tiles to estimate the skew of individual text regions. White tiles are defined as the widest background areas that can be accurately represented by rectangles. The core idea of the algorithm is that the space that lies between characters on the same line as well as between two adjacent text lines is also arranged in parallel lines of the same orientation. For text regions where a reasonable number of white tiles can be extracted, the author claims a good accuracy (error $< 0.1°$). Problems may appear for short paragraphs (e.g. 1–2 text lines) where the set of tiles in the white space cover has a low cardinality.

In contrast to multiple skew, *non-uniform skew* is normally caused by inaccuracies in the digitization process and/or the physical degradation of the documents. In the earlier years of document analysis, non-uniform skew was mostly treated as a 2-D problem but more mature methods have appeared recently treating the document scans as proper 3-D surfaces and allowing for perspective distortions. In case the deformations of the image are significant, non-uniform skew detection and correction is commonly known as *dewarping.*

Spitz [251] addresses a common problem with scanners using a sheet feed mechanism: an uneven feed rate leading to a deformed page aspect. The technique introduced by Spitz is capable of handling compressed domain images, more specifically images compressed using a CCITT Group 4 coding scheme (e.g. faxes, lossless TIFFs). Interestingly, the author does not require a prior segmentation of the text regions and instead divides the document image into a resolution-dependent number of horizontal stripes of constant height. Using the stripes, a vertical skew profile of the page is computed and subsequently used to cut, rotate and reassemble the image. As observed in the original paper, this approach has the advantage of simplicity and speed, however it may not produce optimal skew detection and correction results in some circumstances.

Ezaki et al. [99] still essentially treat the document image as a 2-dimensional surface and construct a warping model of the document image via a set of cubic splines. Each spline is fitted robustly to either a the text lines or an inter-line space. The spline parameters are optimized globally over the whole document using a dynamic programming approach. This allows its application on documents containing only short formulae, small figures and little text.

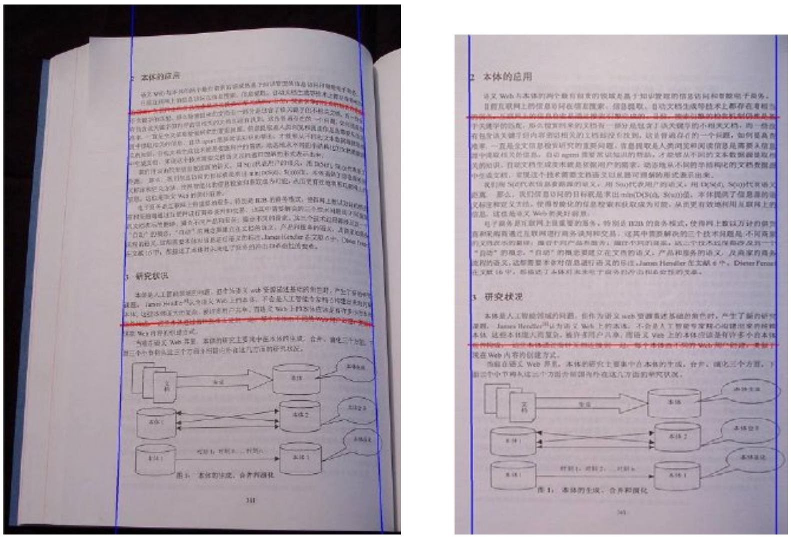


Figure 4.3: Document image before and after rectification using the Wu et al. [287] method. Source: [287]

The Document Image Dewarping Contest [238] held in 2007 as part of the camera-based document analysis and recognition workshop (CBDAR) compares several methods for document dewarping on a larger dataset consisting of 102 images captured with a handheld camera. The winning algrithm of Wu et al. [287] is one of the first algorithms to explicitly model document scans as fully-fledged 3-dimensional surfaces. Just as previously introduced by Dance [83], the method of Wu et al. estimates the perspective parameters by using the text lines and the left and right column boundaries (see fig. 4.3). In addition, Wu et al. take into account the surface curvature by explicitly modeling the appearance of the document (here: the top page from a bound book) as a parametrized function. Finally, for document pages exhibiting pronounced shadows new methods for geometric and photometric distortion modeling using a shape-from-shading and in-painting approach have recently been proposed [300]. Such images are often the result of using a strong flash close to the physical page – a typical case for modern mobile phone cameras.

It is interesting to note that practically all methods for document dewarping use the OCR accuracy as evaluation measure, either directly or indirectly via the Levenshtein (edit) distance to the ground truth. In case of the techniques taking part in the Document Image Dewarping Contest, a mean edit distance of less than 1% was achieved by using Omnipage Pro 14.0, a commercial OCR software. Unfortunately, OCR accuracy as evaluation measure has the great disadvantage of being OCR engine-specific, not allowing an accurate comparison of the performance of dewarping algorithms which already produce very good OCR results (as we have seen, many actually do), as well as offering no clue about how well the algorithms perform in non-textual areas. Only very recently, in 2011, Bukhari et al. [63] have introduced a more comprehensive image based evaluation measure which addresses these limitations. They compute a score between a ground truth image and the dewarped results using the ratio of matching SIFT features and the matching error as the mean of the vectorial distances between the matching SIFT descriptors. They use the novel methodology to more accurately rank the methods from the Document Image Dewarping Contest, together with a newer active contour (snake) based book dewarping method [62].

4.2 Seamless Skew and Orientation Detection

In the current section we present two new algorithms for global skew detection using the same generic framework. The algorithms belong to the class of nearest neighbor approaches and use connected components as basic building blocks. Since connected components can be computed for any kind of color space of the input image (given an appropriate distance function), the proposed framework may be applied directly to any digital document image.

We first introduce a basic, layout- and script-independent algorithm which is able to deal with document images having a global skew angle within [−90°, 90°]. A more robust version of the algorithm is then introduced as a specialization for Roman script documents. The improved algorithm can deal with document skew from the full angle range of −180° to 180°, while at the same time exhibiting a better accuracy. Both algorithms are very fast, their runtime speed being directly comparable to that of the early projection profile-based methods [50].

4.2.1 Layout- and Script-Independent Skew Detection

In this section we describe a generic, script-independent skew detection algorithm capable of handling documents exhibiting global skew angles from the range [−90°, 90°]. The proposed framework is afterward entirely reused in the more advanced version of the algorithm, capable of detecting both skew and orientation for Roman-script documents and which is presented in the next section.

In order to be able to robustly deal with generic document images containing non-text regions of significant size, the first step consists of a basic filtering of potential characters/character parts from non-text content (incl. noise). Note that for document images known not to contain significant noise or any large halftones or graphics this step can be skipped entirely. We offer formal proof of this assertion in section 4.2.3, as well as a modality of computing the degree to which the result accuracy depends upon the ratio of remaining non-text components to the number of text components.

The foremost purpose of the filtering is to reduce the amount of small noise in the image, be it impulse-like or small connected components from large dithered halftones/drawings. More specifically, we denote as “large” halftones and graphics those regions whose total number of connected components is on the same or a higher magnitude order with that of the components located within the textual regions from the same document page. Such large non-text components and/or high noise levels are extremely rare in practice in any professional scans – for example, in the UW-I [215] test set consisting of 979 scanned test images (many showing parts of the adjacent page) a single image falls into this category.

Two simple filtering rules were applied for each connected component throughout all our experiments:

1. The width and height do not differ by more than a factor of 10. The relatively large threshold ensures that thin letters (such as “i” or “l”) are kept, along with any components consisting of a few wrongly merged characters (possibly as an artifact of the binarization process). Note that this condition was found to be optional in the performed tests, as its effect is almost unnoticeable under normal circumstances.

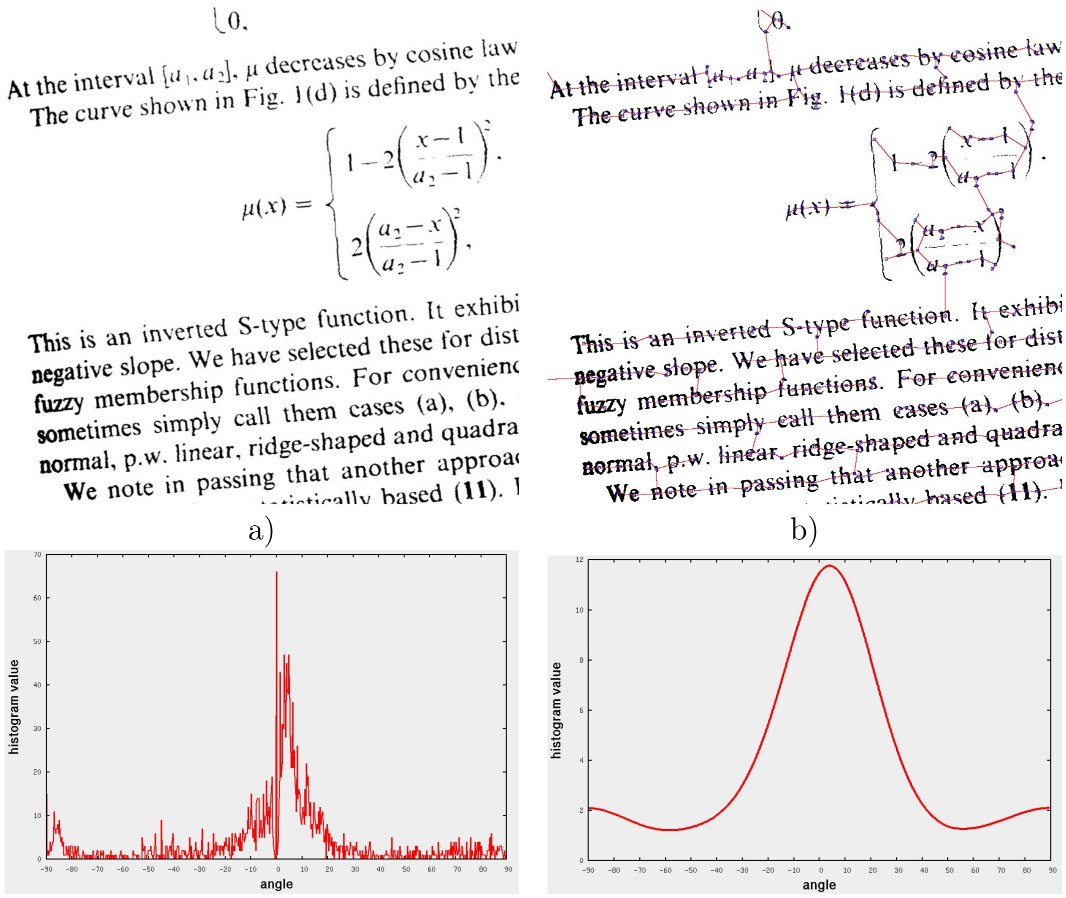
2. The width and height are larger than certain thresholds. The thresholds in our case were set to 50% of the size of the dominant character width/height on the page, determined as described in [155]. In this way the thresholds are resolution-independent. Note that since neither the orientation nor the skew of the page are known at this stage, the width and height thresholds must also be used interchanged in the condition.

The reader may observe that both filtering conditions are very generic and can be applied to any kind of digital document images (including most script types in use nowadays).

Next, we take the centers of the bounding boxes of the filtered connected components and compute the Euclidean minimum spanning tree (MST) of this set of points. It was experimentally determined that for an unrestricted skew range, the center points provide a more robust estimate of the global skew than both the upper and lower mid-points of the bounding boxes. The Euclidean MST for a set of points can be computed either directly [18] or indirectly by using the property that the MST edges are a subset of the edges of the Delaunay triangulation [122] and determining the respective subset by any standard MST algorithm, such as that of Kruskal. A worst-case running time of $O(n \log n)$, where n represents the number of points, is achievable by using either variant.

A binned histogram spanning the angle range $[-90°, 90°)$ is computed from the skew angles of the MST edges. In our experiments we have used a bin size of 0.1°, as skew angles lower than this value are practically indistinguishable by humans.

The final step is the detection of the skew angle from the computed histogram. Unlike other binned histogram-based methods [50] we do not employ an iterative coarse-to-fine search method, as we have found it to be too sensitive at larger skew angles (i.e. $\geq$15–20°). Instead, we assume that the skew angle errors are normally distributed around the global skew angle and convolve the histogram circularly with a Gaussian mask. Thus, the location of the maximum value in the result of the circular convolution is expected to correspond to the desired skew angle. In our experiments we have determined that a Gaussian mask diameter equal to the number of bins corresponding to 90° in the histogram performs well. For obtaining a higher/lower accuracy, the histogram bin size can easily be reduced/extended, having a direct influence on the running time of the algorithm. This is especially visible for very fine-grained histograms, where the $O(n^2)$ convolution time will be the dominant factor in the overall running time of the algorithm.



a) b)

c) d)

Figure 4.4: (a) Portion of original document scan; (b) Same scan with superimposed Euclidean MST edges constructed from the connected components' midpoints; (c) Corresponding binned histogram of the MST skew angles; (d) Histogram convolution result with a Gaussian mask spanning 90°

At this point it is important to note that the algorithm presented in this section is entirely script-independent, since its sole indirect assumption is that character spacing is usually smaller than line spacing. An additional advantage of using an MST and histogram-based approach is that the influence of page portions featuring a multiple- or non-uniform skew is practically non-existent. The Euclidean MST basically “models” text lines as piecewise linear functions as opposed to the more simple and error-prone linear model employed by all known previous approaches.

Using the notation from algorithm listing 2, the components forming the common framework are:

- Filtering of connected components by size (optional) – $Filter(CCList)$
- Euclidean minimum spanning tree of a set of planar points – $EMST(P)$
- Binned histogram of given accuracy from a set of 1-d values – $BinHist(Values, RangeStart, RangeEnd, BinSize)$
- 1-Dimensional circular convolution with a discrete Gaussian kernel – $GaussCircConv(InputArrray, WindowDiameter)$

Algorithm 2 Generic skew detection algorithm

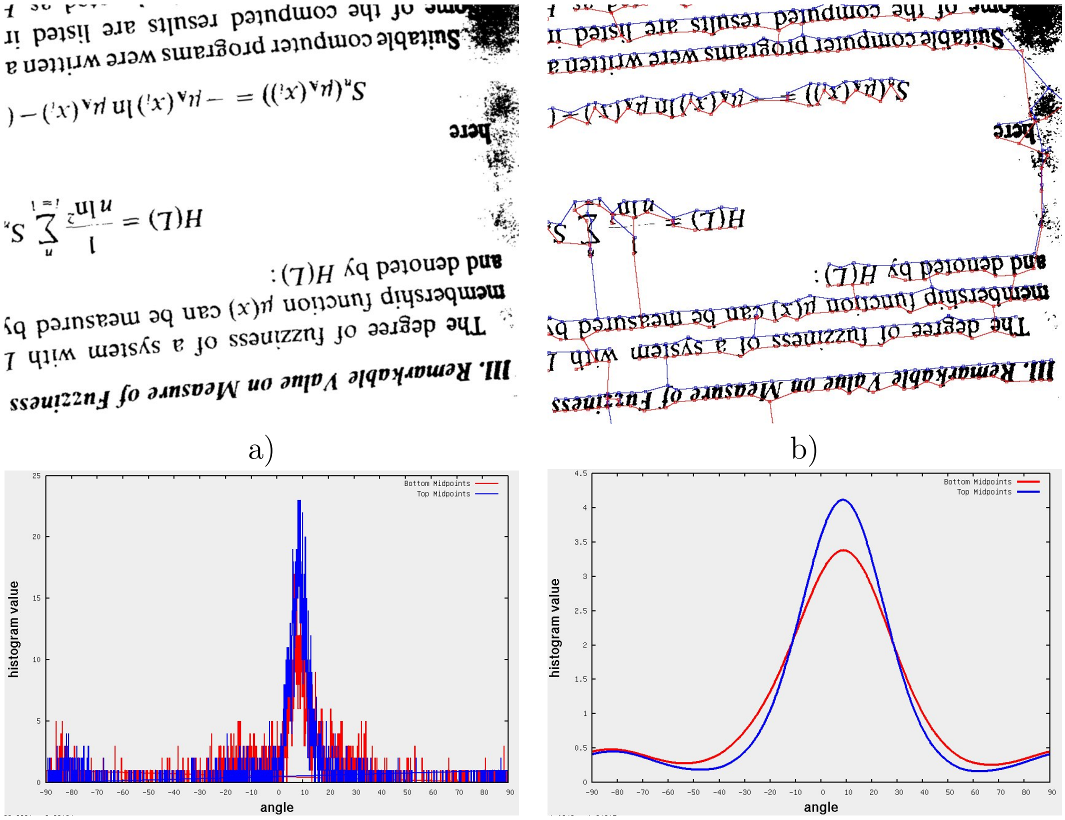
```
Require: binary document image $I$, histogram bin size $bsize$ (default 0.1), dominant
  character height $dch$ as in [1]
Ensure: global skew angle $\alpha \in$ [-90°, 90°); confidence measure for spacing $C_{\text{spc}}$
  $CC \leftarrow Filter(ConnectedComponentLabeling(I), dch)$
  for $i = 1$ to $N_{cc}$ do
    $CenterPoints_i.x \leftarrow CC_i.BoundingBox.Center.x$
    $CenterPoints_i.y \leftarrow CC_i.BoundingBox.Center.y$
  end for
  $EdgeList \leftarrow EMST(CenterPoints)$
  for $i = 1$ to $N_{edges}$ do
    Add $angle(EdgeList_i)$ to $Angles$
  end for
  $H_{conv} \leftarrow GaussCircConv($
                  $BinHist(Angles, -90, 90, bsize), 90/bsize)$
  $C_{\text{spc}} = 1 - \frac{H_{conv}(argmax(H_{conv})+90°)}{max(H_{conv})}$
  Return $argmax(H_{conv})$
```

4.2.2 Seamless Orientation Detection for Roman Script Documents

To the best of the author's knowledge, the idea of using an Euclidean MST for determining the orientation of text lines was first introduced by Ittner and Baird in [140]. However, due of the fact that their method only made use of the center points for each connected component, it was inherently unable to differentiate between top-left and top-right orientations.

In contrast to Ittner and Baird's method and the previously presented algorithm, the method proposed in the following is specific to Roman script documents. More exactly, it makes indirect use of the fact that in a typical text the number of descenders is lower than the number of ascenders. This holds true for many languages having Roman script, such as English, German, Spanish and French. One may easily verify this fact from existing tables containing character frequencies for each language, such as the ones for the English language in [40].

As a first step of the improved algorithm we use the result provided by the basic algorithm as an approximation of the actual page skew. Next, for each of the filtered connected components we compute the top-left and bottom-right coordinates of its bounding box in the document image deskewed using the approximated skew. This can be accomplished by applying the appropriate, pre-computed rotation matrix to each pixel within a connected component. Note that it is not actually necessary to consider each pixel of a component, just the pixels located on its border. The processing time gain by applying this trick has been found to be insignificant, however. We now compute two sets of points, namely the top and bottom mid-points for each component. Afterward, the same processing as in the basic algorithm is applied on each of the two sets of points and two histograms convolved with Gaussian masks are obtained. One may now make use of the ascender-to-descender ratio property to conclude that the histogram containing the higher maximum value corresponds to the actual alignment points (bottom mid-points) of the characters on the page. This holds true because the histogram of the alignment point edges has more angles close to the page skew and their distribution will naturally feature a taller and sharper peak. Thus the correct up/down orientation of the page is determined. Finally,



a) b)

c) d)

Figure 4.5: (a) Portion of original document scan; (b) Same scan with superimposed Euclidean MST edges constructed from the alignment and top-mid points of the connected components, respectively; (c) Corresponding binned histograms of the MST skew angles; (d) Histogram convolution result with a Gaussian mask spanning 90°

we can improve the initial approximation of the page skew by selecting instead the bin containing the maximum value from the histogram of approximated alignment points as the one corresponding to the desired result.

Algorithm 3 Enhanced skew and orientation detection algorithm

```
Require: binary document image $I$, histogram bin size $bsize$ (default 0.1), dominant character height $dch$ as in [155]
Ensure: global skew angle $\alpha \in$ [-180°, 180°); confidence measure for spacing $C_{\text{spc}}$; confidence measure for validity of ascender-to-descender ratio $C_{\text{ad}}$
  $CC \leftarrow Filter(ConnectedComponentLabeling(I), dch)$
  $\alpha \leftarrow BasicSkewDetection(I, bsize, dch)$
  for $i = 1$ to $N_{cc}$ do
    $RotCC \leftarrow RotateAroundCenterPoint(CC_i, -\alpha, I.Center)$
    $MidTopPoints_i.x \leftarrow RotCC.BoundingBox.Center.x$
    $MidTopPoints_i.y \leftarrow RotCC.BoundingBox.Top.y$
    $MidBottomPoints_i.x \leftarrow RotCC.BoundingBox.Center.x$
    $MidBottomPoints_i.y \leftarrow RotCC.BoundingBox.Bottom.y$
  end for
  $EdgeListTop \leftarrow EMST(MidTopPoints)$
  for $i = 1$ to $N_{edgesTop}$ do
    Add $angle(EdgeListTop_i)$ to $AnglesTop$
  end for
  $H_{convTop} \leftarrow GaussCircConv($
            $BinHist(AnglesTop, -90, 90, bsize), 90/bsize)$
  $EdgeListBottom \leftarrow EMST(MidBottomPoints)$
  for $i = 1$ to $N_{edgesBottom}$ do
    Add $angle(EdgeListBottm_i)$ to $AnglesBottom$
  end for
  $H_{convBottom} \leftarrow GaussCircConv($
            $BinHist(AnglesBottom, -90, 90, bsize), 90/bsize)$
  $\alpha \leftarrow argmax(H_{convBottom})$
  if $max(H_{convTop}) > max(H_{convBottom})$ then
    $\alpha \leftarrow argmax(H_{convTop})$
    $C_{\text{spc}} = 1 - \frac{H_{convTop}(argmax(H_{convTop})+90^\circ)}{max(H_{convTop})}$
    {always make sure result lies within [-180°, 180°)}
    if $\alpha \geq 0$ then
      $\alpha \leftarrow \alpha - 180$
    else
      $\alpha \leftarrow \alpha + 180$
    end if
  end if
  $C_{\text{ad}} = 1 - \frac{min(max(H_{convTop}), max(H_{convBottom}))}{max(max(H_{convTop}), max(H_{convBottom}))}$
  Return $\alpha$, $C_{\text{spc}}$, $C_{\text{ad}}$
```

4.2.3 Theoretical Basis

After having described the steps of both skew detection algorithms, we are now able to provide a formal foundation in order to better understand the individual steps and their role.

At first let us consider an input image consisting exclusively of text, that is the document is completely noise-free and contains no non-text (e.g. graphics, halftones, formulae, etc.) regions. More formally, we can express this as $|E_{doc}| = |E_t|$, where E_{doc} is the set of edges from the Euclidean MST constructed using all connected components in the document and E_t represents the set of edges in the Euclidean MST constructed using only the text connected components. We are interested in computing the global skew α_{doc} of the document, which we define as the average skew of the text lines, denoted α_{line}. Considering the MST edges as independent variables X_i, $i = \overline{1, |E_{doc}|}$ we are therefore looking for the best approximation for $\frac{1}{|E_{doc}|}\sum_{i=1}^{|E_{doc}|}\alpha(E_i)$.

The central limit theorem [104] states that the probability distribution of any statistic of independent and identically distributed random variables will converge to a normal distribution as the sample size approaches infinity. As seen in the previous section, in our case the population consists of the set of edges from the Euclidean MST built using the mid-/alignment/top-mid points of the connected components extracted from a document page. The statistic of interest is the mean angle of the vectors. As such, once the number of observed vectors is high enough, its probability distribution will converge to a Gaussian curve (see also Gaussian function description from section 2.1.1):

$$\frac{1}{|E_{doc}|}\sum_{i=1}^{|E_{doc}|}\alpha(E_i) \xrightarrow{d} \mathcal{N}(\alpha_{line}, \frac{1}{|E_t|}\sigma^2_{\alpha_t}) \tag{4.1}$$

Note that, regardless of the distribution of the initial population, the probability distribution of the mean angle is still normal. Thus, our restriction that the population consists solely of Euclidean MST edges is completely irrelevant to the shape of the distribution. The restriction has a different purpose however: to ensure that the mean angle of the vectors converges to the global skew angle of the document. The latter convergence can only be ensured when the majority of the vectors indeed approximate the global skew, which equals by definition the average skew of the *text lines*. In other words, most vectors must lie between connected components located on the same text line. It is precisely at this point that our additional requirement that the inter-line distance is higher than the inter-character distance comes into play. By ensuring this, we have the certainty that majority of the Euclidean MST edges will indeed connect those components located on the same line. The reasoning for this is straightforward: since characters on the same line do *on average* lie closer than characters located on different text lines, such edges will more likely minimize the total cost (=distance) of connecting all input points and consequently they will be the ones selected as part of the Euclidean MST:

$$D_{inter-character} < D_{inter-line} \Rightarrow \lim_{|E_t|\to\infty} \alpha_{line} \to \alpha_t$$

Let us consider now the case in which the input document image also contains non-text layout elements, e.g. halftones, line-art, separators, etc.. These elements will further on be generically referred to as graphic elements. In this situation we have:

$$|E_{doc}| = |E_t| + |E_g| + |E_{tg}|,$$

where $|E_g|$ denotes the set of edges where each joins 2 graphical elements and $|E_{tg}|$ denotes the set of edges each joining one graphical element and one text element. Because graphical elements are generally contiguous independent areas, we know that

$$\left.\begin{array}{l}|E_{tg}| \ll |E_t| \\ |E_{tg}| \ll |E_g|\end{array}\right\} \Rightarrow |E_{doc}| \approx |E_t| + |E_g| \tag{4.2}$$

Additionally, the restriction regarding the inter-line distance is irrelevant for graphical elements and as such the average angle of edges joining graphical elements will converge to zero. Note that this only happens under the usual assumption of a uniform or random normal distribution for the graphical edges. In short, for documents containing both text and graphical elements we can now write:

$$\begin{aligned}&\frac{1}{|E_{doc}|}\sum_{i=1}^{|E_{doc}|}\alpha(E_i) = \frac{1}{|E_{doc}|}(\sum_{i=1}^{|E_t|}\alpha(E_t(i)) + \sum_{i=1}^{|E_g|}\alpha(E_g(i))) \xrightarrow{d} \\ &\frac{1}{|E_{doc}|}(|E_t|\mathcal{N}(\alpha_t,\ \frac{1}{|E_t|}\sigma^2_{\alpha_t}) + |E_g|\mathcal{N}(0,\ \frac{1}{|E_g|}\sigma^2_{\alpha_g})) = \\ &\frac{|E_t|}{|E_{doc}|}\mathcal{N}(\alpha_t,\ \frac{1}{|E_t|}\sigma^2_{\alpha_t} + \frac{|E_g|}{|E_t|^2}\sigma^2_{\alpha_g})\end{aligned} \tag{4.3}$$

The only situation remaining to consider is when the input document image contains noise, in addition to text and graphics. Unfortunately, unlike graphics or text, noise areas are not only contiguous but may also be found interweaved within other regions. The distinction between graphical elements and noise elements located within the same contiguous area is irrelevant in our case, especially since both types of elements are usually considered to be zero-centered, uniformly or normally-distributed. Equation (4.2) now becomes:

$$|E_{gn}| \approx 0 \ \Rightarrow\ |E_{doc}| \approx |E_t| + |E_g| + |E_n| + |E_{nt}| \tag{4.4}$$

As in the case of graphical elements, the average angle of edges joining 2 noise elements or one noise element with a text element will converge to zero (again, under the classic assumption of zero-centered noise). Finally, based on equation (4.4) for documents containing text, graphics as well as noise components the following holds:

$$\begin{aligned}&\frac{1}{|E_{doc}|}\sum_{i=1}^{|E_{doc}|}\alpha(E_i) \xrightarrow{d} \\ &\frac{|E_t|}{|E_{doc}|}\mathcal{N}(\alpha_t,\ \frac{1}{|E_t|}\sigma^2_{\alpha_t} + \frac{|E_g|}{|E_t|^2}\sigma^2_{\alpha_g} + \frac{|E_n|}{|E_t|^2}\sigma^2_{\alpha_n} + \frac{|E_{nt}|}{|E_t|^2}\sigma^2_{\alpha_{nt}})\end{aligned} \tag{4.5}$$

In the following we discuss how the presented generic algorithm can be derived from formula (4.5). One may clearly see that noise has a twofold influence on the accuracy of the determined skew angle of the document. Therefore, minimizing the *number* of noise components taken into account in the construction of the Euclidean MST will produce the largest immediate accuracy improvement. In accordance to this observation, the removal of noise components is accomplished right at the start of the algorithm by the generic filtering step described in the previous section. Noise component filtering also has another reason apparent from formula (4.5): the required estimations for the number of edges belonging to each class (i.e. text, graphics or noise) are in general not available or cannot be directly computed at this early stage. Instead, by ensuring that $|E_t| \approx |E_{doc}|$ (which implies that $|E_g| \ll |E_t|$, $|E_n| \ll |E_t|$ and $|E_{nt}| \ll |E_t|$) we are able to obtain a close enough approximation of the global skew distribution.

Next, for practical purposes we may substitute $|E|$ with $|V|$, where V represents the set of vertices from the respective tree. This is possible since generally in any tree $|E_{tree}| = |V_{tree}| + 1$ and the cardinalities of the sets we are usually working with are at least in the order of 10^3. Additionally, due to the large cardinalities of the sets of connected components we may also safely approximate the means with medians:

$$\sum_{i=1}^{N} \frac{x_i}{N} \approx x_{\lfloor N/2 \rfloor} \approx Median(x_i),\ where\ x_i \leq x_{i+1}, \forall i \in [1, N] \tag{4.6}$$

This allows us to construct the edge angle histogram using the original edge angles, instead of average angles for all edge pairs within the Euclidean MST. The construction of the edge angle histogram represents the second step of the proposed skew detection algorithm.

A convolution with a Gaussian of known parameters (default: $\mathcal{N}(0,\ 15°)$) is afterward employed. It is well-known that the convolution operation with a certain mask produces the maximum response at locations which are spatially most similar to the mask. In fact this represents the basis of many template matching methods. As such, convolution with a Gaussian mask provides us with an elegant solution to the practical difficulty of computing a robust value for the expectation of the skew angle distribution from its rough histogram approximation. Note that the expectation value remains unchanged by this operation, as convolution is equivalent to the addition with a Gaussian, which in our case is known to have a zero mean. The proposed variance of the Gaussian convolution mask corresponds to a digital window approximation of diameter 90° and was experimentally determined to offer a good compromise between accuracy and noise removal strength. The other important consideration at this point is the existence of noise in the average skew angle samples. One cannot realistically expect the initial generic noise component filtering to produce near-perfect results on all types of document images. But luckily this is exactly what the Gaussian kernel is most well-known for: its excellent noise reduction properties (low-pass filtering). So, in conclusion the (circular) convolution with a Gaussian kernel provides the optimal solution for two issues in our case: robust detection of the skew angle distribution peak and noise attenuation. The interested reader may find more information about the useful properties of Gaussians in image enhancement in section 2.1.1.

Finally, since we require a single value for the global skew angle, we are now able to compute the one having the highest probability of being the right one. It is known that the expectation of the normal distribution has the highest probability value. Extracting the global skew at this point amounts to a straightforward search for the maximum value in the distribution histogram and converting its index into the right angular value (using the histogram bin resolution).

4.2.4 Algorithm Assumptions and Confidence Measures

The theoretical analysis of the algorithms from the previous section has provided us with formula (4.5), which allows us in theory to exactly estimate the uncertainty of the detected global skew angle. The keyword here is "in theory", because as already mentioned most of the necessary parameters cannot be reliably estimated at this stage. However, the model parameters can be estimated well after the page segmentation step. As such, an interesting idea to pursue would be an iterative processing architecture in which skew detection and page segmentation are executed as many times as necessary until a fixed certainty threshold for the global skew determination has been reached. While undoubtedly of theoretical interest, such an approach would be much too computationally expensive for application in current mass digitization projects.

In this section we describe a straightforward way of obtaining a meaningful confidence estimate for the computed skew angle without resorting to knowledge from later processing stages. It is important to state from the beginning that the proposed confidence estimates do not have a purely mathematical basis and are not meant to be as accurate as those which can be obtained from formula (4.5). In contrast, our goal is solely to obtain a value from a pre-defined, document-independent interval which directly reflects the certainty that the computed global skew angle for a document is indeed close to its actual global skew. Even such a confidence value is extremely useful for practical purposes, where it is simply not feasible to screen all input documents in order to find out which ones do satisfy the preconditions for our algorithms. In this context, confidence estimates can be used to significantly reduce the portion of documents in need of correctness checking by a human operator.

Let us go over the implicit and explicit assumptions made by the proposed algorithms in their order of appearance. First, we require as input a sequence of all connected components on the document image. We approximate the non-noise components by their center/ alignment/ top-mid points and construct the Euclidean MST of the point set. An implicit assumption from this point forward in both algorithms is that the number of MST edges approaches infinity, i.e. it is high enough to warrant the application of the central limit theorem and provide an actual distribution which resembles the theoretical normal one. Since this assumption is crucial for the entire functioning of the algorithm, we set a minimum threshold of 30 non-noise components below which the returned confidence has its minimum value. Exact relations between the number of samples and the reliability of the outputs were found by Chang et al. [70, 71] for a few situations, so it appears to be possible to derive such formulae in more cases. For practical purposes having an exact formula modeling this relationship is not strictly necessary, however it is an interesting direction to explore in future work.

An explicit assumption of both algorithms is that the character spacing is on average lower than the line spacing. This guarantees the convergence of the MST edge skew angle to the text line skew, as opposed to the layout column skew (defined by the distances between lines). In figure 4.4 (*d*) one may see a typical example of an output histogram distribution – bimodal with the maxima separated by approx. 90°, corresponding to the text line skew and the layout column skew, respectively. The ratio between the average line spacing and the average character spacing is directly reflected by the ratio of the two modes. Consequently, the degree to which the spacing assumption holds can immediately be extracted via the aforementioned ratio. More specifically, a ratio near 1 reflects that the assumption most likely does not hold, whereas a value closer to 0 reflects a high confidence in the validity of the assumption. Using the notation from algorithm 2, the spacing-related confidence value C_{spc} can be defined as:

$$C_{\text{spc}} = 1 - \frac{H_{conv}(argmax(H_{conv}) + 90^\circ)}{max(H_{conv})}$$

More accurately, the second mode of the circular histogram (corresponding to the layout column skew) can be found by searching for the local maximum located closest to $argmax(H_{conv}) + 90^\circ$, as opposed to directly using the histogram value at this location.

One must observe that while the ratio feature works generally well, in cases in which the line spacing is clearly lower than the character spacing it will provide a wrong indication. In practice however, we have not observed such a situation yet and it remains an open question whether there indeed exist scripts/documents where this does happen.

For the basic algorithm which restricts itself to the determination of the skew only, there are no other explicit or implicit assumptions. In the Roman script specialized algorithm however, there exists another explicit assumption: that the ratio of the number of ascenders to the number of descenders is higher than 1. This holds true for many, if not all Roman script languages, such as English, German, Spanish and French. The property can be readily verified by checking the character frequency language-specific tables computed on large text corpora, such as the ones for English [40] or German [57]. It is interesting to note that a important application area of letter frequency tables is in cryptography [57]. The caveat with this feature lies in its generality – while it is very likely to be true for long texts, in case of single document pages with just a few lines of text it may not be stable enough. The ascender-to-descender ratio feature also implicitly assumes another important fact: that the page in question does indeed feature minuscules, i.e. it was not entirely printed using capitals or contains only numbers. For such pages, one must resort to other specialized orientation detection algorithms, such as those proposed by Aradhye [40]. Fortunately, the computed histogram-based approximations of the probability distributions for the global skew angle allow us to detect this kind of situations. In figure 4.5 *(d)* we see a typical example of the computed text line skew distributions. The ratio between the expectations for the global skew of the two point sets constitutes a robust feature for gaging the validity of the ascender-to-descender ratio in the current document. Using the notation from algorithm 3, the ascender-to-descender confidence value C_{ad} is computed as:

$$C_{\text{ad}} = 1 - \frac{min(max(H_{convTop}), max(H_{convBottom}))}{max(max(H_{convTop}), max(H_{convBottom}))}$$

Just like for the previous confidence feature, a value close to 1 indicates a most likely valid assumption, whereas a value close to 0 indicates a high probability that the assumption does not hold.

Since in the case of the basic skew detection algorithm we can only make use of confidence feature C_{spc}, we currently use a hard threshold set to 0.5 and accept all skew detection results with spacing confidence features with a value above or equal to this threshold. For the specialized algorithm, we use in addition the confidence feature C_{ad} for gaging the orientation detection success. The fixed threshold for the minimum ascender-to-descender confidence feature determined from our experiments is set to 0.1. As an alternative, one may also use a weighted linear combination of the confidence features as an overall confidence indicator:

$$C_{\text{tot}} = \alpha C_{\text{spc}} + (1 - \alpha) C_{\text{ad}},\ \alpha \in [0,\ 1]$$

A method for determining a proper generic weighting (α) for the two features remains as topic for future work.

4.2.5 Evaluation

For testing the orientation detection accuracy of our algorithm, we have used 5 different test sets. In all our experiments we have differentiated between 4 different orientations: top-up, top-down (180°), top-left (90° counter-clockwise) and top-right (90° clockwise).

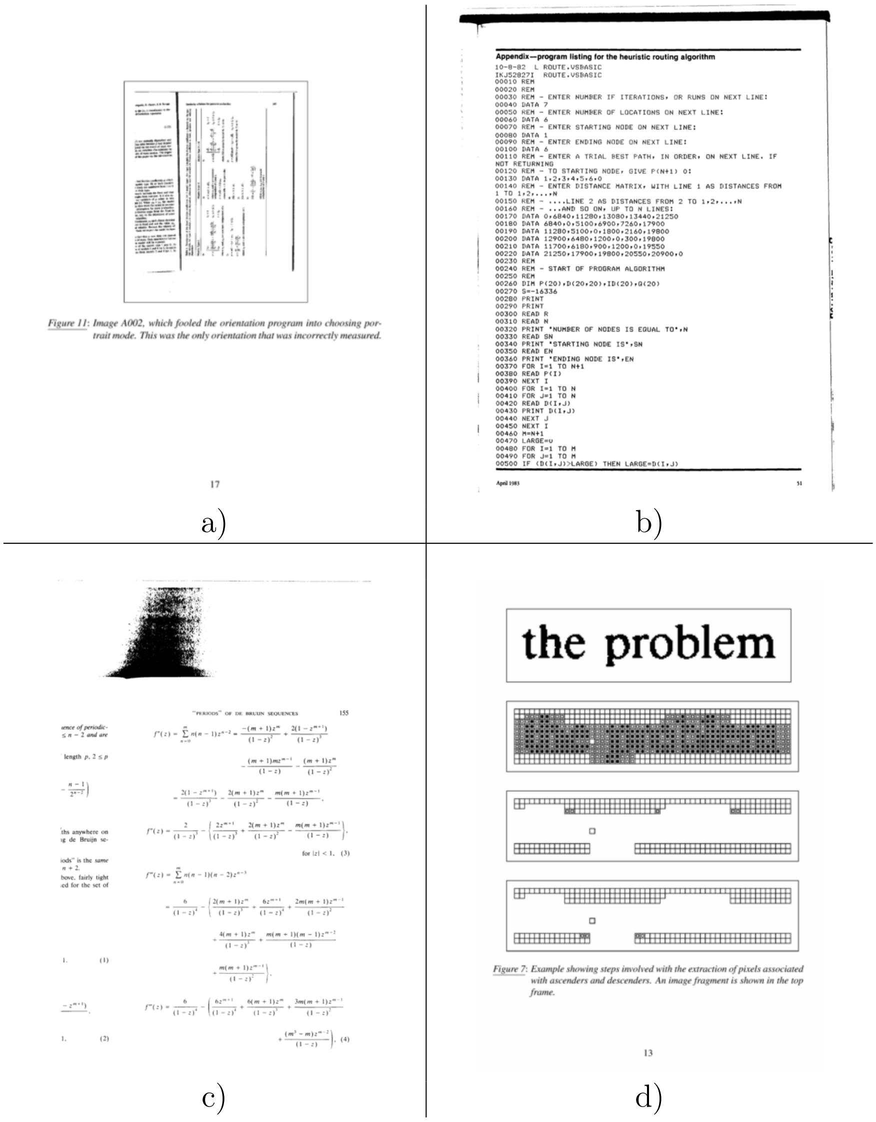


Figure 11: Image A002, which fooled the orientation program into choosing portrait mode. This was the only orientation that was incorrectly measured.

17

a)

Appendix—program listing for the heuristic routing algorithm

```
10-8-82  L ROUTE.VSBASIC
IKJ52827I  ROUTE.VSBASIC
00010 REM
00020 REM
00030 REM - ENTER NUMBER IF ITERATIONS, OR RUNS ON NEXT LINE:
00040 DATA 7
00050 REM - ENTER NUMBER OF LOCATIONS ON NEXT LINE:
00060 DATA 6
00070 REM - ENTER STARTING NODE ON NEXT LINE:
00080 DATA 1
00090 REM - ENTER ENDING NODE ON NEXT LINE:
00100 DATA 6
00110 REM - ENTER A TRIAL BEST PATH, IN ORDER, ON NEXT LINE. IF
NOT RETURNING
00120 REM - TO STARTING NODE, GIVE P(N+1) 0:
00130 DATA 1,2,3,4,5,6,0
00140 REM - ENTER DISTANCE MATRIX, WITH LINE 1 AS DISTANCES FROM
1 TO 1,2,...,N
00150 REM - ....LINE 2 AS DISTANCES FROM 2 TO 1,2,...,N
00160 REM - ...AND SO ON, UP TO N LINES:
00170 DATA 0,6840,11280,13080,13440,21250
00180 DATA 6840,0,5100,6900,7260,17900
00190 DATA 11280,5100,0,1800,2160,19800
00200 DATA 12900,6480,1200,0,300,19800
00210 DATA 11700,6180,900,1200,0,19550
00220 DATA 21250,17900,19800,20550,20900,0
00230 REM
00240 REM - START OF PROGRAM ALGORITHM
00250 REM
00260 DIM P(20),D(20,20),ID(20),Q(20)
00270 S=-16336
00280 PRINT
00290 PRINT
00300 READ R
00310 READ N
00320 PRINT 'NUMBER OF NODES IS EQUAL TO',N
00330 READ SN
00340 PRINT 'STARTING NODE IS',SN
00350 READ EN
00360 PRINT 'ENDING NODE IS',EN
00370 FOR I=1 TO N+1
00380 READ P(I)
00390 NEXT I
00400 FOR I=1 TO N
00410 FOR J=1 TO N
00420 READ D(I,J)
00430 PRINT D(I,J)
00440 NEXT J
00450 NEXT I
00460 M=N+1
00470 LARGE=0
00480 FOR I=1 TO M
00490 FOR J=1 TO M
00500 IF (D(I,J)>LARGE) THEN LARGE=D(I,J)
```

April 1983 51

b)

"PERIODS" OF DE BRUIJN SEQUENCES 155

ience of periodic-
≤ n − 2 and are

length p, 2 ≤ p

$- \frac{n-1}{2^{n-2}}\Big)$

ths anywhere on
ng de Bruijn se-

iods" is the *same*
n + 2.
bove, fairly tight
ed for the set of

1. (1)

$\frac{-z^{m+1})}{}.$

1. (2)

$$f''(z) = \sum_{n=0}^{m} n(n-1)z^{n-2} = \frac{-(m+1)z^m}{(1-z)^2} + \frac{2(1-z^{m+1})}{(1-z)^3}$$

$$- \frac{(m+1)mz^{m-1}}{(1-z)} - \frac{(m+1)z^m}{(1-z)^2}$$

$$= \frac{2(1-z^{m+1})}{(1-z)^3} - \frac{2(m+1)z^m}{(1-z)^2} - \frac{m(m+1)z^{m-1}}{(1-z)},$$

$$f''(z) = \frac{2}{(1-z)^3} - \left\{\frac{2z^{m+1}}{(1-z)^3} + \frac{2(m+1)z^m}{(1-z)^2} - \frac{m(m+1)z^{m-1}}{(1-z)}\right\},$$

for |z| < 1. (3)

$$f'''(z) = \sum_{n=0}^{m} n(n-1)(n-2)z^{n-3}$$

$$= \frac{6}{(1-z)^4} - \left\{\frac{2(m+1)z^m}{(1-z)^3} + \frac{6z^{m+1}}{(1-z)^4} + \frac{2m(m+1)z^{m-1}}{(1-z)^2}\right.$$

$$+ \frac{4(m+1)z^m}{(1-z)^3} + \frac{m(m+1)(m-1)z^{m-2}}{(1-z)}$$

$$\left. + \frac{m(m+1)z^{m-1}}{(1-z)^2}\right\},$$

$$f'''(z) = \frac{6}{(1-z)^4} - \left(\frac{6z^{m+1}}{(1-z)^4} + \frac{6(m+1)z^m}{(1-z)^3} + \frac{3m(m+1)z^{m-1}}{(1-z)^2}\right.$$

$$\left. + \frac{(m^3-m)z^{m-2}}{(1-z)}\right). \quad (4)$$

c)

the problem

Figure 7: Example showing steps involved with the extraction of pixels associated with ascenders and descenders. An image fragment is shown in the top frame.

13

d)

Figure 4.6: a) Document image with the maximum error in skew angle detection due to a contained document image reproduction; b-d) Orientation detection failures due to: all-uppercase listings, math formulas or graphic objects arranged such that the character spacing is higher than the line spacing

The first test set consisted of the 979 test images from the UW-I dataset [215], containing technical journal scans featuring one, two or three layout columns. Our second test set was the OCRopus test set [277], consisting of 9 different images, each scanned in four different resolutions, namely 150, 200, 300 and 400 dpi. Each image was also rotated in all four orientations, thus resulting in a total number of 144 test images. The third and fourth test sets consisted of 100 single-column, respectively 109 two-column journal images, obtained by converting the PDF version of several recent articles into 300 dpi raster images. Each of the images in these test sets was rotated 360 times using bicubic interpolation with an interval of 1°, starting from −180°, thus totaling 36 000, respectively 39 240 images. The last test set consisted of a uniform sample of 100 images from the UW-I test set, selected manually from their "skew-free" version, available as part of the UW-III database. Again, each of the images was considered in 360 different rotated variants, leading to a test set size of 36 000 images. Since the "skew-free" variants of the UW-I images were obtained by the UW creators using an automatic skew detection and correction algorithm, their real skew had to be checked manually and those images were selected where the difference to the real skew was at most about 0.1°. Note that since the scans feature wavy lines, distortions near the book fold and significant skew differences among the different layout columns of the same document, the slant of the text lines even within the same page frame varies considerably – up to about 0.42° in our selection.

In order to evaluate the skew detection accuracy we have used the last three datasets, as they are the only ones large enough to obtain meaningful results. For computing the skew detection error for each image we have considered that the orientation was detected correctly, i.e. the skew angle for each orientation falls between $[-45°, 45°)$. This has allowed us a direct accuracy comparison with the proposed basic algorithm on the complete data sets.

UW-I	**Total images**	**Bloomberg et al. [58] # correct**	**van Beusekom et al. [277] # correct**	**Proposed # correct**
top-up	970	935	963	966
top-left	9	2	7	7
top-down	0	0	0	0
top-right	0	0	0	0
Total		95.8%	99.1%	99.4%

Table 4.1: Orientation detection results on the UW-I dataset

On the UW-I test set our algorithm performed very well and failed only on images containing almost exclusively majuscules, digits and/or mathematical formulae. As also noted in section 4.2.4, for all algorithms relying on the ascender-to-descender ratio such documents are inherently impossible to classify correctly. In such case, a combination with methods using other orientation-dependent features (e.g. [40]) is necessary.

UW-I	**Total images**	**Bloomberg et al. [58] # correct**	**van Beusekom et al. [277] # correct**	**Proposed # correct**
150 dpi	36	36	36	36
200 dpi	36	36	36	36
300 dpi	36	36	36	36
400 dpi	36	27	36	36
Total		93.8%	100%	100%

Table 4.2: Orientation detection results on the OCRopus dataset

For the OCRopus test set, the proposed orientation detection achieved a 100% success rate, the same as the algorithm proposed by van Beusekom et al. [277] and better than the Bloomberg et al. method [58] with 93.8%. The perfect result was possible because the test set consists exclusively of text-only, artificial images. Note that the main purpose of this data set was to test the resolution independence property without the influence of any other factors.

From table 4.3 one may see as expected that the accuracy of the improved algorithm was indeed better than that of the basic algorithm. The rising average erorrs directly reflect the increasing difficulty of the data sets. In case of the UW-I subset the sharp rise of the

	Total images	**SSkewDet Basic**			**SSkewDet Improved**			
		avg. err.	**std. dev.**	**max. err.**	**avg. err.**	**std. dev.**	**max. err.**	**orient. errors**
1 col. PDF	36 000	0.21	0.28	1.95	0.08	0.13	1.2	84
2 col. PDF	39 240	0.22	0.28	1.65	0.09	0.14	1.0	0
UW-I selection	36 000	0.26	0.32	1.55	0.15	0.2	1.0	0

Table 4.3: Skew and orientation detection results on datasets 3–5

average errors and their corresponding standard deviations also points at the inaccuracy of the "ground truth". Overall, the accuracy of both algorithms was very good, being similar to that of algorithms having constraints on the logical layout [46]. In fact, the only observed errors produced by the orientation detection algorithm were for documents where its preconditions were violated (as seen in figure 4.6).

The processing times on a 300 dpi A4 document image (approx. 2500 × 3500 pixels) are 0.01 seconds for the basic algorithm and 0.11 seconds for the improved version. For a 200 dpi image the times are 0.01 seconds and 0.07 seconds, respectively. The computer used for the tests had a Core2Duo 2.66Ghz processor. In comparison, the fastest state-of-the-art algorithm reported around 0.01 sec processing time on 200 dpi images [46] on a relatively similar computer configuration.

4.3 Conclusions

This chapter has introduced the problem of document skew estimation and presented a comprehensive review of the state of the art in the area. Global skew detection was selected as the current work's focus. This follows the observation that in the case of document collections of (semi-)professional scans (as found in mass digitization projects) it represents by far the most common type of skew.

Our contribution to the research area consists of two new algorithms for global skew detection falling in the category on nearest-neighbor approaches and employing the same framework. An important distinguishing feature of the proposed framework is the existence of a theoretical foundation, in contrast to the ad-hoc nature of most other state-of-the-art global skew detection algorithms. Another crucial feature contributing to the robustness and speed of both algorithms is the fact that, unlike other recent approaches [46, 277], we do not require any kind of prior layout analysis, such as text line determination. This fact is highly significant because it completely discards all hidden or layout-dependent parameters otherwise necessary. Consequently, we do not require any explicit parameters within the algorithms. It is interesting to note that despite its simplicity, the underlying Euclidean MST essentially approximates the text lines via piecewise linear functions, as opposed to the more error-prone linear model used in all other state-of-the-art techniques. Finally, we introduced a way to assign meaningful confidence estimates to the global skew angles computed by both algorithms. Such confidence measures represent an invaluable tool for automatically assessing the overall DIU system output in view of further manual correction or for presentation purposes. Extensive testing on a total set of about 110 000 images was used to validate the proposed techniques and compare them to other state-of-the-art methods with regard to both skew- and orientation detection accuracy.

A further evaluation on a grayscale/color document image dataset with respect to different binarization/color reduction algorithms would be of great interest, as this would be more in line with the current demands for real-life document digitization projects. Unfortunately to the author's best knowledge no large datasets suitable for this task currently exist.

We believe that holistic skew estimation by integrating multiple sources of information represents a highly promising research direction. More specifically, our current model disregards other valuable sources of information, such as the direction and curvature of solid separators (see sections 5.1.1 and 5.2.1), edges of halftones, as well as all shading information. These elements, together with the available text line and word directions can be integrated into a fine-grained whole-page skew map, thus allowing an accurate non-uniform skew correction. Beside the improvement of the overall robustness gained by the use of a larger set of (mostly) uncorrelated features, such an approach would allow for a uniform treatment of all skew types.

Chapter 5

Page Segmentation

It sounded logical – but I could not forget Kettering's Law: "Logic is an organized way of going wrong with confidence."

– R.A. Heinlein *(The Number of the Beast)*

Page segmentation is also known in the specialized literature as *physical* or *geometric layout analysis*. Its purpose is to segment a document image into homogeneous zones and categorize each zone into a certain class of *physical layout elements*. Most commonly, the physical layout elements are divided into text, graphics, halftone pictures, horizontal and vertical rulers. As already mentioned in the introductory section of the current work, some publications consider more and others fewer classes of physical layout structures, depending on the target area. For example, identifying mathematical formulae and chemical structure diagrams is important when having to deal with textbooks or journal papers in the area of natural sciences, but largely irrelevant when processing newspapers or fiction novels.

Ideally, the document image analysis process should be based solely on the geometric characteristics of the document image, without requiring any apriori information (such as a specific document format). Actual implementations and theoretical methods rely upon different apriori information about the document layout. Most frequently, an important distinction is made between the so-called *Manhattan* and *non-Manhattan* (arbitrary) layouts. One must note that, contrary to the belief of many authors, a consensus on the definition of Manhattan layouts does not currently exist in the document analysis community [66]. This is probably due to different characteristics considered to define the Manhattan layout. One view is that the main characteristic is given by the background regions (corresponding to the Manhattan streets) which always clearly delimit the foreground areas at 90° angles and are never completely contained within foreground regions [202, 257]. The other view is that the foreground regions directly correspond to the city blocks in Manhattan, thus are always rectangular [126, 239]. The two perspectives are not equivalent, since the first one allows for foreground regions to be *naturally* represented as simple isothetic polygons (polygons without holes having all edges parallel to the coordinate axes). Since document processing is performed on digitized images, one may argue that practically any region can be represented as an isothetic polygon (at its most extreme having 1–pixel long edges). This ambiguity is for the most part clarified by the requirement (present in both cases) of having a *clear* separation between regions. In other words the spacing of elements (or the texture) in both regions must be strictly lower (or different) than that of the background region(s) located in–between. In this paper we adhere to the train of thought that Manhattan layouts allow the existence of simple isothetic polygonal foreground regions. Nowadays, most printed newspapers and scientific journals feature a Manhattan layout.

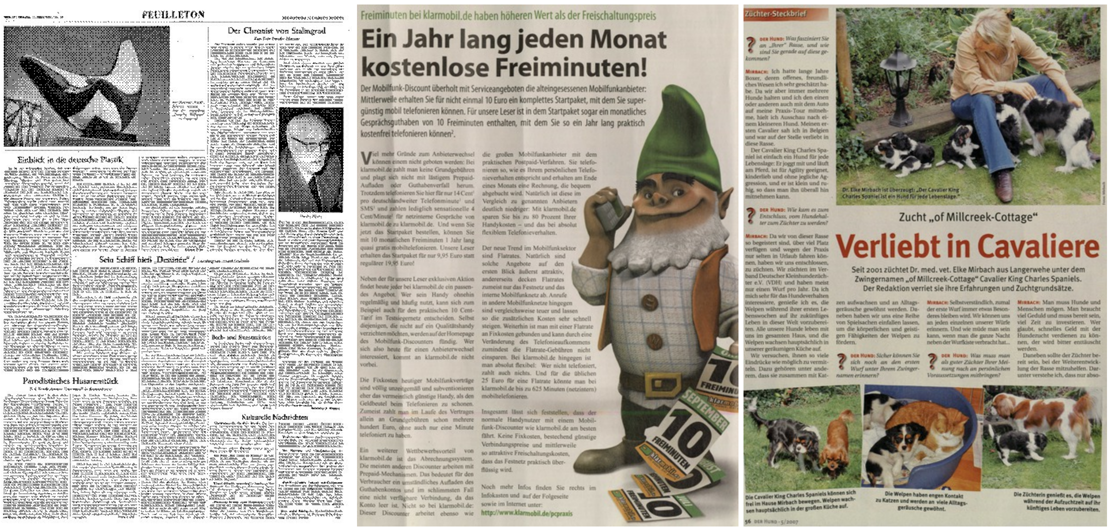


Figure 5.1: Document scans exhibiting different layouts: rectangular-, Manhattan- and Non-Manhattan (arbitrary) layout (left to right)

Another type of assumption is related to the characteristics of the input image. In most cases the input image must be binary, as well as noise- and skew-free, although a few algorithms have been proposed which are able to deal with grayscale/color images, skewed and/or noisy documents. In the case of mass digitization, pre-processing steps are practically unavoidable due to the enormous variety within the input material. Consequently, the common limitation to a binary-only skew-free input image is in practice of no consequence.

A much more important issue arises on the exact definition of a physical page region, especially in the case of text regions. Many authors (e.g. [38]) consider that the correct separation of text regions exhibiting different characteristics (such as font size, stroke width, etc.) represents the task of geometric layout analysis. In our opinion, this is a dangerous requirement, as it is highly unclear what “different characteristics” actually mean in the case of text regions. A wide variety of (ever changing) layout styles are in use today and in some situations the features necessary for a correct segmentation of the text regions are simply not available at this early a point in the processing chain. Take for example a sequence of words typewritten in italics, but located within an otherwise similar but non-italic text body. Without any further assumptions, it is unclear if the sequence of words should indeed be considered to be an independent geometric region or not. At the very least, a simple threshold based on the length of the region (possibly in relation to the containing layout column) is probably necessary in this case to make a correct decision. After all, if the sequence is very short it is likely that it represents a feature detection error. Another common unclear situation is at the end of a paragraph and the start of a new, indented paragraph with the exact same font characteristics, featuring no additional vertical spacing at the start. Human-generated page segmentations (used as ground truth for evaluation [38, 239]) normally show one region corresponding to each of the paragraphs. However, since the paragraphs have identical font characteristics, the decision as to whether and exactly where to separate the paragraphs from each other falls over to their indentation characteristics. These in turn depend heavily on the publisher’s current layout style. Clearly such decisions are not universally valid and furthermore they fall into the realm of logical layout analysis, as they are exactly of the same type as those used to label logical regions. In effect, having this kind of unclear rules for text region separation is detrimental for the evaluation process. As such, in the current work we shall consider that the exact separation of text regions is not in any way the task of the page segmentation module. Instead we consider it as an early part in the logical layout analysis step. The evaluation results in table 5.2 directly reflect our position in this matter.

The current chapter is divided into three main parts followed by a summary and directions for future research. In the first part, we go over the research developments in the areas of document separator detection, page segmentation and segmentation evaluation methodologies. The second section introduces a series of improvements over the current state of the art, as currently implemented in the Fraunhofer DIU system. As mentioned in section 1.1, the Fraunhofer DIU system has already been extensively used in mass processing of heterogeneous newspaper and book archives totalling well over 1 million pages. The third section describes the evaluation methodologies for page segmentation used in the current work and discusses the experimental results obtained on a few data sets containing scanned images of different types: newspapers, journals, magazines and historical documents.

5.1 State of the Art

Page segmentation methods are traditionally categorized into three groups: *top-down* (model-driven), *bottom-up* (data-driven) and *hybrid* approaches.

In top-down techniques, documents are recursively divided from entire images to smaller regions. These techniques are generally very fast, but they are only useful when apriori knowledge about the document layout is available. To this class belong methods using projection profiles [123], X-Y cuts [124], or white streams [23].

Bottom-up methods start from pixels, merging them successively into higher-level regions, such as connected components, text lines, and text blocks. These methods are generally more flexible and tolerant to page skew (even multiple skew), but are also slower than top-down methods. Some popular bottom-up techniques make use of region growing ([142, 150, 198]), run-length smearing [280], or mathematical morphology [26, 112].

Many other methods exist which do not fit exactly into either of these categories; they were consequently called hybrid methods. Hybrid approaches [96, 211] try to combine the high speed of the top-down approaches with the robustness of the bottom-up approaches. Within this category fall all texture-based approaches, such as those employing Gabor filters, multi-scale wavelet analysis [91], or fractal signatures [262].

Although useful, the aforementioned classification methodology leads to many ambiguities, evidenced by the fact that often the same algorithm is assigned to different classes by different authors [66]. Several other classification schemes for page segmentation algorithms were proposed [66, 201, 202]. In the current work, we adopt a straightforward classification of the page segmentation methods, based on the functionality offered by each algorithm. As such, we divide page segmentation algorithms into two classes: region detection algorithms which segment the document into homogeneous regions (but in many instances do not actually classify them) and region classification algorithms, which solely classify a set of pre-determined regions.

In order to segment a document into independent regions, for many algorithms it is advantageous to make use of the set of all separators present on the respective document page. Some methods [59] require a full list of separators explicitly, whereas for most others [124, 150, 198, 257] such a list may easily be incorporated (e.g. as constraints) and used to increase the accuracy of the segmentation process. There also exist monolithic segmentation algorithms [142] which include separator detection methods. In such cases, the integrated separator detection methods tend to have an ad-hoc character and can be substituted with more generic methods with relatively little effort. Before delving into state-of-the-art page segmentation algorithms, we go over several popular methods for generic separator detection.

It is worth noting that both commercial OCR products [1, 14] and open-source OCR engines [13, 15] (see section 1.2.2) are constantly improving their geometric layout analysis modules. While their traditional focus on born-digital documents remains obvious, their transition towards agile distribution models (e.g. mobile apps, web services) in the hope of a larger user basis is increasingly forcing them to adopt more robust segmentation methods. Currently, the vast majority of research algorithms are suitable only for rectangular or Manhattan layouts [66, 178, 202, 239]. Page segmentation methods capable of reliably processing non-Manhattan and multiple skew layouts with irregular column structure remain a rarity.

5.1.1 Separator Detection

Separators present in typical documents can be classified into two main classes: 1. long horizontal/vertical line segments, henceforth dubbed *solid separators*, and 2. large, elongated empty areas (dubbed *whitespaces*). This is due to nearly universal conventions of legibility, such as those described in the Chicago Manual of Style [197]. Depending on the publisher and the publication type, a document may contain either or both types of separators. Thus, without apriori knowledge about the document layout, one must be able to identify both kinds of separators.

Arguably, the simplest methods for detecting *solid separators* make use of connected components. In general, such methods assume that each separating line consists of a single connected component (or several closely-aligned components). After labeling the connected components, vertical and horizontal lines are detected by using relatively simple rules based on the component size, shape or run-length characteristics. In the early ICDAR Page Segmentation Competitions [34, 36, 113], several connected component-based detection methods [125, 187] have been evaluated. Surprisingly, it was found that they are able to perform moderately well (about 50% detection ratio), even when applied on document images having different layout types, scanning resolutions and noise levels. One must however keep in mind that the competitions are exclusively focused on documents showing little to no distortions, digitized under constant lighting conditions, in other words the kind of digitized documents one expects from a professional scan service provider.

Another popular method of detecting solid separators is the use of the Hough transform. The straightforward application of the Hough transform returns only the determining parameters of each infinite line; furthermore, due to the voting algorithm, short segments are normally not detected. These issues, along with the low execution speed, have been the main focus of the algorithms for separating line detection proposed in the scientific literature [43, 189].

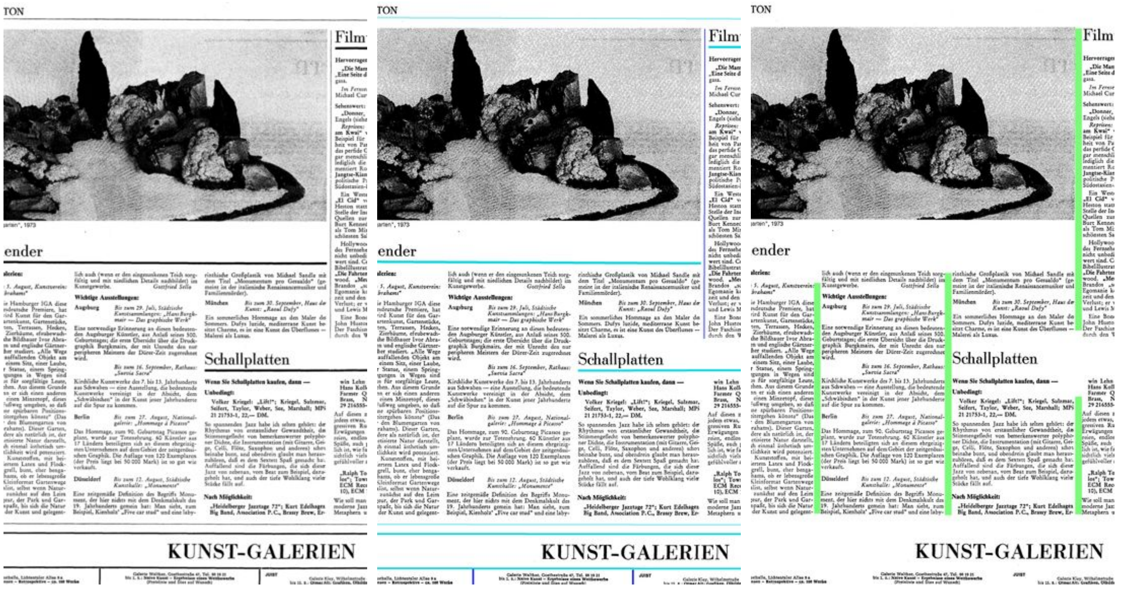


Figure 5.2: Examples of separator types found in documents: left – original portion of newspaper scan; center – solid separators (cyan= horizontal; blue= vertical); right – vertical whitespaces (green)

Algorithms for separating line detection based on mathematical morphology [85, 176] have also been evaluated in the ICDAR page segmentation competition [34, 36]. Their detection performance was found to be close to that of connected component-based methods. Morphology-based approaches, however, have the advantage that they can be directly applied to a grayscale image. In general, such algorithms work by initially improving the quality of the separators by means of successive opening and closing operations. The second and final step is the extraction of vertical/horizontal separators, in a manner similar to the connected component-based approach previously described. Considerably better detection results were reported using the morphological approach of Gatos et al. [112]. This method makes use of the average character width and height in order to better estimate the mask employed by the morphological operators.

Based on directional single-connected chains (DSCC), a fast algorithm capable of detecting diagonal lines with arbitrary slopes was proposed by Zhang et al. [302]. A DSCC is defined as an array of adjacent black pixel run-lengths. Separating lines are detected by merging DSCCs under certain constraints. This approach was found to perform well, even in case of noisy images and severely broken lines [112, 301]. A more detailed description of this algorithm may be found in section 5.2, as part of the Fraunhofer page segmentation method. Other proposed methods for solid separator detection use 2-D wavelets [291] or general-purpose vectorization algorithms [169].

The problem of finding *whitespaces* within a document image is closely related to the determination of a full covering of the document background using disjoint rectangular areas. In principle, given the page area and a set of obstacles (in the form of points or bounding boxes), covering algorithms try to find at each step the maximal empty rectangular area. Afterward, the rectangle is added to the set of obstacles, and the procedure is repeated until the whole page background has been covered. Transforming a covering algorithm into one for finding the whitespaces can be accomplished easily: one must only add a stopping criterion (such as a minimum area for the rectangle) as well as allow a certain overlap for the determined rectangles [59].

There exist several algorithms for solving maximal empty rectangle problems, such as those from computational geometry [203] and document analysis [53]. These early algorithms, although computationally efficient, have not found widespread use, mainly due to their relatively difficult implementation [59]. More recently, two new globally optimal algorithms were proposed by Breuel [59, 60], the latter also being able to detect rectangles at arbitrary orientations. Although both methods employ a branch-and-bound search strategy, they were found to be fast enough on real-world DIN A4-sized document scans [239]. Our tests on larger newspaper scans from section 5.3 confirm this conclusion.

5.1.2 Region Detection

One of the most well-known and widespread methods for segmenting document images with *rectangular layouts* is based on *recursive X-Y cuts* (RXYC). The original method was introduced in 1984 [190], and since then many authors have improved upon the original algorithm [123, 124]. In its initial variant, the X-Y cut algorithm is a classical top-down algorithm, based on a hierarchical (tree-like) decomposition of the document page. The root of the tree represents the entire document image, while the set of all leaf nodes in the tree represents the final page segmentation. At each step of the algorithm, the horizontal and vertical projection profiles of each node are computed by considering all their contained black pixels. In the two obtained profiles, the width of each valley is compared to some predefined threshold values (one for the X- and another for the Y-axis). If a valley is wider than the respective threshold, the node is split vertically or horizontally at the mid-point of the valley into two children nodes. The process continues until no nodes can be split further. As a final step, noise regions are removed using a few heuristic criteria. Arguably the most important improvement in the RXYC algorithm, as proposed by Ha et al. [123], is the usage of the bounding boxes of connected components for computing the projection profiles. This results in a significant speedup, and practically transforms the original top-down algorithm into a hybrid algorithm (connected component labeling is a bottom-up process).

A popular method which can handle documents with *Manhattan layouts* is the *run-length smearing algorithm (RLSA)* [286]. The basic 1-D RLSA, when applied to a sequence of black and white pixels, has the effect of linking together black intervals that are separated by white intervals having a width less than C pixels (where C is a fixed value). The 1-D RLSA is applied row-by-row and column-by-column (with different values of C), resulting in two distinct images, which are subsequently combined pixel by pixel using the logical AND operator. It is worth mentioning that a more efficient method of performing the 2-D smearing was introduced by Wahl and Wong [281]. Finally, each obtained block of black pixels is categorized into text, graphic/halftone images and horizontal/vertical separators by using a few pre-defined rules. These rules use features such as the number of horizontal white-black transitions, the ratio of the number of block pixels to the area of the surrounding rectangle, the mean horizontal length of the black runs and the total number of black pixels in a block. More recently, the RLSA algorithm was modified so as to be able to work on arbitrary document layouts [257]. The improved algorithm, called selective constraint run-length algorithm (CRLA), uses several properties of the connected components in the original image in order to assign one of the 4 possible labels to each pixel. Thereafter, the selective CRLA procedure is applied twice on the labeled image, yielding as output the text regions in the page. Although straightforward to implement, both algorithms are heavily dependent on the choice of the horizontal and vertical C values, and are quite sensitive to noise [239].

An algorithm usable for documents with *Manhattan layouts*, which is capable of dealing with moderate amounts of skew and noise, was proposed by Jain and Yu [142]. Based on a novel representation of the document image, the block-adjacency graph (BAG), the authors describe a bottom-up algorithm capable of both segmenting and classifying the physical regions of a document image. The algorithm was shown to perform well on a wide variety of non-noisy journal and newspaper images. Problems were identified in documents where large variations in font size and text spacing were present, as well as in the discrimination between drawings and halftone images. A more detailed description of the Jain and Yu method can be found alongside the Fraunhofer page segmentation method in section 5.2.

Relatively few algorithms can deal with documents having *arbitrary layouts*. One such method, which has performed well in recent comparisons [177, 239] was introduced by Kise et al. [150]. It is a bottom-up algorithm, based on constructing the approximated area Voronoi diagram constructed from points sampled from the boundaries of the connected components. Afterward, many superfluous Voronoi edges are removed by using two characteristic features: a minimum inter-character distance and the area ratio of connected components. The output of the algorithm is a list of polygonal regions, representing the physical regions of the given document page. The obtained regions are not classified, thus, most commonly, a further necessary step is the transformation of the arbitrarily-shaped polygons into isothetic polygons or rectangles, in view of subsequent classification [177].

The well-known open-source OCR engine Tesseract [15] incorporated in 2009 its own geometric layout analysis module [250]. The new segmentation module is capable of handling Manhattan layouts and irregular, complex column structures. Smith proposes a bottom-up method based on connected component grouping to detect the tab stops (i.e. white rectangles) in a document image and uses the tab stops to deduce its column layout. Subsequently, the document is segmented into homogeneous polygonal regions under the restraint posed by the column layout. An innovative method for page region segmentation was recently introduced by Gao and Wang [110] of the Palo Alto Research Center (PARC). Their algorithm first creates an over-segmented representation of the document into words and compute a word graph via generic visual features. A set of zone hypotheses are generated from the word graph. Document image decomposition is then formulated as a maximum a posteriori zone inference problem and the zone partition having the maximum probability given the extracted word graph is computed using the A* search algorithm.

Many other algorithms for region detection have been proposed in the literature. For a more complete overview one may consult the most recent surveys, such as [66, 178, 201, 239]. The ICDAR page segmentation competitions [33, 38] offer a good reference point regarding the maturity of important state-of-the-art systems and methods.

5.1.3 Region Classification

With respect to the classification strategies employed, region classification algorithms are typically divided into two major categories [282]: *rule-based* (grammar-driven) and *statistical-based* (parametric or non-parametric). Rule-based algorithms use a set of heuristic rules or pre-defined syntax rules of grammars to derive their decisions. The heuristic rules or the syntax of the grammar are fixed and empirically determined, most commonly tuned to specific layouts or other document properties. In the statistical-based algorithms, the decision parameters are usually obtained by an off-line training process. In this section, we briefly present several notable document zone classification algorithms, which can work independently of the region detection process. At this point it is worth noting that there also exist research papers presenting algorithms for document whole-page classification [67]. The goal in such case is to classify a given document page as a whole into one of several pre-defined categories, such as title-, index-, regular- or advertisement page. In order to achieve a reliable whole-page classification, higher-level features are commonly employed, including OCR results and the positions and labels for the physical regions. Consequently, these methods are a basic form of logical layout analysis which makes the subject of chapter 6.

A *rule-based region classification* algorithm specifically adapted for technical journal pages was described by Lee et al. [161]. In a manner very similar to Niyogi and Srihari [194], they construct a knowledge base, which encodes both publication-specific characteristics and common characteristics of technical journals in the form of rules. Based on their functionality, some rules are dedicated to page segmentation, while the rest are used in region type identification. The system comprising of over 100 rules is capable of classifying regions into text lines, equations, images, drawings, tables and rulers. Experiments performed on 372 scanned images from the journal "IEEE Transactions on Pattern Analysis and Machine Intelligence" showed that the method produced a correct geometric layout analysis in more than 99% of the test images.

In their newspaper image segmentation algorithm [187] evaluated during the First International Newspaper Segmentation Contest [113], Mitchell et al. used a bottom-up approach to segment a newspaper image into regions. Subsequently, each region was classified into text, title, inverse text, photo, graphic/drawing, vertical line, and horizontal line by using several pre-defined rules. The utilized rules were based on features such as size, shape, black pixel numbers and black run-length characteristics. This algorithm did not achieve high recognition rates (above 50%) for any of the region types [113], perhaps due to the insufficient number of rules utilized. However, it is worth noting that another algorithm, using a similar rule-based approach (proprietary to Océ Technologies B.V.) had the highest overall classification rate in the ICDAR2003 Page Segmentation Competition [36].

In contrast to rule-based algorithms, *statistical-based methods* typically require a significant amount of training data, but have the major advantage that all parameters relevant for the classification are automatically learned. Wang et al. [283] describe such an approach, which represents each rectangular region by using a 25-dimensional feature vector. An optimized decision tree classifier is then constructed from the ground-truthed University of Washington III database [215] and used to classify each zone into one of nine possible zone content classes. The authors propose a performance evaluation protocol, according to which the achieved accuracy is 98.45%, with a mean false alarm rate of 0.50%.

Jain et al. [141, 143] use a neural network to learn a small set of masks which best discriminate between text, background, drawings, and pictures. The convolution of the masks with the input image produces texture features that are further used in a neural network [143] to classify each pixel into one of three classes (text+drawings, images, background). The regions from the first category are subsequently binarized and separated into text and drawings by a component size threshold. The method was shown to be robust to different image layouts and it is also able to discriminate between text paragraphs written in different languages (such as Chinese vs English). A more detailed and complete overview of texture-based classification algorithms is given by Tuceryan and Jain [274].

A more recent development is the Document Image Content Extraction (DICE) system, proposed by the document research group at the Lehigh University, USA. Their system is capable of performing classification on a pixel-level and only then enforces local uniformity constraints [27]. In this way, the authors avoid an arbitrary lock down on an artificially-restricted set of possible region shapes. Note that the DICE system was one of the systems participating in the ICDAR 2009 Page Segmentation Contest and proved to be capable of reasonably segmenting complex layouts and color documents even when restricted to a rather small training set.

Keysers et al. [148] review the specialized literature and select a promising set of fully generic features, directly expanding on the work of Wang et al. [283]. They test each feature and several combinations of features with respect to their classification performance on a subset of the University of Washington UW-III document database containing around 13 800 zones. Considering that all features investigated can be computed from binary images (no color information is used), the achieved error rate of 1.5% for 8 zone types is highly competitive. More interestingly, the authors identify a few subsets of features which allow a very fast computation and still achieve a low error rate of 1.8 – 2.1%.

Finally, Baird and Casey [52] discuss in detail the issues arising from the attempt to solve the document content classification problem in its full generality. They mention the lack of agreement within the research community as to what constitutes a representative set of samples and discuss how such a set could be constructed by the principled enrichment of sample sets using synthetic variations. They conclude with an overview of possible classification methodologies capable of learning efficiently from such huge data sets. From the number of remaining open issues, it becomes clear that the generic problem of document zone classification is still far from solved.

5.1.4 Evaluation Methodologies

As seen in the previous sections, many algorithms for performing geometrical layout analysis have been proposed during the last two decades. Despite the abundance of methods for performing document image analysis, currently it is still a difficult task to compare the accuracies of different methods. Most often, new algorithms are tested on different datasets, under different initial conditions and having widely varying objectives [201]. In the last 5 years, significant progress has been made with respect to evaluation techniques, however sufficiently large and varied datasets open to the public still represent a thorny issue.

The problem of automatic evaluation of page segmentation algorithms has in the last 10 years become an important issue as the number of approaches proposed in the literature has grown. Major problems arise due to the lack of a common dataset, a wide diversity of objectives, and the lack of a standardized, meaningful quantitative evaluation [239]. Attempts to provide an adequate dataset and a general evaluation methodology have been made in the past (e.g. [215, 296]), without much success ([201, 239]). A promising effort in this direction is currently being made by the PRImA Research Lab, from the University of Salford, UK (see [31, 37]). In general, evaluation methodologies can be divided into two categories: *text-based* and *region-based.*

The *text-based* approach was the first proposed in the literature [145, 146]. Following such a methodology, the quality of page segmentation is evaluated by analyzing the errors in the text produced by the OCR module. First, page decomposition and OCR are applied to the document page, and the result is output as a character string. The quality score is then computed based on an approximation of the cost of human editing to convert the obtained string into the ground truth string. For example, in [145] and [146], the quality score is obtained as the sum of the costs associated with each edit operation from the best possible edit sequence which transforms an OCR output to the correct text. The authors consider three possible edit operations: insert, delete and move, each having its own cost. Determination of the best possible edit sequence is done via specialized string-matching algorithms [275]. Mao et al. [177] presented an empirical performance evaluation methodology and used it to compare three research algorithms [150, 190, 198] and two popular commercial products (now both incorporated into Nuance's Omnipage [14]). As the goal of commercial products is to provide a high OCR accuracy, their evaluation methodology was based on text-line detection accuracy. The advantages of text-based approaches is that ground truth files are very easy to create and they do not require the page segmentation module to specifically output any kind of zoning results. Although the underlying string-matching algorithms are quite elaborate, the overall approach is straightforward. Therefore, text-based zoning evaluation approaches have been well accepted by the document image analysis community [296]. Nevertheless, such a methodology suffers from the limitation that it can only deal with text regions, while at present, documents containing non-textual regions are by far preponderant.

To overcome these limitations, *region-based* approaches have been introduced in the late '90s [31, 164, 296]. Here, a document is regarded as having a hierarchical logical structure (e.g. physical layout structure and logical structure). Segmentation quality may thus be evaluated at different levels of the document representation. The performance metrics employed are based on finding matches between the homogeneous regions from the segmentation output of the considered system and the corresponding existing ground truth. According to [66], the major difficulty for such approaches is the definition of a distance measure between two sets of regions: it has to encompass and to balance several elements such as correspondence between regions, overlap degree between regions and presence of unmatched regions in the two sets.

Liang et al. [164] propose a quality measure which is based on the overlapping between rectangular regions coming, respectively, from the ground truth and the page segmentation algorithm. Given two sets of rectangular regions: $G = \{G_1, G_2, \dots, G_M\}$ corresponding to the ground truth regions, and $S = \{S_1, S_2, \dots, S_M\}$, corresponding to the obtained segmentation regions, two matrices are computed as follows:

$$\sigma_{ij} = \frac{Area(G_i \cap S_j)}{Area(G_i)},\ \tau_{ij} = \frac{\text{Area}(G_i \cap S_j)}{Area(S_j)},\ i = \overline{1, M}, j = \overline{1, N}$$

Correct matches and errors (i.e. missing, false detection, merging and splitting) are detected by analyzing the matrices (σ_{ij}) and (τ_{ij}). The overall quality score is defined as a weighted sum of these quantities.

Yanikoglu et al. [296] presented a region-based page segmentation benchmarking environment, named Pink Panther. In their approach, a region is represented as a polygon together with various attributes (such as child regions and partial order information). By analyzing matches between ground truth and segmented regions, errors like missing, false detection, merging, and splitting are detected and scored. Most importantly, the detailed shape of regions is not taken into account for matching purposes, so that small irrelevant differences between ground truth and segmented polygons are ignored. Instead, the authors perform a per-pixel region matching by using just the contained black pixels of each region. The overall page decomposition quality is computed as the normalized weighted sum of the individual matching scores. Antonacopoulos et al. [31] present a more elaborated version of this algorithm which, among other features, allows each region to specify its background and foreground colors.

A more recent performance metric was proposed by Shafait et al [240]. It is interesting to note that approach of Shafait et al. shares many similarities with the proposed evaluation method for newspaper scans, as it was also developed for complex, multi-column layouts. Their method is region-based and works on pixel-level, thus allowing arbitrary-shaped segmentation regions. For region matching Shafait et al. propose a relative tolerance ratio of 0.1, regardless of the region type. Additionally, the authors use an absolute threshold (i.e. a fixed number of foreground pixels) for ensuring that overlaps greater than this value are never ignored. The evaluation of the quality of a page segmentation is done by computing the following metrics: total over- and under-segmentations, over- and under-segmented components, missed components and false alarms.

Finally, the biennial ICDAR page segmentation competitions starting from 2001 have seen the introduction of several novel methodologies. The early competitions [34–36, 113], up until 2007, used a classic region-based approach computing the intersection between ground truth and segmentation results on a pixel-level. The ICDAR 2009 segmentation contest [38] saw the introduction of scenario-based evaluation [29], which allowed the segmentation scores to more accurately reflect the performance of algorithms under real-life application cases. This in turn enabled a more accurate and meaningful comparison of the strengths and weaknesses of a large range of algorithms. The latest ICDAR document segmentation competition in 2011 focused specifically on the challenging issue of historical materials. Evaluation scenarios were extended in order to be able to reflect the additional difficulties posed by degraded printed documents [79]. More details about the scenario-based evaluation methods may be found in section 5.3.2.1, as they were used to evaluate the Fraunhofer DIU system on two datasets.

5.2 Page Segmentation for Mass Digitization

As seen in the previous sections, the majority of the state-of-the-art page segmentation methods are only capable of dealing with rectangular or Manhattan layouts. Furthermore, they are almost exclusively targeted toward single- or double-column layouts and are tuned for certain pixel resolutions. These limitations make them unsuitable for use in mass digitization projects, which are naturally characterized by a wide variety of layouts and large variations in digitized material quality. The current section presents a possible way in which the aforementioned constraints can be eliminated, via the practical example of the Fraunhofer DIU system. In the following one may find detailed descriptions of the proposed algorithms, as well as exhaustive evaluation results.

5.2.1 Separator Detection

The most challenging issues a *solid separator detection* algorithm has to face are severely broken lines, wavy lines and correctly dealing with intersections between characters and line segments.

From figure 5.3, it is clear that simple approaches, such as those based on measurements of the bounding boxes of connected components, Hough transforms and morphological

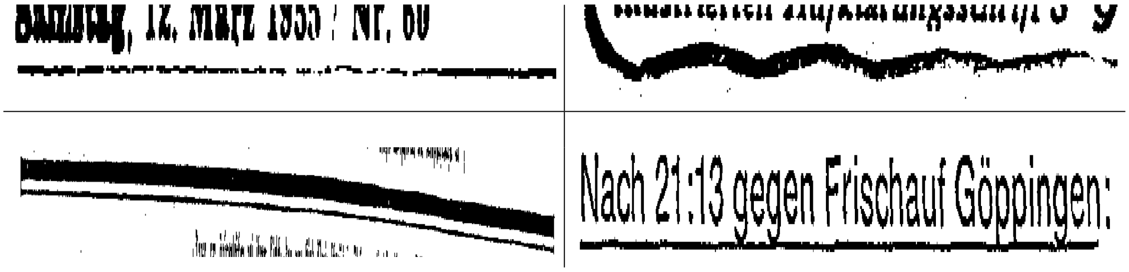


Figure 5.3: Common examples of situations having to be dealt with during solid separator detection: broken segment; wavy segment; curved segment; line-character intersections

transforms cannot handle all the presented situations correctly. A promising approach however, is the one described for the purpose of line extraction from forms by Zheng et al. [302], based on directional single-connected components (DSCC). For the purpose of representing solid separators with pixel-level accuracy, we also make use of the DSCC structure. A DSCC is defined as a collection of vertical/horizontal black run-lengths in which any two adjacent ones are vertically/horizontally overlapped. Additionally DSCCs end as soon as one run-length has more than than one neighboring run-length on either side (single-connectedness property). Figure 5.4 shows all vertical DSCCs (longer than a certain threshold) extracted from a single printed character.

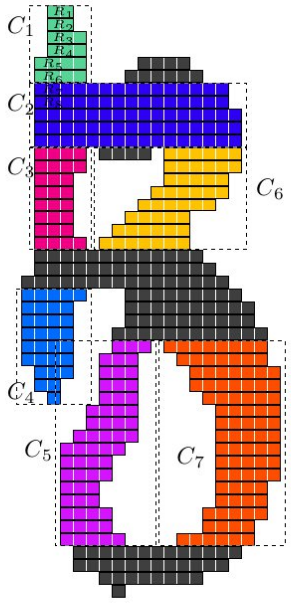


Figure 5.4: Gothic minuscule "z" and all its contained vertical DSCCs with a height greater than 4 pixels superimposed (C_1 – C_7)

A complete separator is then formed by merging all DSCCs having approximately the same direction and located close to one another. The main direction of each DSCC is approximated using a robust *mid-point line fit*. In the example from figure 5.4, good candidates for merging would be the DSCCs C_1, C_3 and C_4. The original algorithm has already been shown to have a good performance when working with severely broken separators [301], as it was designed specifically to be able to deal with forms filled in by hand and containing numerous intersections between handwriting and the form. Moreover, we observe that the definition of a DSCC is not inherently limited to straight line segments, but naturally allows the representation of (arbitrarily slanted) wavy and curved separators. A practical issue arising with the standard algorithm is the fact that the DSCC merging procedure is very slow for high-resolution images, even when using the search area reduction trick [302]. The reason for this is the computational complexity of $O(N^2)$ in the number of DSCCs together with the fact that a regular DIN A3 newspaper image contains about 50 000 – 100 000 DSCCs (most of which are very short). An additional issue with the standard algorithms is that, for very short separators or separators with a high deviation of the run-length widths, the results of using mid-point line fit are often imprecise and the merging procedure will consequently perform poorly.

In order to solve both issues, we employ a morphology-based improvement procedure similar to that proposed by Gatos et al [112]. In essence, a series of morphological operators are used on the document image to improve the appearance and contiguity of separators, with masks specifically tailored for nearly vertical or horizontal lines. The crucial part of the improvement procedure is that special care must be taken so as not to merge adjacent printed characters and thus generate many false potential line segments. We accomplish this by adapting the size of the morphology masks to that of the characters from the dominant font on the page. The methodology proposed for determining the width and height of the dominant has been described in detail in section 3.1.3.1. The morphologically-optimized pages will contain more clearly defined separators with uniform widths, dramatically reducing the number of candidate DSCCs and simultaneously increasing the accuracy of the mid-point line fit for each DSCC segment. Since a small number of false positives caused by the morphological optimization is practically unavoidable, we subsequently employ the image/text area removal procedure proposed in [112] as pruning step. A visual example of the execution of the combined procedure is shown in figure 5.7. In the following we describe the complete algorithm in detail.

As the first step, we use the novel algorithm described in section 3.1.3.1 for computing a robust estimate for the *dominant character width and height*. It is worth mentioning that we are able to obtain accurate size values even when dealing with complex, multi-font document images containing graphic elements. Next, based on the obtained dominant character width AW and height AH we construct several masks to be used by the morphological operators for improving nearly horizontal and vertical lines. As proposed by Gatos et al., two separate images I_H and I_V are used for detecting horizontal and vertical separators, respectively. Each of the images is obtained as the result of the application of a certain set of morphological operators on the original image, as one can see in figure 5.5. The goal of this step is to connect broken horizontal/vertical segments, but not to connect neighboring characters. Note that in the resulting images, there will still exist occasional cases when neighboring characters are wrongly connected, but such isolate situations have only a minor influence on the overall performance of the algorithm.

It is now possible to apply the DSCC-based line detection procedure for extracting horizontal, respectively vertical separators on the obtained improved images I_H and I_V. Zheng et al. define two types of DSCCs, corresponding to vertical, respectively horizontal separators. Only the case of horizontal separators will be treated here, as the situation for vertical separators can easily be inferred by analogy. A vertical black run-length R_i is a one-pixel wide sequence of vertically neighboring black pixels, defined as follows:

$$\begin{aligned}R_i(x_i, ys_i, ye_i) = \{(x, y) | \forall I(x, y) = 1,\ x = x_i,\ y = \overline{ys_i, ye_i}, \\ I(x_i, ys_i - 1) = I(x_i, ye_i + 1) = 0\},\end{aligned}$$

where $I(x, y)$ is the value of the image pixel (x, y), with 0 representing background pixels and 1 representing foreground pixels; x_i, ys_i and ye_i are respectively the x-, starting-,

$$IM_H = IM \cup (((IM \ominus B_{HR}) \cup (IM \ominus B_{HL})) \oplus B_H),$$

$$\text{where } B_{HR} = [\overset{\xleftrightarrow{AW}}{111\ldots\boxed{1}}],\ B_{HL} = [\overset{\xleftrightarrow{AW}}{\boxed{1}\ldots111}],\ B_H = \overset{\xleftrightarrow{0.5AW+1}}{\begin{bmatrix} 1 & . & . & . & 1 \\ . & . & \boxed{1} & . & . \\ 1 & . & . & . & 1 \end{bmatrix}} \updownarrow 0.2AH+1$$

$$IM_V = IM \cup (((IM \ominus B_{VD}) \cup (IM \ominus B_{VU})) \oplus B_V),$$

$$\text{where } B_{VD} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ . \\ . \\ . \\ \boxed{1} \end{bmatrix} \updownarrow 1.2\,AH,\quad B_{VU} = \begin{bmatrix} \boxed{1} \\ . \\ . \\ . \\ 1 \\ 1 \\ 1 \end{bmatrix} \updownarrow 1.2\,AH,\quad B_V = \overset{\xleftrightarrow{0.2AW+1}}{\begin{bmatrix} 1 & . & 1 \\ . & . & . \\ . & \boxed{1} & . \\ . & . & . \\ 1 & . & 1 \end{bmatrix}} \updownarrow 0.5AH+1$$

Figure 5.5: Creation of the morphological masks for horizontal and vertical line improvement, and their application sequence for obtaining the images I_H and I_V from the initial image I. $\ominus$ denotes denotes morphological erosion, while $\oplus$ denotes dilation [112]

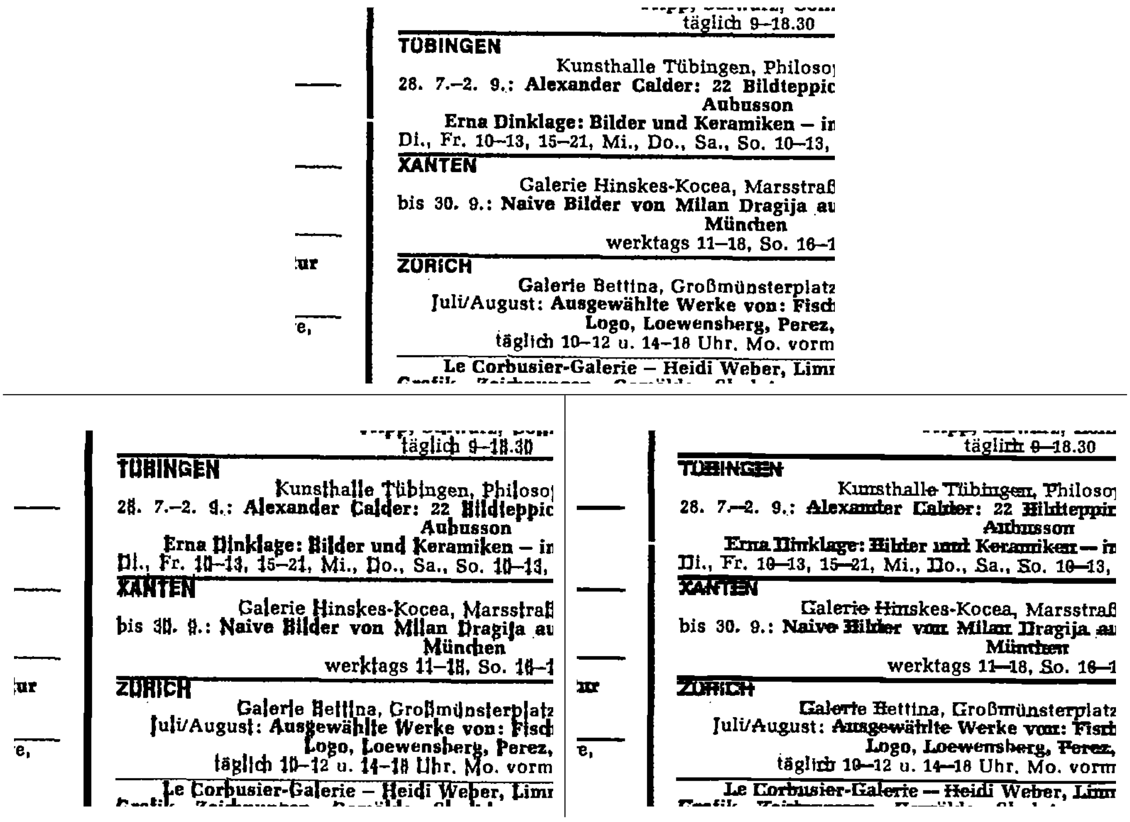
and ending y coordinates of R_i. A horizontal DSCC C_h is a sequence of vertical run-lengths: $\overline{R_1 R_2 \ldots R_m}$, where each two neighboring run-lengths R_i and R_{i+1} satisfy the following conditions: 1. they are connected in the horizontal direction, that is $x_i = x_{i+1}$ and 2. they are single-connected, meaning that on each side of R_i and R_{i+1} there exists a single connected run-length, and that run-length belongs to C_h. Naturally, the single-connectedness property is satisfied everywhere, except to the left of the chain's starting run-length and to the right of its ending run-length.

Ideally, a straight line segment is formed by a single chain. In reality, the printed separator is most commonly broken, intersected by characters, or subject to other (non-linear) deformations, and will be formed by more than one non-overlapped DSCC. For merging purposes, a straight line is fitted to each DSCC, by using the mid-points of its vertical run-lengths. Algorithms for finding the best linear fit for a given set of points are readily available in the specialized literature (e.g. [219], pp. 661–712). The "co-line distance", an indicator whether two DSCCs $C = \overline{R_1 R_2 \ldots R_n}$ and $C' = \overline{R'_1 R'_2 \ldots R'_m}$ lie on the same supporting line, can then be computed as follows:

$$d_{C'C} = \begin{cases} \infty, d_x \leq 0 \\ d_x + \dfrac{\sum_{i=1}^{m}\left(\frac{ys'_i + ye'_i}{2} - y(x'_i)\right)}{M}, d_x > 0 \end{cases},$$

where $d_x = max(x_1, x'_1) - min(x_n, x'_m)$; $y(x'_i)$ is the mid-point fitting of C; x_1 and x_n are the x coordinates of the left and right edges of chain C; x'_1 and x'_m are the x coordinates of the left and right edges of chain C'. Only the run-lengths having a size differing by at most a factor of two from the median run-length size in a DSCC are used to compute the co-line distance. Two DSCCs C and C' are merged in case either one of the following conditions is satisfied:

1. Gap- & linear extending condition: $d_x < 1.5 \times AW$ and $\sqrt{d_{C'C} - d_x} < 1.2 \times W$,



täglich 9–18.30

TÜBINGEN

Kunsthalle Tübingen, Philoso

28. 7.–2. 9.: Alexander Calder: 22 Bildteppic

Aubusson

Erna Dinklage: Bilder und Keramiken – ir

Di., Fr. 10–13, 15–21, Mi., Do., Sa., So. 10–13,

XANTEN

Galerie Hinskes-Kocea, Marsstraß

bis 30. 9.: Naive Bilder von Milan Dragija au

München

werktags 11–18, So. 16–1

ZÜRICH

Galerie Bettina, Großmünsterplatz

Juli/August: Ausgewählte Werke von: Fisch

Logo, Loewensberg, Perez,

täglich 10–12 u. 14–18 Uhr. Mo. vorm

Le Corbusier-Galerie – Heidi Weber, Limr

Figure 5.6: Results produced by the morphological line improvement using mask dimensions proportional to the dominant font. Top: portion of original image I; Bottom: its corresponding result images I_V (left) and I_H (right)

where W is the median height of the run-lengths in chain C

2. Black connectedness condition: the space between the two chains (on the supporting lines of both DSCCs) consists at least 75% of black pixels

At this point it is possible to discard all chains with a slope greater than a certain threshold, since in most cases we are only interested in nearly horizontal or vertical separators. The Fraunhofer DIU system uses by default a threshold of 1.5° for the allowed deviation from the horizontal/vertical. In order to eliminate all false separators created by the morphological optimization procedure within text regions, we make use of the image/text area removal procedure proposed by Gatos et al.. As a first step, all remaining chains are removed from the original image. A smearing procedure is then applied, once with a vertical mask of size $1.2\,AH$, and a second time (on the resulting image), with a horizontal mask of size $1.2\,AW$. The effect of this procedure is that all graphics, images or text will appear as individual connected components in the resulting image. Finally, all chains located within the bounding box of a connected component from the smeared image are eliminated.

Before concluding with the presentation of the enhanced separating line detection algorithm, a few noteworthy practical issues will be discussed. As mentioned in the introductory paragraphs of this section, the number of initial DSCCs extracted from a typical 4500 × 6300 DIN A3 newspaper image is in the order of 50 000 – 100 000, possibly more in case of a noisy image. We are able to reduce this number significantly (3 – 4 times) via the morphological improvement procedure. For DIN A4- or A5-sized documents, such as journals or books, this reduction is sufficient, even when a high scanning resolution (600

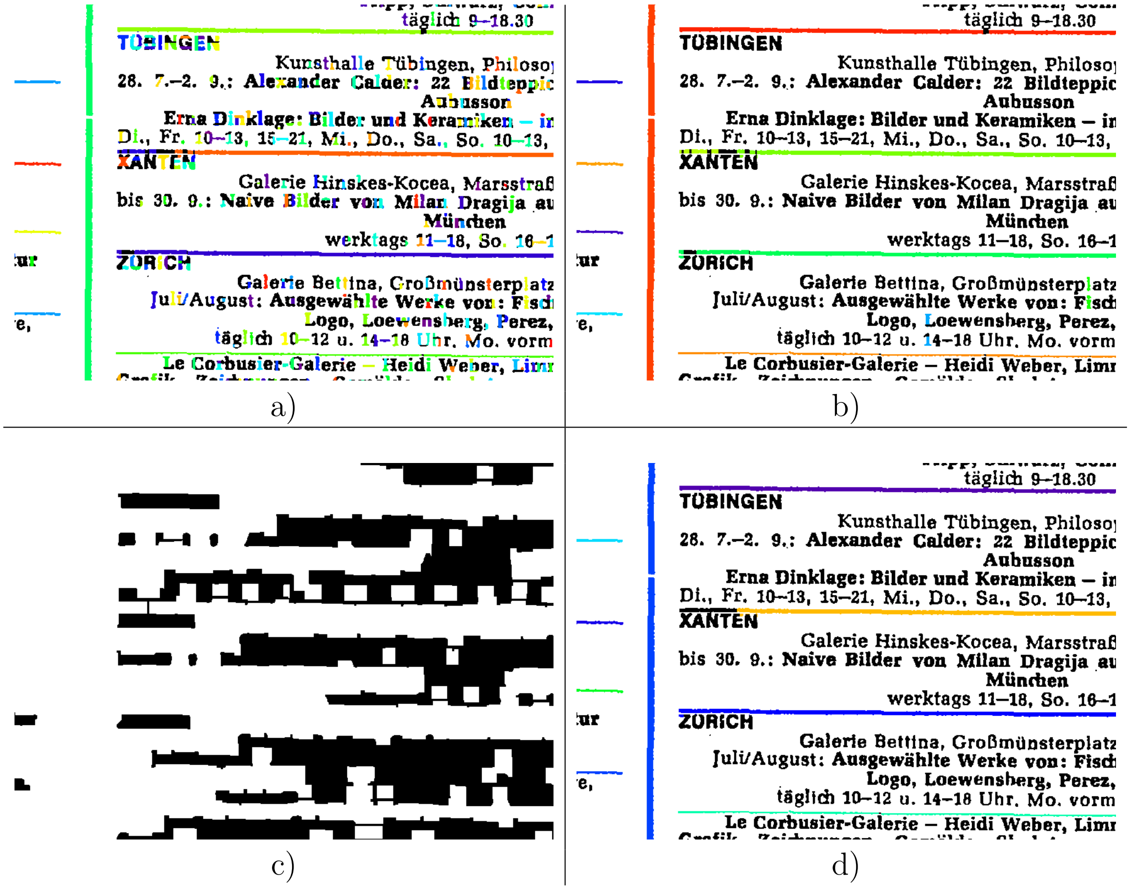


Figure 5.7: Illustrated steps of the proposed DSCC-based separator detection algorithm: a) initial DSCCs; b) image after the merging and discarding of short DSCCs; c) smeared image used as pruning mask; d) final set of solid separators

dpi) has been used for the digitization. However, in case of newspaper images (DIN A3 and above), the number of extracted DSCCs is still too large for the merging algorithm, which has a complexity of $O(N^2)$ in the number of DSCCs. Another reduction in the number of initial DSCCs (2 – 3 times) is to discard from the very beginning all chains shorter than 3 pixels in length. The merging procedure may be accelerated at the cost of accuracy as follows: we divide the list of vertical/horizontal DSCCs into portions having a pre-specified maximum number of elements. Then, the merging procedure will only be applied internally on the DSCCs within each resulting list segment. The list division should be done in such a way that as many potential mergings as possible will still be performed. One can achieve this goal by initially ordering the list of horizontal/vertical DSCCs by their minimum x, respectively y component and only afterward splitting of the separator lists. Finally, the smearing and morphological operators can be greatly sped up by using integral images. The integral image as a tool useful in image analysis was first described by Viola et al. [279], inspired after the concept of summed-area tables used in computer graphics since 1984 [81]. By means of the integral image, any rectangular sum of gray values within an image can be computed using just four memory access operations. Although for small document images and/or small local window sizes, this improvement is not directly noticeable, for larger document images it makes a huge difference in the required computing time (a factor of 100 is typical).

Until now we have seen how the Fraunhofer DIU system extracts a complete set of vertical and horizontal solid separators from a document page. For the sake of generality, we also extract all logical separators or *white spaces*. We require that all solid separators have been entirely erased from the input image. This allows us to detect the inter-column spaces much more robustly, since now two kinds of separators will potentially be used to describe it. While solid separators are in many situations broken (especially in low quality scans), white spaces act as a failsafe and can provide the needed separation constraints. Recently, two new algorithms for white space detection were proposed in [59] and [60], the latter also being able to detect rectangles at arbitrary orientations. Both algorithms employ a branch-and-bound search strategy for extracting the set of maximally empty (rotated) rectangles and are guaranteed to return globally optimal solutions.

For execution speed reasons and because of the high reliability of the skew and orientation detection algorithm introduced in section 4.2.2, the Fraunhofer DIU system employs the algorithm with no rotation tolerance. A few noteworthy changes have been made to the base algorithm, which we describe in the following. Most importantly, we skip the detection of horizontal white spaces entirely. In practice, we have found horizontal white spaces to be highly unreliable (e.g. frequently occurring between figures and their captions or between titles and the article bodies), and pruning such spurious occurrences presents a more complex problem in itself. After the application of the pruning rules described by Breuel [59], we additionally discard all superfluous separators. A superfluous white space is one that does not have non-white space regions on both sides in its direct vicinity. Finally we employ a refinement procedure in order to reduce the size and/or split the separators to their exact useful length. The first part of the refinement procedure consists in splitting all separators located directly between two vertically-overlapping connected components and having a width of less than 0.8 times the average height of the two adjacent components. This heuristic is necessary on pages with large font differences (such as large titles vs. body text) in order to prevent the splitting of textual regions featuring a large font size. Since separators are henceforth regarded as hard (non-intersection) constraints for all further purposes, the importance of this step becomes clear. The second and final part of the refinement procedure consists of the trimming of all separators to their exact necessary size. In practice it was found to be highly desirable to trim the separators at both ends so that they end exactly above or below the last layout elements located in close proximity to the separator. Needlessly long separators, when regarded as hard constraints often interfere with the correct functioning of the (generalized) text line creation (as introduced in the following section). Most likely to be affected by untrimmed separators are the larger layout elements, such as titles or bright halftones.

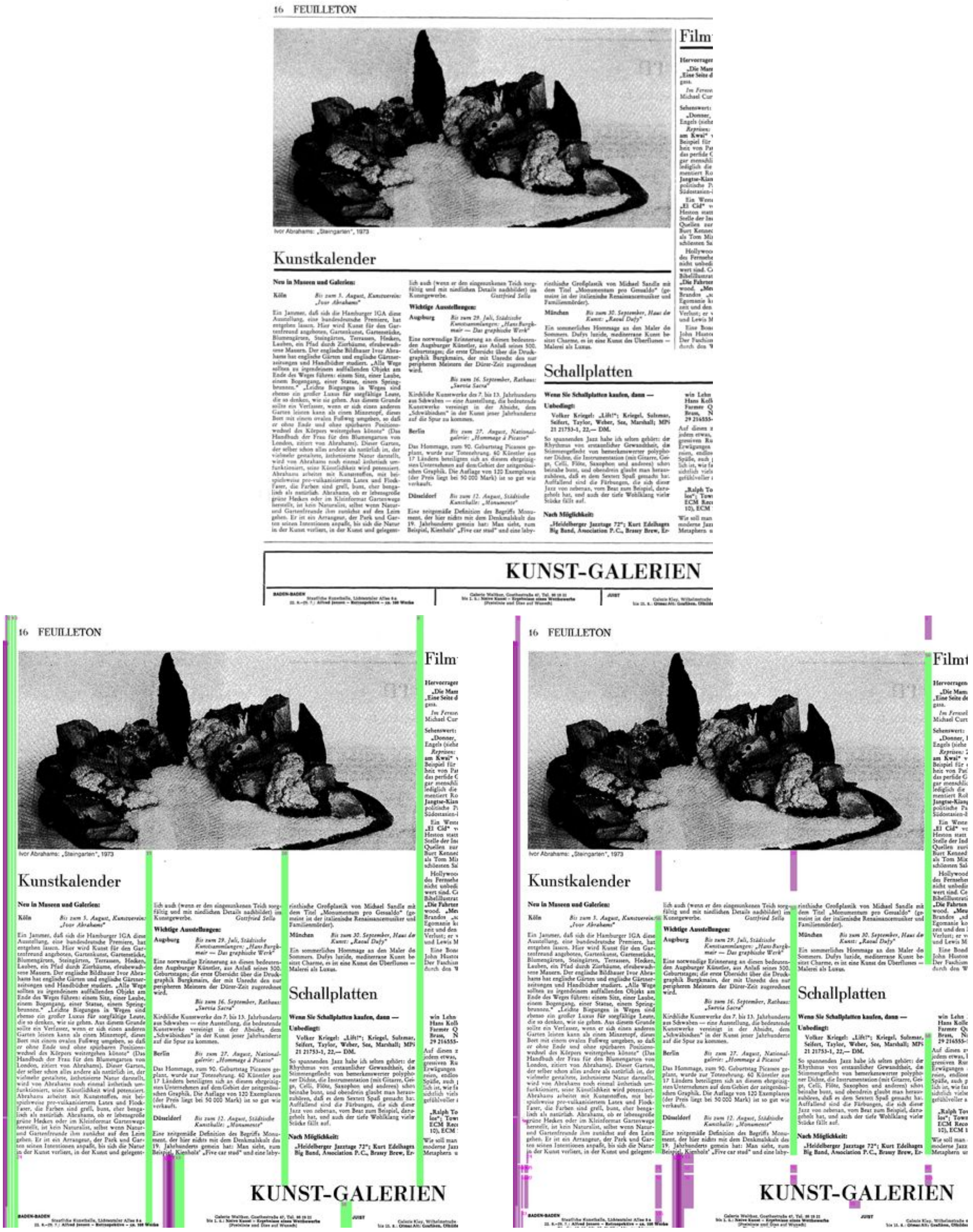


Figure 5.8: Top: portion of original newspaper image; Bottom-left: initial set of detected vertical whitespaces; Bottom-right: final set of vertical whitespaces after cutting using horizontal solid separators, deletion of redundant whitespaces and exact resizing. Green – valid whitespaces; magenta – discarded whitespaces

5.2.2 Region Detection and Classification

The original algorithm of Jain and Yu [142] belongs to the class of pure bottom-up page segmentation algorithms. Its main strengths are its inherent design versatility and the descriptive power of the resulting document representation. These features make it in our view a perfect candidate as a framework for testing and building different specialized algorithms combinations. In this section, using Jain and Yu's framework we describe a hybrid, resolution-independent algorithm capable of handling isothetic polygonal regions and multiple text orientations on a single page.

As precondition, we require that the document image be binary, margin noise-free, show no apparent global skew and contain no solid separators. The method for solid separator detection presented in the previous section allows a pixel-accurate representation of the separators, which in turn can be used to erase them from the input document image.

The first step of the algorithm consists of the *labeling of connected components*. For achieving this goal, the Jain and Yu algorithms creates the block adjacency graph (BAG) [299] in one pass over the image and then uses the nodes from the created BAG to extract the connected components. In the Fraunhofer DIU system, a different method is used for labeling the connected components. The method is based on the algorithm described in [90] and allows the efficient, constant-time extraction of connected components without the need of any additional memory beside the label image.

Subsequently, the algorithm of Jain and Yu creates a *hierarchical representation* of the document, by grouping connected components into generalized text lines (GTLs) and the GTLs into region blocks. The hierarchical representation of the document makes it easy to extract features at any level of the model, downward to single pixels and upward to region (i.e. logical) blocks. We retain the same representation scheme, while generalizing the scheme for building the GTLs and the higher-level regions. For consistency reasons, we will use in the following the same notations as in the original paper [142]. Any object o within the hierarchical representation will henceforth be regarded in terms of its bounding box with the left, right, bottom and top coordinates denoted respectively as $X_u(o)$, $X_l(o)$, $Y_l(o)$, $Y_u(o)$. The horizontal and vertical distances between any two objects in this case are:

$$D_x(o_i, o_j) = max(X_u(o_i), X_u(o_j)) - min(X_l(o_i), X_l(o_j)),$$
$$D_y(o_i, o_j) = max(Y_u(o_i), Y_u(o_j)) - min(Y_l(o_i), Y_l(o_j))$$

If $D_x(o_i, o_j) < 0$, we say that the objects o_i and o_j overlap horizontally, whereas if $D_y(o_i, o_j) < 0$, the two objects are said to overlap vertically.

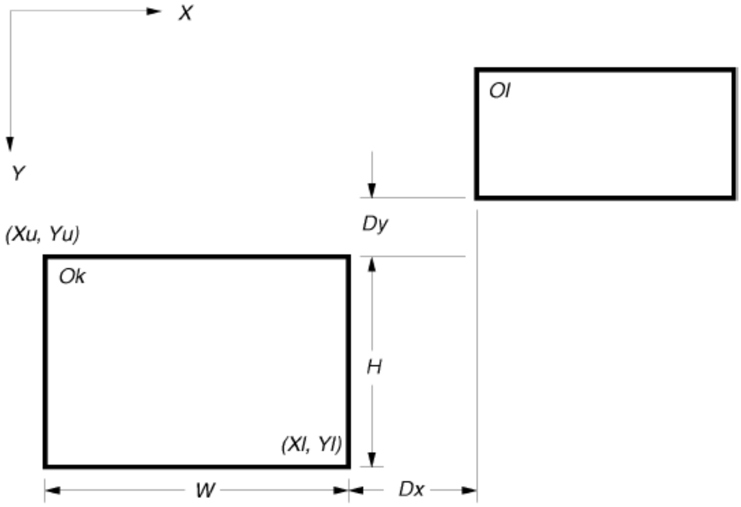


Figure 5.9: Visualization of object coordinate notations and inter-object distance calculations using bounding boxes (axis-aligned rectangles) [142]

Given a set O of objects, which contains at least two elements, we say that the objects are horizontally close and vertically overlapped (denoted $HClose(O, T_d)$) or vertically close and horizontally overlapped (denoted $VClose(O, T_d)$) at a distance T_d, if $\forall o_j, o_k \in O, \exists \left\{o_j, o_{j_1}, o_{j_2}, \dots, o_{j_p}, o_k\right\}$ such that $\forall o_{j_l} \in O, l = \overline{1,p}$ and

$$\begin{cases} D_x(o_j, o_{j_1}) < T_d, V_y(o_j, o_{j_1}) > 0.25 \\ D_x(o_{j_p}, o_k) < T_d, V_y(o_{j_p}, o_k) > 0.25 \\ D_x(o_{j_{l-1}}, o_{j_l}) < T_d, V_y(o_{j_{l-1}}, o_{j_l}) > 0.25, l = \overline{2,p} \end{cases}$$

or, respectively:

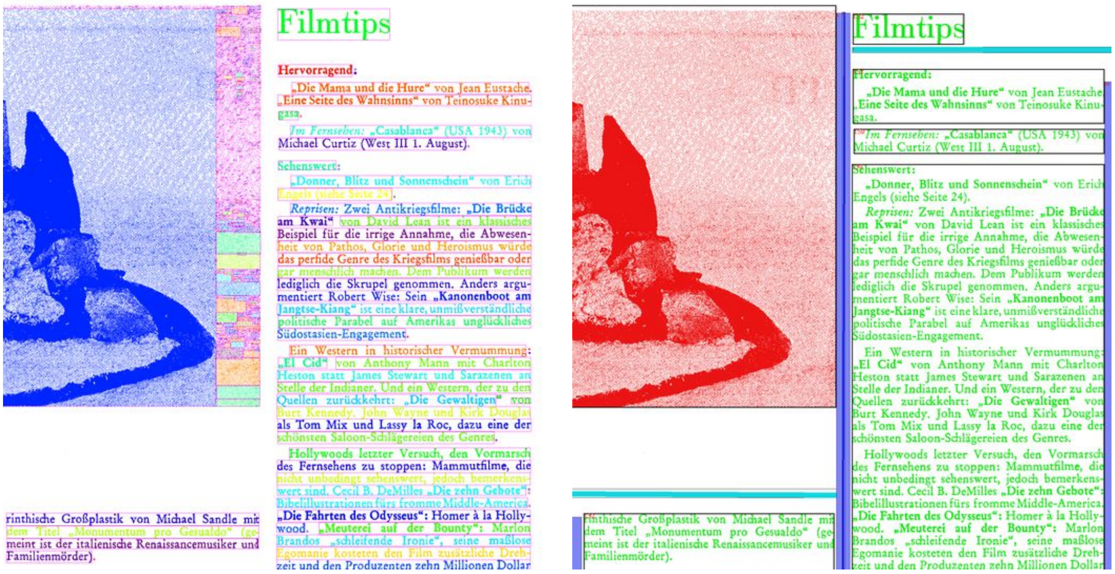
$$\begin{cases} D_y(o_j, o_{j_1}) < T_d, V_x(o_j, o_{j_1}) > 0.5 \\ D_y(o_{j_p}, o_k) < T_d, V_x(o_{j_p}, o_k) > 0.5 \\ D_y(o_{j_{l-1}}, o_{j_l}) < T_d, V_x(o_{j_{l-1}}, o_{j_l}) > 0.5, l = \overline{2,p}, \end{cases}$$

where

$$\begin{aligned} V_x(o_i, o_j) &= \frac{-D_x(o_i, o_j)}{min(X_l(o_i) - X_u(o_i), X_l(o_j) - X_u(o_j))}, \\ V_y(o_i, o_j) &= \frac{-D_y(o_i, o_j)}{min(Y_l(o_i) - Y_u(o_i), Y_l(o_j) - Y_u(o_j))} \end{aligned}$$

Using these additional definitions, one can now create one by one each GTL from the connected components as a maximal set O of objects which satisfy $HClose(O, D_h)$. GTLs are then classified into text and non-text as explained in the next paragraph. In a similar manner, region blocks are constructed from the obtained GTLs. A text region is a maximal set O of text GTLs which satisfy $VClose(O, 0.25cm)$, whereas a non-text region is a maximal set O of non-text GTLs which either satisfy $VClose(O, 0.25cm)$ or $HClose(O, 0.5cm)$. At his point, it is important to notice that the direct application of the GTL building algorithm with a static threshold D_h, as suggested by Jain and Yu, is very sensitive to the presence of noise. More specifically, if very small connected components are located at a distance less than the pre-defined threshold, the procedure will wrongly keep integrating them into the current GTL and expanding its bounding box. For enhancing the noise resilience of the GTL creation procedure, the Fraunhofer DIU system uses a dynamic threshold. The closeness threshold D_h equals twice the median height of the connected components contained in the GTL under construction. At the same time, only connected components with a height differing by at most a factor of 2 from the median height of the GTL are allowed to be appended. These conditions together hinder the creation of GTLs consisting primarily of noise, as well as the integration of noise components into otherwise semantically sound GTLs. It should be noted that in contrast to the original form of the Jain and Yu algorithm, we consider that new connected components/GTLs can only be appended if there exist no horizontal or vertical separators in between them and the GTL/region. The intersection tests, despite being necessary for every append operation, are still very fast, as they amount to simple line segment to bounding box intersection checks.

At the start of the region classification process, GTLs are categorized into text and non-text using simple statistical features. At this point, a basic, fast classification is entirely sufficient for our purposes. After obtaining the final regions however, it can be advantageous to use more powerful classifiers and features in order to guarantee the best labeling performance. For example, Keysers et al. [148] have recently investigated different types of features and feature combinations with regard to their classification performance for pre-segmented page regions. A GTL is categorized as text if either of the following conditions is met by its constituent connected components:



Filmtips

Hervorragend:

„Die Mama und die Hure" von Jean Eustache. „Eine Seite des Wahnsinns" von Teinosuke Kinugasa.

Im Fernsehen: „Casablanca" (USA 1943) von Michael Curtiz (West III 1. August).

Sehenswert:

„Donner, Blitz und Sonnenschein" von Erich Engels (siehe Seite 24).

Reprisen: Zwei Antikriegsfilme: „Die Brücke am Kwai" von David Lean ist ein klassisches Beispiel für die irrige Annahme, die Abwesenheit von Pathos, Glorie und Heroismus würde das perfide Genre des Kriegsfilms genießbar oder gar menschlich machen. Dem Publikum werden lediglich die Skrupel genommen. Anders argumentiert Robert Wise: Sein „Kanonenboot am Jangtse-Kiang" ist eine klare, unmißverständliche politische Parabel auf Amerikas unglückliches Südostasien-Engagement.

Ein Western in historischer Vermummung: „El Cid" von Anthony Mann mit Charlton Heston statt James Stewart und Sarazenen an Stelle der Indianer. Und ein Western, der zu den Quellen zurückkehrt: „Die Gewaltigen" von Burt Kennedy. John Wayne und Kirk Douglas als Tom Mix und Lassy la Roc, dazu eine der schönsten Saloon-Schlägereien des Genres.

Hollywoods letzter Versuch, den Vormarsch des Fernsehens zu stoppen: Mammutfilme, die nicht unbedingt sehenswert, jedoch bemerkenswert sind. Cecil B. DeMilles „Die zehn Gebote": Bibelillustrationen fürs fromme Middle-America. „Die Fahrten des Odysseus": Homer à la Hollywood. „Meuterei auf der Bounty": Marlon Brandos „schleifende Ironie", seine maßlose Egomanie kosteten den Film zusätzliche Drehzeit und den Produzenten zehn Millionen Dollar

rinthische Großplastik von Michael Sandle mit dem Titel „Monumentum pro Gesualdo" (gemeint ist der italienische Renaissancemusiker und Familienmörder).

Figure 5.10: Left: GTLs created from the connected components of a document image; Right: region blocks created from the GTLs (green: text; red: non-text; blue: vertical separator; cyan: horizontal separator)

- The height of the GTL is less or equal than *0.75 cm* and the standard deviation of the bottom edge of its connected components is less than *0.13 cm*.
- The height of the GTL is higher than *0.75 cm*, but its width is greater than *2.5 cm* and the ratio between the standard deviation and the mean values of the height of its connected components is less than *0.3*.

In both previous conditions, only those connected components are considered which have a height of at least 0.3 times the height of the containing GTL. Such components are considered to be noise, punctuation marks or dots belonging to letters, such as *"i"* or *"ä"*. After the non-text regions have been completely constructed, each of these regions is checked if it represents a halftone picture. In contrast to the original approach of Jain and Yu, we regard all remaining regions as belonging to the drawing class. Note that it is only possible to do this if the page is guaranteed not to contain any solid separators, incl. those part of tables or forms. In our case, this requirement was covered by the precondition. The basic criteria used to decide if non-text element represents a halftone picture are the following:

- Both the width and height of the region are greater than 1.3cm, and
- The ratio of the number of contained black pixels to its area is greater than 0.4.

At this point it is worth noting that the algorithm as presented above is indeed fully resolution-independent if the scanning resolution is considered to be known. For professional and semi-professional scans, this is indeed always the case. However, in case of camera-captured images the resolution is most often unavailable. In such situations, it is possible to obtain a reliable lower bound of the equivalent scanning resolution via the already computed dominant font character height feature (see section 3.1.3.1). The character height provides a baseline for computing the lower limit of the equivalent scanning resolution by assuming it to be equal to the minimum easily human-readable character height, which is about 0.09cm.

We are now able to introduce a series of generalizations which shall allow the presented algorithm to be seamlessly applied to document featuring *multiple skew*, without any limitation with respect to the to the skew angle differences among the regions. The reader must take note that the generalized framework presented in the following has not yet been subjected to rigorous testing – i.e. all tests in section 5.3 were obtained using the single-skew algorithm variant. To our knowledge, currently there exist no algorithms in the specialized literature which are able to accommodate for multiple skew without posing significant angle restrictions. For example, the original algorithm of Jain and Yu [142] limits the possible orientation variations to ±5°. From the algorithm description in the current section we can see that the parts hindering its multi-skew feasibility are the building of GTLs and subsequently the region building from the GTLs. The GTL building process can be made fully generic by introducing the notion of a *dominant skew direction*. We define the dominant skew direction of a connected component/GTL as the skew of the local homogeneous region to which it logically belongs. As such, if a textual region is printed using a 30° slant, this value also represents the main direction of all its contained GTLs and characters. The GTL building procedure via bounding box proximity search can also be applied in case of GTLs having a dominant direction different from 0° by simply considering a coordinate system rotated by the dominant direction. Once the GTLs "generalized" in this way have been constructed, the regions may also be built in a similar manner via the same dominant direction (merging for GTLs on the 90° rotated direction). As such, the key to a multiple skew generalization is the detection of a robust estimate for the dominant direction for each connected component. At this point, we must refer the reader to section 4.2, where we have introduced a generic, layout-independent framework for global skew detection. The same framework can be "localized" so that it computes the dominant direction, as well as a confidence value for each connected component. The localization of the skew angle for each connected component can be readily accomplished by restricting the histogram building step to use only the Euclidean MST edges connecting the components located at a distance of at most N steps from the considered component. A suitable value for the neighborhood radius N, such as 30 can be used. Note that the radius of the neighborhood is directly proportional to the minimum size of the detectable differently slanted regions. On one hand, a low value of N will allow the detection of very small slanted regions, on the other it will cause much more instability in the accuracy of the local skew detection. Therefore finding an optimal value for the neighborhood radius is an interesting direction for investigation. Dominant directions with a low corresponding confidence value may simply default to 0°, or be computed by interpolating between the dominant directions of its neighbors within the Delaunay triangulation (used as basis for constructing the Euclidean MST). An important observation is that the confidence values can be used to provide a strict ordering for the construction of GTLs, in that one may always construct the next GTL by taking as starting point the connected component with the highest confidence for the dominant direction. Finally, one may observe that the additional computational effort for the per-connected component dominant direction detection in comparison to the global skew detection is negligible. Most computational resources are necessary for the construction of the Euclidean MST – executed only once per document. Subsequently, the neighbors within radius N of each connected component can be determined via a quick breadth-first search in the tree.

Until now we have considered for computational efficiency purposes that all layout elements (connected components, GTLs, regions) are solely described by their bounding boxes. By taking advantage of the hierarchical structure one is however perfectly able to *describe Manhattan and non-Manhattan (arbitrary) layouts*. For example, regions can be refined as sets of GTL bounding boxes, which in turn may be refined as sets of connected component

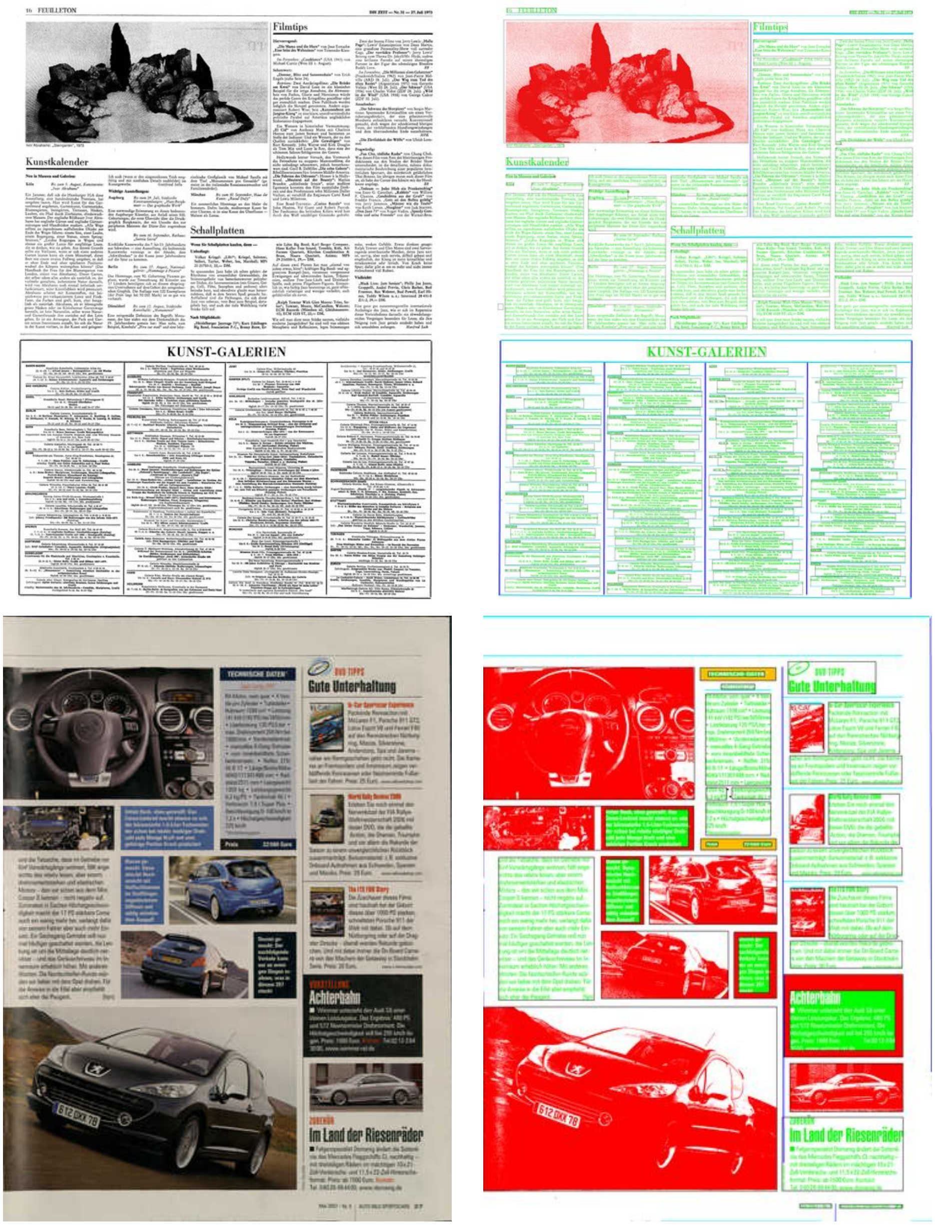


Figure 5.11: Examples of segmented document images: top – newspaper scan, bottom – magazine scan (green: text, red: halftone, orange: drawing, blue: vertical separator, cyan: horizontal separator)

bounding boxes. In most cases however, it is actually not desirable to have an overly accurate representation of certain layout elements. This happens because some of the components which should be contained in one such region may have been lost during the binarization step or simply ignored because of their small size. Common examples in the case of text areas are punctuation marks, accents, broken letter parts, while for non-text regions most usually encountered are bright portions of halftones from which only relatively small connected components remain. What we actually desire at this point is to obtain the coarsest page representation which is still as accurate as possible. In order to accomplish this, the Fraunhofer DIU system computes the exact intersection between each text and non-text region and splits one or both bounding boxes in such a way that the intersection becomes empty. The core idea is to convert the regions to an isothetic polygonal representation (i.e. set of bounding boxes) only where it is actually necessary. At this point we put priority on the exact description of textual regions, as they are much more valuable than halftones/drawings in view of further logical layout analysis and semantic processing (e.g. text mining). Therefore, we now assume all non-text regions to be rectangular (i.e. described solely by their minimum bounding axis-aligned rectangle), while the text regions will each be described by a set of bounding boxes representing the contained text lines. Considering all connected components from the non-text regions as unassigned, we can now apply once again the modified GTL creation algorithm, albeit starting from each existing text line. In this way, where appropriate, text lines will be extended into the drawings/halftones and the contours of the non-text regions shall be transformed into isothetic polygons only where intersections occur. As a final step, from the list of bounding boxes corresponding to each textual region, one can readily compute a minimal enclosing isothetic polygon.

A second important observation is the fact that although text regions are now well-separated from non-text, the text region merging procedure employed was inaccurate and thus regions with different physical or logical characteristics were merged together (such as titles merged with body text). As mentioned at the start of the current chapter, we do not consider that a more accurate separation is the responsibility of the page segmentation algorithm, since it may depend on higher-level (logical) layout information – e.g. indentation, detailed font characteristics, table cell alignment, column layout etc. . Therefore, all evaluation performed in section 5.3.1 on the newspaper dataset uses the output of the page segmentation algorithm in this state. However, for the magazine and journal dataset, as well as for the historical document dataset evaluated as part of the ICDAR page segmentation competitions a further refinement of the text regions was necessary. In order to obtain a more accurate segmentation of the text lines and regions, the Fraunhofer DIU system employs several steps which are mentioned in this work as part of the logical layout analysis. The respective steps and features are described in detail in section 6.3.1 and we will only briefly go over them here. We start by refining the dominant character size estimate, this time by taking only the connected components located within the text regions. By considering that the dominant font of the document is (nearly) always written in non-slanted characters having a regular stroke width, one can now reliably detect italics and boldface snippets. Information about the alignment of GTLs (now fully-fledged text lines), as well as a more accurate merging of vertically overlapping lines can be computed with the help of the layout columns (see section 6.3.1.1). The text regions computed as part of the page segmentation process are now completely discarded. Instead, by making use of the additional text and layout features, we compute the set of text regions minimizing a font and alignment style inter-line merging cost via dynamic programming. Most noteworthy is that the similarity/distance cost function is nearly universal, as the set of typographical features is relatively static and already well established in the publishing industry.

5.3 Evaluation

In order to achieve success in mass digitization, one must first be able to effectively process a wide variety of document types. Despite having employed the Fraunhofer DIU system in the processing of more than 1 000 000 scanned document pages with visually satisfactory results, only now we can make the claim that the test results are indeed sufficient to give a good idea about the DIU system's real-world capabilities. The exact quantitative evaluation of page segmentation results on real-life and heterogeneous document scans is only now starting to become a reality due to the recently proposed evaluation algorithms and rich data sets. The current section will present rigorous testing and comparison results for the Fraunhofer DIU system on several datasets with significant differences in composition and quality. At present we are not aware of any other DIU system capable of handling all datasets used in our tests. Consequently all comparisons are restricted to individual datasets.

Attempts to provide an adequate dataset for the testing of document analysis algorithms have been made in the past [215, 296], without much success until around the year 2007 [201, 239]. The most well-known document dataset preceding this date is the UW-III dataset which consists of around 980 journal images with relatively simple layouts and no significant noise. An interesting approach on automated newspaper ground truth generation was recently proposed by Strecker et al. [254]. Their method uses a 4-by-16 grid structure to place manually ground truthed elements on a page, using different possible styles mapped via a optimization procedure. The resulting documents can be printed, then scanned for better simulating normal conditions. While the Strecker et al. approach cannot match the variety of a dataset containing actual newspaper scans, it represents a good starting point, especially for page segmentation methods requiring a larger amount of training data.

A very promising and recent effort in this direction is being made starting from 1999 by the PRImA Research Lab, from the University of Salford (see [31, 37]). The PRImA dataset puts particular emphasis on real-life, color magazines and technical journals and the manual ground truth contains exact polygonal representations for all contained regions. It is in a state of continuous growth and development and was employed in the biennial ICDAR Page Segmentation competitions starting with 2003 until 2009. Recently, in 2011, the first ICDAR segmentation competition took place where the data set, also gathered by the PRImA Research Lab, consisted of mixed historical documents (books, chronicles, journals). As seen in the review section of the current chapter, evaluation methodologies have also been continuously refined along with the data sets.

In the following we shall discuss the results obtained by the Fraunhofer DIU system on a new dataset restricted to newspaper material, as well as on the ICDAR 2009 and 2011 page segmentation competition datasets [33, 38]. Since the three segmentation evaluations have been performed throughout a number of years, the evaluation methodologies show important differences and are presented along with the experimental results.

5.3.1 Newspaper Dataset

The newspaper dataset was put together in order to allow the evaluation of the system on the specific use case of newspaper mass digitization. In contrast to magazines, books and journals, newspapers pose a harder challenge regarding their varying multi-column layouts, much larger font size and style differences and common non-linear deformations for the solid separators. Also, since newspaper pages are generally much larger than book or magazine pages, they are also more prone to physical deformations (before or during scanning). Conversely, newspapers generally have simpler rectangular- or Manhattan layouts and rely much less on color as a means to provide vital information.

The ground truth gathered for testing purposes consists of 22 manually annotated newspaper images coming from 6 different German-language newspaper publishers: “Frankfurter

Allgemeine Zeitung" – 7 images, "Neue Zürcher Zeitung" – 3 images, "Liechtensteiner Volksblatt" – 3 images, "Saarbrücker Landes-Zeitung" – 2 images, "Süddeutsche Zeitung" – 3 images and "Völkischer Beobachter" – 4 images. All images were received in digital form from professional scan providers and feature varying resolution of 300 to 400 dpi. The image visual quality varies significantly, ranging from nearly perfect images (FAZ), to partially washed-out images (SZ) and moderately noisy images (SLZ). Additionally, the time period from which the newspaper pages originate ranges from the near-beginning to the near-end of the 20^{th} century. Having documents from different publishers allowed the evaluation of the response of the proposed system to different layouts exhibiting varied font types (Antiqua vs. Gothic/Fraktur) and sizes. Several newspaper pages partially containing advertisements were also included. This was motivated by the fact that the editorial content in newspapers nearly always has a Manhattan layout, unlike advertisements where the layout is much more varied.

In total, the number of ground-truth regions used in our evaluation was around 1500. Although this number is insufficient for an accurate performance benchmarking, we believe it is high enough to be able to get an good estimation of the overall strengths and weaknesses of the page segmentation modules. One must note that a typical newspaper page covers an entire DIN A3- or A2-sized paper, whereas books pages are most commonly A5-sized and magazines and journals A4-sized. Keeping this fact in mind, one may see that the size of the newspaper dataset is actually on par, if not larger than most other available data sets with respect to the printed surface to be analyzed. For comparison, the ICDAR 2009 test set [38] consists of 55 images depicting DIN A4-sized journals and magazines, thus roughly totaling $3.5m^2$ of printed material, whereas the newspaper dataset comprises around $4.1m^2$ printed using similar font sizes.

Figure 5.12: Sample scans from the newspaper dataset used in the evaluation, coming from three different publishers. Left to right: Frankfurter Allgemeine Zeitung, Süddeutsche Zeitung, Saarbrücker Landes-Zeitung

5.3.1.1 Evaluation Methodology

We have seen in the first section of the chapter that there exist several page segmentation evaluation methods. Because of their restricted availability for public use and/or for different platforms, a novel methodology was developed. The used evaluation methodology belongs to the class of *region-based* approaches, due to their improved informative value and versatility compared to text-based methods. We have seen that text-based methods only take into account the text information, whereas for complex documents it is in addition desirable to get a similar measure of goodness for non-text regions.

As in [296], the proposed methodology performs region matching on a per-pixel basis, by using only the foreground pixels from each region. This aspect is very important, allowing the algorithm to ignore small differences between ground truth and the obtained regions. Matching is done between each region belonging to the ground truth region set $G = \{G_1, G_2, ..., G_M\}$, and each one belonging to the obtained page segmentation $S = \{S_1, S_2, ..., S_N\}$. It is worth noting that theoretically this methodology for region matching would allow any region shape to be used, although in practice for performance reasons one may wish to limit the possible shapes to rectangular or isothetic polygonal regions. The area of a region is defined to be equal to the number of its contained foreground pixels. Using this definition, one can compute the matrix: $(K_{ij}) = Area(G_i \cap S_j)$, where $1 \leq i \leq M$, $1 \leq j \leq N$. In order to allow (slanted) separators and frames to be correctly evaluated, we have initially considered these region types to be *hollow* [296]. A hollow region contains only those foreground pixels within its shape which are not part of any non-hollow region from the same region set. The pixels belonging to each segmentation region from a given region set (either ground truth or obtained as a result) can then be efficiently computed by utilizing an extra image, representing an exact map of the respective segmentation (as suggested in [296]). In conformance to the common practice, the evaluation procedure was divided into zoning and labeling. Zoning refers to the correct decomposition of the document page into regions, disregarding their types, whereas labeling is concerned with the correctness of the assigned region types (here, for each foreground pixel). Two parameters influence the behavior of our evaluation methodology:

- Zoning tolerance ratio Z_r – belongs to $[0, 1)$. Represents the tolerance, as a ratio between the number of overlapped pixels and the area of a region. A region i is said to have a matching region j, iff (if and only if) $(K_{ij}) \geq Z_r \times Area(i)$. The lower Z_r is, the more precise the zoning procedure becomes, at the expense of not tolerating small inaccuracies in ground truth. Throughout testing, Z_r was set to 0.1 for separators, and to 0.075 for all other region types.
- Leftover tolerance ratio L_r – belongs to $[0, 1)$. Represents the leftover tolerance and is used to determine when a region is noise, was missed, split or mislabeled, i.e. the number of pixels "left out" is greater than the region's area multiplied by L_r. The leftover ratio was set to 0.2 for separators (as they are much more affected by noise) and to 0.075 for all other region types.

By using the aforementioned matrix K and the two parameters Z_r and L_r, the evaluation process computes for a given pair of page segmentations, the number of regions falling into each of the following categories:

- *Noise* (zoning). A region j from the computed region set S is considered to be noise, iff $\sum_{i=1}^{M} K_{ij} < L_r \times Area\,(S_j)$
- *Missed* (zoning). A region i from the ground truth region set G is considered to be missed, iff $\sum_{j=1}^{N} K_{ij} < L_r \times Area\,(G_i)$
- *Split* (zoning). A region i from the ground truth region set G, which was not missed, is considered to be split in two cases: a) if the number of matching regions from S is greater than 1; b) $Area\,(G_i) - \sum_{j=1}^{N} K_{ij} < L_r \times Area\,(G_i)$ and the number of matching regions from S is exactly 1 (i.e. an insufficient match).

- *Merged* (zoning). A region i from the ground truth region set G, is considered to have been (incorrectly) merged with another, if it constitutes a match for a region in S, along with at least one other ground truth region.
- *Mislabeled* (labeling). A region i from the ground truth region set G, is considered to have been mislabeled if $\sum_{j=1}^{N} K_{ij} \geq L_r \times Area\,(G_i)$, where only those regions j (from S) having a different label from the chosen region have been considered.

Additionally, in order to better characterize the errors of the page segmentation process, two additional properties are computed:

- *Mislabeling ratio* (labeling) – belongs to $[0, 1)$ and is computed solely for the mislabeled regions. Denotes for each region type, the average ratio of the mislabeled portions. By using this property, one can readily observe whether errors occurred because regions were not identified correctly (i.e. split), in which case this value is relatively low, or if the regions were usually identified correctly, but were simply mislabeled as another region type.
- *Mislabeling count* (labeling) – is a $T \times T$ matrix, where T denotes the number of all possible region types. For two region type indices i and j, the matrix element at location (i, j) contains the number of times ground truth regions of type i have been incorrectly mislabeled as belonging to type j. Note that due to region splitting, multiple types may have been (wrongly) assigned to the same region, thus it is possible for the sum of the values on row i to be greater than the total number of ground truth regions of type i actually evaluated.

It is important to notice that some of the previous definitions will possibly not be fully consistent if there exist multiple regions from the same set overlapping in the same image area. Therefore, a precondition of the evaluation algorithm is that every foreground pixel belongs to at most one region. Also, a significant problem was encountered when considering separators (or frame regions) as hollow and evaluating them together with the other regions (as was the case in [296]). Doing so will yield very few noise- or missed separators, as they will most likely be part of a text or drawing region. Moreover, in such case a mislabeling will be reported only if the respective separator is large enough in comparison to the enclosing region. This is certainly not what one usually expects from an evaluation of separator detection accuracy. This problem was corrected by performing the evaluation of the “hollow” region types separately from the other regions, and at the same time removing their “hollow” attribute in order to increase their noise resistance. In this way the evaluation results are much more robust and fulfil the consistency expectations.

5.3.1.2 Results

We start by evaluating the separator detection performance, as an important building block for page segmentation. Since white spaces are in effect just logical separators with no fixed bounds, it is unclear how or whether it makes sense to evaluate white space detection performance. Scientific papers concerning the evaluation of white space detection do not appear to exist currently. Because of this, we have decided to discard them from our evaluation entirely. Note that our decision does not imply that white spaces are not useful for the segmentation of newspapers, but is simply caused by the lack of a robust and expressive evaluation procedure.

One must observe that table 5.1 does not include columns for mergings or mislabelings, because such situations never occurred in the performed tests. Splits in the horizontal

	#	Noise	Missed	Split	Prec. (%)	Rec. (%)	F_1 Score (%)
Horizontal sep.	249	11	3	25	95.0	88.2	91.5
Vertical sep.	271	12	3	16	95.2	92.6	93.9
Total	520	23	6	41	95.1	90.5	92.7

Table 5.1: Solid separators detection results on the Fraunhofer newspaper dataset

or vertical separators occurred most commonly as the result of severely broken lines in the binarized newspaper images or, in a few situations because of separators having a non-standard appearance (see figure 5.13). The combined solid separator detection procedure achieves an F_1 score of 92.7%. This represents a clear improvement over the results reported by the individual algorithms in [112], namely 80.6% for the morphology-based algorithm and 70.1% obtained on the same data set by the original DSCC algorithm [302]. However, the test set used in [112] consisted of a more heterogeneous collection of document images, such as forms, magazines, scientific papers, tickets, bank cheques, certificates and handwritten documents. As we shall see for the other evaluation data sets, the Fraunhofer DIU approach for solid separator detection achieves an overall top performance.

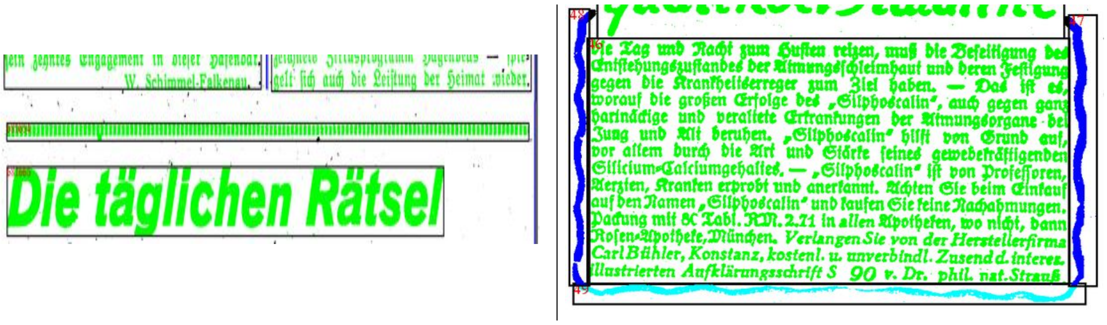


Figure 5.13: Results of solid separator detection: left – missed dashed horizontal separator (labeled as text due to the regularity and size of the dashes); right – correctly labeled wavy separators (green: text, cyan: horizontal separator, blue: vertical separator)

Next, we analyze the results for the overall page segmentation process. One must keep in mind that the segmentation results are heavily influenced by the correctness of the detected separators (both solid and white spaces). Another crucial observation is that we have intentionally ignored the splits and merges for text regions. As discussed in the introductory part of the chapter, we believe that page segmentation can only be expected to detect geometrical regions, and not also include a logical analysis and merging/splitting the text (as would be required for headers, captions, etc.). For example questions like "should page segmentation be able to differentiate between text areas/words printed in bold/italics from areas printed in regular fonts?" or "when is the font size difference large enough to warrant the split of a text region?" can only be reliably answered in the context of logical layout analysis. The answers to such questions generally require a certain semantic knowledge about the layout type employed by the publisher. The layout geometry and semantics, font properties and printing quality change drastically from one publisher or data set to another, even for the same publisher over the years. Thus settling such issues in a generic and deterministic manner at the level of geometric layout analysis is virtually impossible.

	#	Noise	Missed	Split	Merged	Mis-labeled	Misl. ratio	Prec. (%)	Recall (%)	F_1 Sc. (%)
Text	917	6	2	–	–	35	0.79	99.3	95.9	97.5
Halftone	19	0	0	0	0	0	0.0	100	100	100
Drawing	43	0	0	12	12	18	0.87	100	58.1	73.5
Total	979	6	2	12	12	53	–	99.3	94.3	96.7

Table 5.2: Geometrical layout analysis – segmentation and labeling results for the Fraunhofer newspaper dataset. Region types considered: text, halftone, drawing

We now look more closely at the obtained figures. An interesting insight is offered by the mislabeled ratio column, namely one can see that the mislabeled drawings and text regions, were in general not split, but instead mislabeled completely. This fact is significant because it shows that the zoning part of the algorithm outperforms the labeling part, thus the algorithm would benefit more by improving the latter (e.g. by employing a larger set of discriminant features [148] and possibly a better classifier). Notice that a region may be at the same time split and merged, which is usually the case in drawings. The obtained results for text and image regions are close to those reported by Jain and Yu [142], despite the much more complex document dataset. Most of the test images used by Jain and Yu belong to the UW-III document database [215], which, as we have previously mentioned, contains predominantly journal pages (incl. many synthetic ones) featuring simple layouts [37].

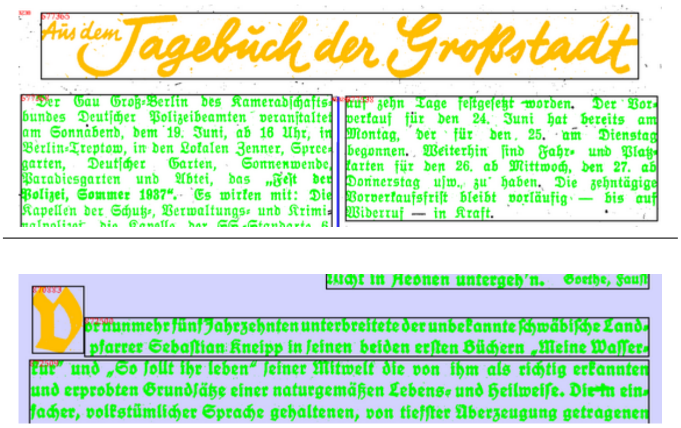


Figure 5.14: Examples of observed page segmentation labeling errors. Left: handwritten region identified as drawing; Right: Initial mislabeled as drawing. (green: text, orange: drawing)

In general, based on the values from the mislabeling count matrix (not included here), we observe that the most common errors for drawing regions are splits into text and image components. Typical examples are advertisements areas with complex layouts. In such situations a higher-level classifier (as part of the logical layout analysis) may be used to merge the parts into a cohesive region by using features based on the positioning, size and type of the constituent parts. The errors in case of text regions are of a different nature: they are nearly always segmented correctly, but labeled as drawings. This phenomenon is most frequent in case of slanted text, handwriting portions or paragraphs beginning with an initial. A remedy in this case would be the introduction of additional region types as part of the page segmentation process. The classification into the new region types can simply be appended as the last step of the algorithm. In section 5.1.3 we gave an overview of document analysis papers which investigate low-level features and classifiers with respect to their suitability for this task.

The average CPU time spent for the complete page segmentation of a newspaper page, scanned at 300 dpi (containing approx. 5000 × 7000 pixels) is about 15 seconds on a Core2Duo 2 GHz processor (using a single execution thread). More than 75% of the processing time is spent on separator detection (solid separators and white spaces).

5.3.2 Magazine- and Historical Document Datasets

The first dataset for evaluating the page segmentation results in the current section represent a subset of the PRImA dataset, maintained by the PRImA Research Lab at the University of Salford, UK [30]. According to its authors, the PRImA dataset contains a wide selection of documents featuring both complex and simple layouts, along with ground truth and extensive metadata. The ICDAR 2009 Page Segmentation Competition [38] dataset represents step forward in both complexity and size as compared to the previous such competitions [34–36, 113]. As training data a subset of 8 document images was selected by the organizers, while the disjoint testing set comprised 55 images showing similar characteristics. A few sample images can be seen in figure 5.15. Overall, the dataset contains color scans of contemporary magazines and technical journals with little noise, however exhibiting large font changes and layout complexity variations. A further important characteristic of the ICDAR 2009 dataset is the fact that physical page degradations are barely present and color information is paramount to the segmentation process (e.g. separators and body areas printed in light colors on bright background). As typical use case for the digitization of contemporary material we mention media research – e.g. extraction of product advertisement information, popularity- and sentiment analysis, etc.

In stark contrast regarding its characteristics stands the historical document dataset of the ICDAR 2011 layout analysis competition [33]. The historical document dataset was by comparison only recently created as part of the European IMPACT project [10]. The purpose of the dataset is to offer a good sample of the conditions and artifacts of historical documents, as present in the archives of many national European libraries. Furthermore, the composition of the dataset also reflects the needs and priorities of libraries, in terms of the types of documents which dominate their near- to medium-term digitization plans. The documents feature several languages and their age ranges from the 17^{th} to the early 20^{th} century. The 100 images comprising the competition dataset cover difficult page segmentation issues such as densely printed areas with minimal spacing, irregular spacing, margin noise, layout columns of varying width and margin notes. Most of the aforementioned layout particularities are only present in the historical document class, as opposed to the newspaper and magazine datasets. The training set for the ICDAR 2011 layout analysis competition consisted of 6 documents, whereas the evaluation set comprised 100 images (including the 6 training samples).

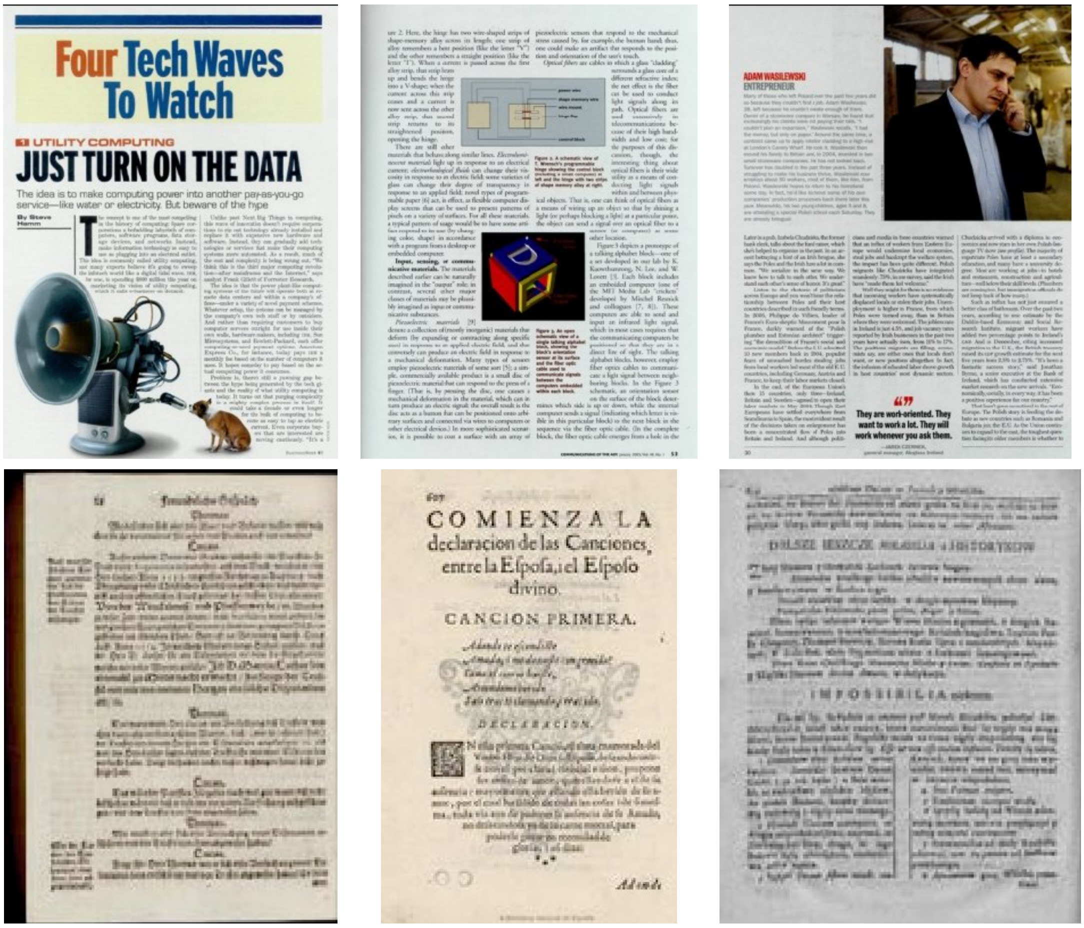


Figure 5.15: Sample scans from the datasets used in the evaluation. Top row: journal and magazine dataset (ICDAR 2009); Bottom row: historical printed document dataset (ICDAR 2011)

5.3.2.1 Evaluation Methodology

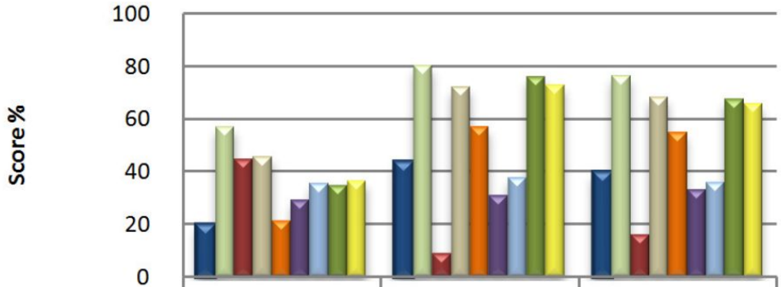
The ground truth for both datasets consists of exact isothetic polygonal representations of all document regions together with type information (categories: text, line-art, halftones, separators). Note that although more metadata is available in this case for the regions (e.g. reading order, provenience, OCR results), it is unnecessary for the purpose of strictly evaluating page segmentation results. An important particularity of both datasets is that the ground truth text regions are delimited with respect to their semantics – i.e. titles, sub- and intermediary titles, paragraphs and captions represent distinct regions, even where the font characteristics differ only little from adjacent regions. In other words, for such regions higher-level semantics need to be employed as part of the automatic page segmentation. The additional logical layout analysis steps performed were described in the last paragraphs of section 5.2.2.

We have seen the main ideas behind earlier region-based evaluation approaches in section 5.1.4. Until the year 2007 (including the ICDAR 2007 Page Segmentation Competition [35], it was typical for such methods to operate in a fully generic manner. More specifically, (pixel-)exact intersections for two sets of regions (represented as polygons) are computed and the matching score depends solely on the ratios between the intersection and ground truth areas. A serious disadvantage of the generic approach is its inability to differentiate between error types – for example merging two consecutive paragraphs on the same layout column is considered to be as severe as merging two paragraphs located on different layout columns. For practical purposes however, such as a further manual correction or simply visual quality, differentiating between error types is crucial for obtaining a segmentation quality value that is consistent with human expectations. Even more significantly, the goals of the document image understanding process are constantly subject to change (e.g. for different projects, individuals, libraries). For some, OCR quality is paramount, for others the extraction of halftones (e.g. from photo albums) is the main goal, or in yet another case (e.g. for preservation purposes) one must be able to reproduce the original as exactly as possible using only the discovered, compressed semantic representation. The ICDAR 2009 evaluation method [29] was the first to introduce scenario-driven evaluation and assess the participating methods under different application scenarios. A scenario is defined as the complete set of weights for the possible error types occurring during page segmentation. For instance, in a scenario where only the extraction and quality of text is relevant, the weights for all other regions types can be set to 0. Common application scenarios include general recognition, text indexing, full text recognition, image and graphics extraction [79]. More recently, Clausner et al. [79] expanded upon their initial evaluation method via a more comprehensive error type classification, while maintaining the same region-based framework. The evaluation methods allow a fine-grained specification of the scenarios via approx. 800 hierarchically-grouped error type weights. The Clausner et al. [79] method was employed in the ICDAR 2011 Historical Document Layout Analysis Competition.

5.3.2.2 Results

The 2009 ICDAR Page Segmentation competition was the first one to use the novel scenario-driven evaluation. This change had the positive side-effect of allowing the organizers to re-compute the results of all previously submitted methods to the ICDAR competitions into meaningful quality values and present them side-by-side (see fig. 5.16). Note that a pure, unweighted region-based approach would have generated nearly indinstinguishable scores for the methods (see [38]). The other methods compared in figure 5.16 are: DICE, a research method proposed by the Document Analysis Group of the Lehigh University (USA); REGIM-ENIS, a multi-script, multi-lingual method by M. Benjelil from the University of Sfax (Tunisia); Google's Tesseract page segmentation module [250]; the leading commercial OCR engine Abbyy Finereader, version 8.1; the open source OCR engine OCRopus, version 0.3.1, developed by the Image Understanding and Pattern Recognition group of the German Research Center for Artificial Intelligence (DFKI); BESUS, a research method from the Bengal Engineering and Science University (India) – winner of the 2007 ICDAR Page Segmentation Competition; and two methods from the Tsinghua University (China) – one of them winner of the 2005 ICDAR Page Segmentation Contest.

It is interesting to observe that the obtained results correlate well with the evaluation results for the newspaper dataset from the previous section. Although the two datasets have an entirely different composition, a clearly distinguishable trend is that the segmentation of textual regions versus the non-text ones was consistently better. This is in concordance to our goals, as in our view text regions contain more valuable information than drawing/halftones for the purpose of further semantic processing. The text-weighted application scenario evaluated in the competition (see [38]) shows the text-biased character of the current Fraunhofer DIU system even more markedly: for text regions the quality



	Non-text	Text	Overall
DICE	19.58	43.16	39.12
Fraunhofer	56.93	80.15	76.45
REGIM-ENIS	44.70	9.14	16.08
Tesseract	45.85	72.06	68.44
FineReader	21.44	57.00	54.90
OCRopus	29.26	31.03	32.98
BESUS	35.63	37.68	35.85
Tsinghua1	34.91	76.12	67.59
Tsinghua2	36.58	72.93	65.70

Figure 5.16: PRImA segmentation quality measure (generic scenario) for the ICDAR 2009 Page Segmentation Competition dataset, consisting of 63 scans from contemporary magazines and journals. Source: [38]

difference to the next-best method grows significantly (50% difference). Another observation is that the relatively poor performance in the case of non-text was mainly caused by difficulties encountered in the color reduction algorithm. More specifically, because of low contrast between halftones and some separators and the document background (e.g. yellow on light grey), large parts of such non-text regions were simply not present (wrongly labeled as background) in the binary image used for page segmentation. Since the scores of the other methods on non-text are even lower, we infer that this represents a general problem which should be of interest to the document analysis community. We shed more light on color reduction-related issues for document scans in section 3.2.

The 2011 Historical Document Layout Analysis Competition [33] employed a refined version of the 2009 evaluation method in order to be able to accurately reflect the additional difficulties posed by historical documents. Figure 5.17 shows the side-by-side comparison of the participating methods, including the Fraunhofer DIU system. The other methods depicted are: the leading commercial OCR engine Abbyy FineReader, version 9; MEPhI, a research method of A. Vilkin from the Moscow Engineering Physics Institute (Russia); Jouve, from the eponymous commercial organization specialized in digitization services (France); and EPITA, a method submitted by G. Lazzara of the Paris Graduate School of Computer Science (France). Note that for this dataset we have used exactly the same settings as for the newspaper and magazine data sets, with the sole exception of an additional margin noise removal. Several input images show significant margin noise in the form of black borders or large artifacts near the side(s) of the documents and its (partial) removal is crucial for the correct functioning of subsequent processing modules. In the

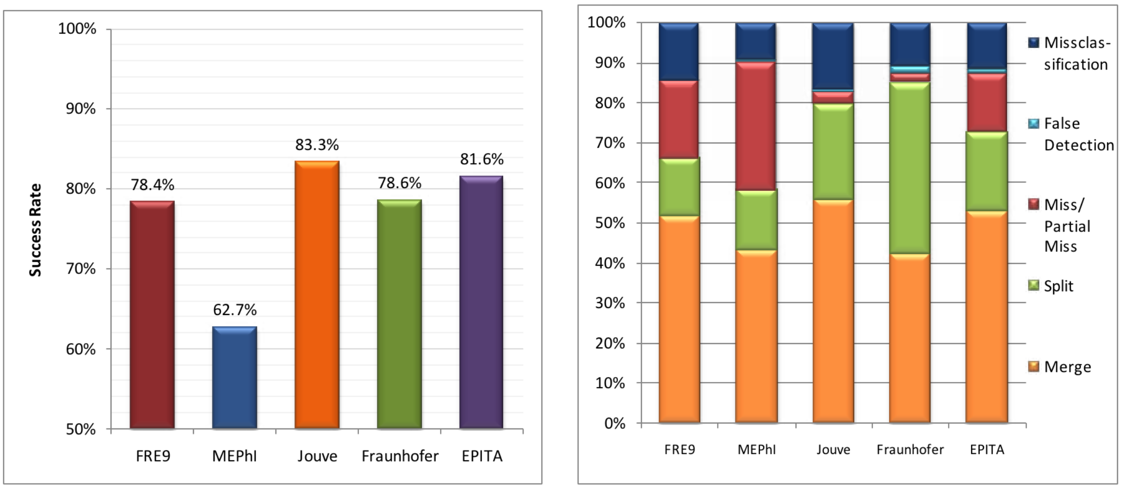


Figure 5.17: Left – PRImA quality measure (generic scenario) on the ICDAR 2011 Historical Document Layout Analysis Competition dataset, consisting of 100 scans from various historic books, chronicles and newspapers; Right – breakdown of error sources for each method. Source: [33]

introductory chapter, section 1.2.1 we have mentioned a few research algorithms proposed for solving this problem.

In general, one may see that despite the simpler layouts of the historical documents, the segmentation quality scores of all methods are far from perfect. Combined with the fact that the Fraunhofer DIU system was able to obtain about the same result on color documents featuring more complex layouts (i.e. the magazine dataset), the crucial importance of the pre-processing steps becomes apparent. In particular we note that artifact- and margin noise removal have a highly-heuristic nature and produce unreliable results when compared for example to global skew detection. The other methods participating in the ICDAR 2011 segmentation competition also employ specialized techniques to deal with artifacts and margin noise, ranging from adaptive, multi-scale binarization to filtering heuristics based on connected components. The imperfect artifact removal of the Fraunhofer system is directly reflected in the comparatively higher number of falsely detected regions (as seen in the error breakdown in figure 5.17). Unfortunately, since no other research methods could compete on the historical document dataset we are unable to gain a deeper insight into generic issues pertaining to historical documents. A problem specific to our system was the existence of pages featuring margin notes located close to the body text. Since we do not handle such notes any differently than all other text regions, and between them and the body text the spacing is too thin for detecting white spaces, most margin notes were wrongly segmented. A final issue is that several pages feature multiple skew (incl. text regions rotated by 90°). Although the described Fraunhofer page segmentation module allows for a variant which supports this layout feature, it was not enabled so as to be directly comparable with the system evaluated on the other two document datasets. Finally, we notice that the Fraunhofer system produces very good results on most images, but tends to fail catastrophically on a few, thus resulting in an overall lower score than expected. This seems to be the case not only on the historical document dataset, but also on the magazine and newspaper datasets. Detecting such failures automatically and marking them accordingly would be of great help for a subsequent prioritizing of the manual correction of the results.

5.4 Conclusions

The problem of page segmentation was discussed in detail in the current chapter. We started with a definition of the document page segmentation problem, introduced the three common types of layouts and put a clear delimitation between page segmentation and logical layout analysis. We continued with an overview of the state of the art in separator detection, page region segmentation, page region classification and evaluation methodologies.

In the context of mass digitization, we introduced the flexible page segmentation approach implemented in the Fraunhofer DIU system, featuring many improvements over the current state of the art in document image analysis. At the heart of the proposed region detection algorithm lies the generalized text line (GTL) structure, introduced by Jain and Yu [142], enriched with dominant direction information. In contrast to the Jain and Yu approach, we constructed isothetic polygonal regions in a fully resolution-independent and noise-resilient manner. For accomplishing this, we used as hard merging constraints the complete set of separators found in the document. To this end, we proposed a general-purpose solid separator detection method, obtained from the combination of two promising research algorithms [112, 302]. Robust and accurate white spaces are found using an enhanced version of Breuel's globally-optimal algorithm [59]. We extended the region detection algorithm for documents with arbitrary layouts (i.e. featuring text to non-text intersections) and introduced a variant allowing the segmentation of pages with multiple skew, while posing no restrictions on the allowable skew angle difference. The generality of the Fraunhofer DIU system was put to test on three datasets containing document scans with wide variations in layout complexity and image quality. Each dataset has a different target group as focus: newspapers, magazines and historical documents, respectively. To our knowledge, ours is the first DIU system able to obtain provably satisfying results on a large, heterogeneous dataset resembling "in-the-wild" mass digitization.

From the obtained segmentation results, we were able to identify promising directions for future work, not only for our system but for the document analysis community in general. Most importantly, we see that color information is still insufficiently exploited in document processing: for modern magazines many non-text elements are simply lost in later stages because of this deficiency, whereas for historical documents remaining degradation artifacts have a negative result on the segmentation quality. In section 3.2 we addressed this issue from a color reduction perspective. Page segmentation algorithms capable of natively operating on multi-thresholded images (i.e. N colors instead of just 2) represent a promising research direction. Finally, page segmentation algorithms capable of computing overall or per-region confidence values (or intervals) alongside the labeled polygonal regions would allow a more efficient manual correction and even self-improving systems.

Chapter 6

Logical Layout Analysis

Of all the species, yours cannot abide stagnation. Change is at the heart of what you are.

– M. Hurley and G. Roddenberry *(Star Trek: The Next Generation)*

In general, paper documents have an inherent logical structure emphasized by the use of many varied layout structures. In publishing, this is done with the purpose of greatly improving the comprehension of the contents by human readers [197]. The objective of logical layout analysis (LLA) is to (partially) extract these "hidden" semantics and thus provide a solid basis for a rich visual presentation, interlinking and intelligent search facilities for collections of electronic documents to a human user.

Under the hood, LLA builds upon the results of the page segmentation/ geometric layout analysis step. It further segments the physical regions into meaningful *logical units* according to their type (e.g. text lines, paragraphs), assigns a *logical label* to each of the segmented regions, as well as determines the *logical relationships* between the logical regions. The set of the possible logical labels is different for each type of document. For example: title, abstract, paragraph, section, table, figure and footnote are possible logical objects for technical papers, while: sender, receiver, date, body and signature emerge in letters. Logical relationships are typically represented in a hierarchy of objects, depending on the specific context [66]. Examples of logical relations are cross references to different parts of a book, the grouping of layout parts into independent articles/ sections, as well as the (partial) reading order of articles on a newspaper page. Taking these aspects into consideration, it becomes clear that LLA may only be accomplished on the basis of some kind of apriori information (knowledge) about the document class and its typical layout, i.e. a model of the document. The prior knowledge can be available in many different forms, such as heuristic rules, classification trees, formal grammars, probabilistic models (e.g. Hidden Markov Models) a.s.o., and is an important point where LLA algorithms differ. Note that in general it is not required (nor in many case even wanted) that all physical regions from a given document be assigned logical labels by the same method. Many LLA algorithms are purposely restricted to the detection or extraction of specific logical structures, some by design, others in order to be able to achieve a satisfactory performance. Prominent examples of such algorithms are the recognition of complete receiver address information from (handwritten) envelopes (e.g. US Postal Service projects [235]) and the automated generation of table of contents structures from books [87, 92] using only the geometric layout information and OCR results.

In this chapter we will put special focus on the processing of documents with complex layouts, like newspapers. In the case where the processed document type is a periodical, logical layout analysis will be referred to as *article segmentation*. We start with a brief overview of LLA methods proposed in the scientific literature. We then present a novel algorithm for front page detection specifically aimed at mass digitization projects. The proposed algorithms allows the semi-automatic segmentation of large periodical collection into issues and is thus indispensable for any further semantic processing. A more complex and complete logical layout analysis for periodical publications is performed by a novel minimum spanning tree-based technique detailed in section 6.3. It is worth noting that both methods have already been extensively used as part of the Fraunhofer DIU system in a few large scale (> 500 000 pages) document digitization projects. Both algorithms are evaluated rigorously on significant subsets of these large, real-life datasets and complete experimental results and observations are available in their respective sections. Finally, as basis for future work we present a theoretical approach to the logical region labeling problem based on Conditional Random Fields (CRF). As usual, the last section is dedicated to a short review, as well as offering several pointers towards challenging topics for future work in document logical layout analysis.

6.1 State of the Art

The number of LLA algorithms in the specialized literature is much lower than that of geometrical layout analysis algorithms. Even so, this section can only present the main ideas of several landmark methods and the interested reader is advised to consult one of the more recent layout analysis papers, such as [75, 109, 276]. Note that all dedicated surveys partially covering LLA [66, 126, 178] are currently outdated and only offer a good summary of early layout analysis methods (i.e. up to the year 2000).

In one of the earliest works on logical layout analysis, Tsujimoto and Asada [273] regard both the physical layout and logical structure as trees. They transform the geometrical layout tree into a logical layout tree by using a small set of generic rules suitable for multi-column documents, such as technical journals and newspapers. The physical tree is constructed using block dominating rules. The blocks in the tree are then classified into head and body using rules related to the physical properties of the block. Once this logical tree is obtained, the final logical labels are assigned to the blocks using another set of rules. The logical labels considered are: title, abstract, sub-title, paragraph, header, footer, page number, and caption. A virtual field separator technique is introduced, in which separators and frames are considered as virtual physical blocks in the physical tree. This technique allows the tree transformation algorithm to function with a low number of transformation rules. The authors tested their algorithm on 106 document pages from various sources and reported a logical structure recognition accuracy of 88.7%. The most common error causes were inaccurate physical segmentation results, insufficient transformation rules, and the fact that some pages did not actually have hierarchical physical and/or logical structures.

A general algorithm for automatic derivation of logical document structure from physical layout was described by Summers [256]. The algorithm is divided into segmentation of text into zones and classification of these zones into logical components. The document logical structure is obtained by computing a distance measure between a physical segment and predefined prototypes. The set of properties assigned to each prototype are the parameters from which each distance value is calculated. The properties include contours, context, successor, height, symbols, and children. Basic textual information was also used in order to obtain a higher accuracy. The algorithm was tested on 196 pages from 9 randomly selected computer science technical reports. The labeling result of each text block was characterized as correct, over-generalized, or incorrect. Two metrics, precise accuracy and generalized accuracy, were used to evaluate the performance. Both average accuracy values were found to be greater than 86%.

About at the same time, Niyogi and Srihari [194] introduced the DeLoS system for document logical structure derivation. DeLoS solves the logical layout analysis problem by applying a general rule-based control structure, as well as a hierarchical multi-level knowledge representation scheme. In this scheme, knowledge about the physical layouts and logical structures of various types of documents is encoded into a knowledge base. The system uses three types of rules: knowledge rules, control rules, and strategy rules. The control rules manage the application of knowledge rules, whereas the strategy rules determine the usage of control rules. A document image is first segmented using a bottom-up algorithm, followed by a geometric classification of the obtained regions. Finally, the physical regions are input into the DeLoS system and a logical tree structure is derived. The DeLoS system was tested on 44 newspaper pages. Performance results were reported in terms of block classification accuracy, block grouping accuracy, and read-order extraction accuracy.

Gao et al. [163] presented a system for comprehensive typography extraction from born-digital books (e.g. in PDF format). It is worth noting that their methods have been incorporated in a commercial electronic book publishing software package. While the restriction of born-digital documents may appear too harsh, the authors do so only in order to be able to skip all otherwise complex DIU processing steps up to and including page segmentation. The presented system is able to segment and classify body text, chapter headings, detect the page body area, headers, footers, layout columns and cope with multiple skew. The test set comprises 300 books with various styles and the authors report precision and recall values higher than 92% for each of the aforementioned tasks.

In the recent years, research on logical layout analysis has shifted away from rigid rule-based methods toward the application of machine learning methods in order to deal with the required versatility. There are several examples for this. Esposito et al. [97] employ machine learning in almost every aspect of document analysis, from page segmentation to logical labeling. Their methods are based on inductive learning of knowledge that was hand-coded in previous approaches. Chen et al. [75] use a set of training pages to learn specific layout styles and logical labels. An unknown page is recognized by matching the page's layout tree to the trained models and applying the appropriate zone labels from the best fit layout model. Similarly, the method of van Beusekom et al. [276] finds for a given unlabeled page the best matching layout in a set of labeled example pages. The best match is used to transfer the logical labels to the unlabeled page. The authors see this as a light-weight yet effective approach. Rangoni and Belaïd [220] use an artificial neural network as basis for their approach. Instead of a multi layer perceptron where the internal state is unknown, they implement a Transparent Neural Network that allows introduction of knowledge into the internal layers. The approach features a feedback mechanism by which ambiguous results can be resolved by proposing likely and unlikely results to the input layer based on the knowledge about the current context. The input layer can respond by switching between different feature extraction algorithms, e.g. for determining the word count in a given block.

The logical layout analysis methods described so far have not been evaluated rigorously on layouts more complex than journal papers. The complex, multi-column newspaper layouts are targeted by Furmaniak [109]. This is one of very few publications on the subject of article segmentation. It appears that the scarcity of publications in the area directly reflects the difficulty of the task. Despite the challenging nature of the problem, the author realizes that the mass digitization of newspapers represents the next step after the current wave of book digitization projects. He proposes a method for learning the layout of different newspapers in an unsupervised manner. In a first stage, a word similarity analysis is performed for each pair of neighboring text blocks. The second stage uses geometric and morphological features of pairs of text blocks to learn the block relations that are characteristic for a specific newspaper layout. Results with high confidence from the word similarity analysis serve as ground truth for the training of the second stage. The method of Furmaniak obtains promising results and further strengthens the machine learning approach to logical layout analysis.

It is worth mentioning that leading commercial OCR products such as ABBYY FineReader [1] (now at version 11) and Nuance Omnipage [14] (version 18) are also increasingly improving their logical layout analysis. Currently, both products claim to be able to recognize the structure of tables as well as identify large titles, headers and footers. In addition, FineReader can (partially) handle footnotes, recognize page numbers and detect the layout structure of documents. Interestingly, Omnipage recognizes the importance of a self-improving document analysis system and employs a machine learning approach in order to learn new layouts. Other less widespread DIU systems are also moving in the direction of automatic logical layout analysis (see section 1.2.2).

As introduced in the first chapter of this work, logical layout analysis regions use as base building blocks the geometrical regions produced by the page segmentation process. The shortcomings in page segmentation evaluation mentioned in section 5.1.4 have lead to the situation that, to the best of the author's knowledge, virtually no objective evaluation methodologies targeted solely at logical layout analysis exist. More recently, the new scenario-based evaluation methods used in the ICDAR page segmentation competitions [33, 38] partially incorporate logical layout segmentation rules and weights. Note that while both methods are still primarily focused on geometric layout analysis, a more substantial extension for logical layout analysis would be possible.

6.2 Front Page Detection

Mass digitization projects aimed at periodicals often have as input streams of completely unlabeled document images. In such situations, the results produced by the automatic segmentation of the document stream into issues heavily influence the overall output quality of a document image analysis system. This section will present a possible solution to for the robust and efficient detection of front pages in periodical collections.

As seen from the previous chapters, a large number of scientific articles have been published on topics concerning specific areas of document image analysis (see surveys [66, 178]). In comparison, the problem of front page detection for issue separation is still largely unexplored, although it is indispensable when working with large collections of unlabeled images of periodicals (such as magazines or newspapers).

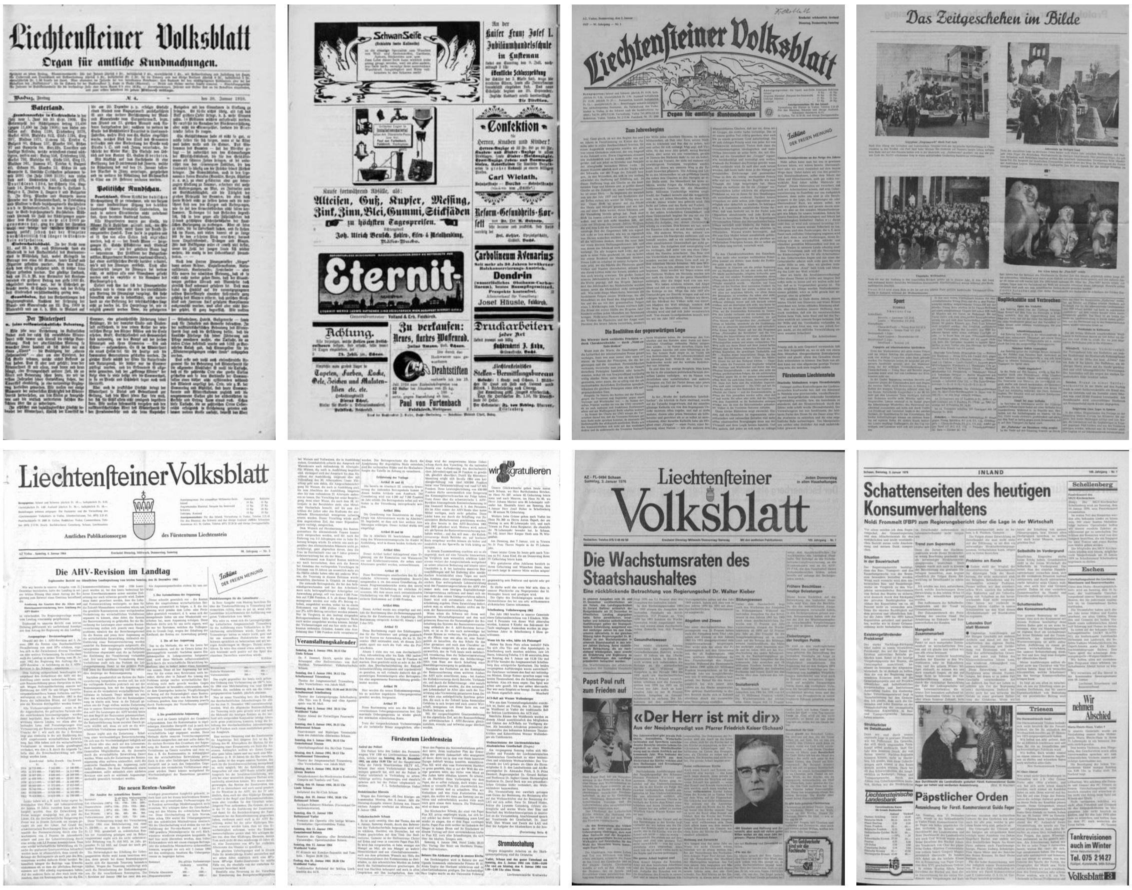


Figure 6.1: Selection of digitized images from an unlabeled German-language newspaper stream spanning about 70 years (Liechtensteiner Volksblatt). Note that in the original dataset the ratio between front pages and non-front pages is much lower, around 1:16

Two topics closely related to front page detection are logo recognition and structural pattern recognition. The detection and recognition of logos from document images are still being actively pursued in the document analysis community [73, 156, 304]. In the vast majority of papers dealing with logo recognition, logos are handled as unitary entities and various statistical classifiers (e.g. support vector machines, neural networks, Fisher classifiers, etc.) are used for the classification task. In contrast to the common approaches for logo recognition, in the area of structural pattern recognition objects are described by their topology or shape, most commonly encoded as different types of graphs. The graphs can afterward be matched exactly or error-tolerantly using existing methods, many of which have a strong theoretical basis (e.g. [184, 272]). Such approaches offer a great deal of flexibility, as proved by their successful application in domains as distinct as object recognition via skeletons or shapes [88, 232] and on-line graphics recognition [293]. Their main drawback however, is that structural approaches are inherently sensitive to noise, therefore specialized solutions are necessary for dealing with this problem. Noise robustness is of particular importance when working with real-world, error-prone document scans. Also, one must note that graph and subgraph isomorphism in general is an NP-complete problem, thus matching is only feasible for small graphs (exact speed/graph size measurements can be found in [184]). For special cases, such as closed contours, algorithms exist which are at the same time fast and guarantee the finding of optimal solutions [232]. In most other cases, in order to circumvent the graph size issue, different heuristics for approximating the optimal matches have been proposed [118, 293].

6.2.1 Methodology

The proposed algorithm works by computing and assigning a reference weighted graph to a front page model. As vertices in the constructed graph we use the salient connected components from the front-page specific elements, such as the title and the logo of the periodical. The exact methodology for the choice of salient components is described in section 6.2.1.2. Each of the salient components is subsequently described by a Gaussian distribution, approximated from a number of training samples. For all candidate pages, a weighted graph is computed in a similar fashion from those connected components which fit one of the trained Gaussian distribution models. Finally, the graph associated to each candidate page is matched against the front page graph. A correspondence is found if the matching score for the best correspondence exceeds the threshold value associated to the front page model. The methodology for calculating the edge weights and the threshold for each model graph is presented in section 6.2.1.4.

6.2.1.1 Pre-processing

We assume as input to the algorithm a deskewed and binarized document image. We have mentioned in the previous chapters many methods for page skew detection and document image binarization. It is important to note that most existing techniques for geometric and logical layout analysis of documents make the same assumption about the input document image [66]. This means that a regular document image analysis system will not require any additional (computational or implementation) effort in order to satisfy the precondition of the proposed front page detection method.

6.2.1.2 Salient Component Identification

From the binarized image, connected components (both black and white) must be labeled. This can be accomplished efficiently using any standard algorithm (e.g. [90]). In the case of noisy input images, it is advantageous at this point to apply certain hole-filling algorithms or morphological operators so as to remove a significant part of the white/black noise. This additional step is helpful (but not mandatory) for subsequently obtaining more consistent feature descriptors for the salient components. Note that the subsequent steps described in this section are only relevant for the training of the front page models and need not be performed for the test data.

Next, we choose the salient components from among the set of labeled connected components. Most commonly the chosen salient components will be single characters, connected character groups or logos specific to the front page model being considered. The choice of the salient components must currently be performed at least in part manually, as we are not aware of any fully automated selection method satisfying (most of) the criteria presented in the following. Some general guidelines for choosing the salient components are: salient components are not likely to be merged with other connected components even in the presence of noise, the selected set of salient components should span over as much of the title section area as possible and each component should be large enough so that the danger of it being mistaken for noise is minimized. For practical purposes, it is relatively easy to implement (as was done in our case) a semi-automatic salient component candidate selection method. This method can simplify the manual task considerably by filtering out those components which do not satisfy a set of simpler criteria, such as a minimum area, a certain aspect ratio and a high enough proximity to other candidate components.

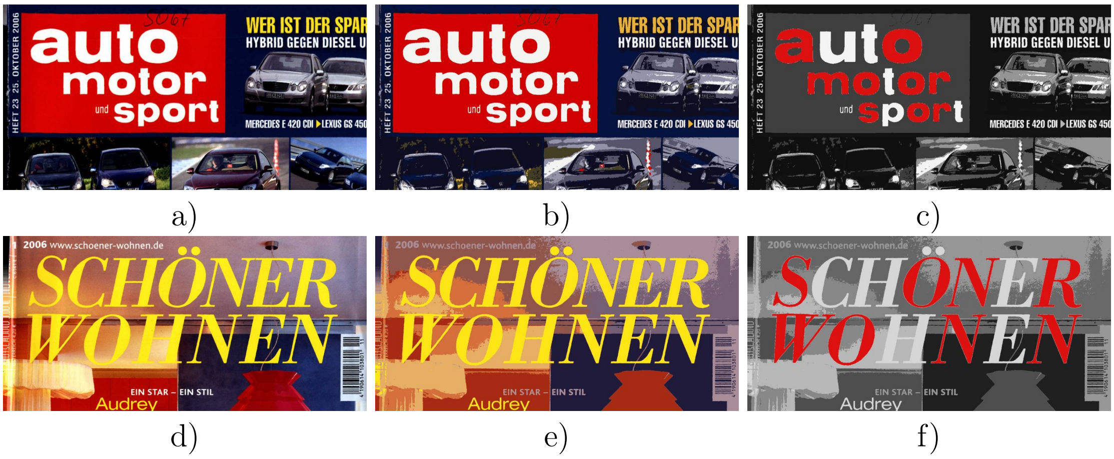

a) b) c)
d) e) f)

Figure 6.2: Illustrations depicting the main steps of the salient component selection process: (a),(d) – portions of original magazine front page; (b),(e) – color reduction result; (c),(f) – selected stable salient components marked in red

As an alternative for automating the choosing of the salient components, one may resort to one of the existing affine region detectors [186]. A good repeatability and accuracy is provided for example by maximally stable extremal regions [180], i.e. those parts of an image where local binarization is stable over a wide range of thresholds. The density of the detected stable regions contained in or partially overlapping a certain connected component can be used as a clue about the stability of the respective component. Although any salient region detector may be used, a very important factor to consider at this point is its runtime performance, which varies widely [186].

6.2.1.3 Feature Extraction and Matching

As features for describing a salient component, we have chosen to use the coefficients of the discrete cosine transform (DCT) applied to the component's bitmap. The DCT is known to be well-suited for the decorrelation of the pixels of images. It is used in the JPEG image compression standard [213]. In our case, the feature descriptor for the image of a component contains only the coefficients in the upper left triangle (low-pass) of the DCT transformed image. Early experiments showed that using the first 36 coefficients generally give good results, but decreasing or increasing the number of coefficients to 21, 28 or 45 produces very similar results. It is worth noting that the number of 36 coefficients was also determined to be the most appropriate from the visual experiments performed during the development of the MPEG-7 standard [139]. The feature extraction is similar to the one used by Eickeler et al. [95] for face recognition. In their paper, the authors show that the probability density function of the descriptors can be accurately modeled by a Gaussian distribution with a diagonal covariance matrix.

Note that it is entirely feasible to use a different set of features instead, such as those commonly employed for character recognition [270] or for generic content-based image retrieval [68]. As observed in [270], there exist no features which perform best in all situations and therefore the best feature type is in general highly dependent on the application

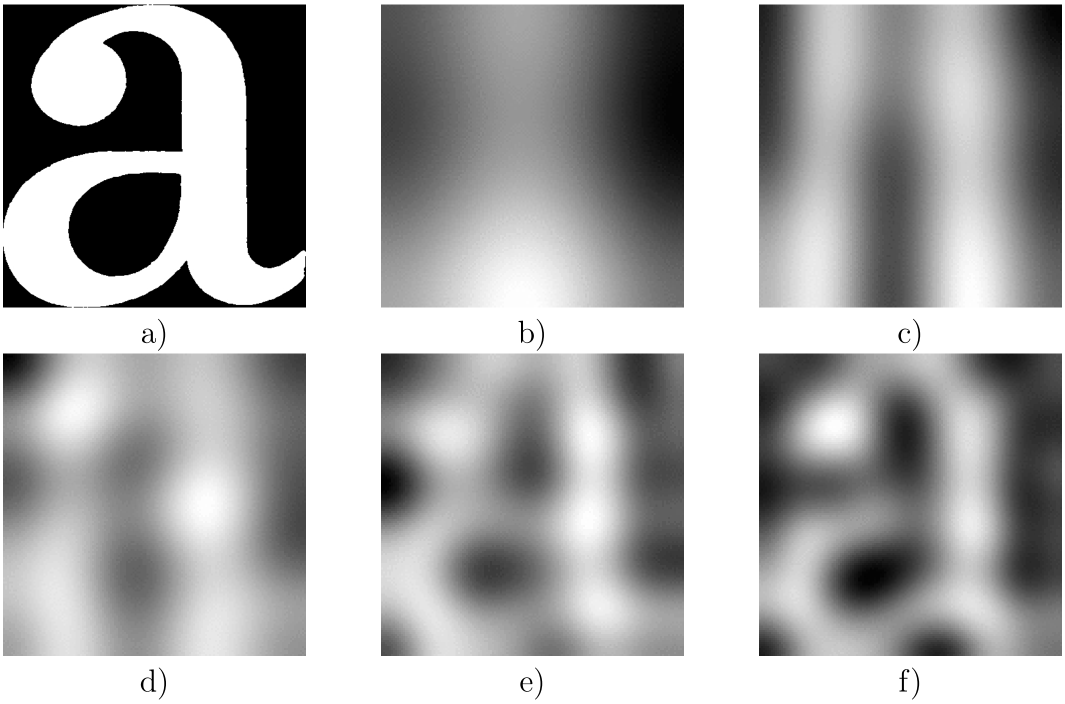


Figure 6.3: (a) Binary image depicting the latin letter "a" and its reconstruction using the most significant N DCT coefficients: (b) $N = 6$; (c) $N = 16$; (d) $N = 24$; (e) $N = 36$; (f) $N = 54$

domain.

An important reason for selecting the DCT was its scaling invariance property and the efficient algorithms available for its computation. The scaling invariance is important because many publishers frequently re-scale the size of their front page titles and/or logos (e.g. in order to accommodate for last minute news or advertisements). The DCT equation employed is:

$$C(u,v) = \sum_{x=0}^{W} \sum_{y=0}^{H} \frac{I(x,y)}{W\,H} \cos\left[\frac{(2x+1)u\pi}{2W}\right] \cos\left[\frac{(2y+1)v\pi}{2H}\right]$$

Here, W and H respectively represent the width and the height of the connected component, and $I(x,y)$ represents the image gray value at the position (x,y).

Other types of features, such as the Zernike moments, also offer rotation- and limited distortion invariance. In our case, the rotation invariance is rendered unnecessary by the skew correction already applied on the document (as a prerequisite). The distortion invariance is of very limited practical use for real-world document images, since such pronounced distortions are extremely rare in any professional scans, as commonly employed in large-scale digitization projects.

Using the Expectation-Maximization algorithm [183], a Gaussian distribution with a diagonal covariance matrix is now trained for each salient component having as input several of its feature descriptors (obtained from different example front pages). From the performed experiments, we have determined that between 5 and 10 samples per component are sufficient for obtaining an accurate Gaussian distribution model. In order to prevent overfitting, a relatively large floor value of 0.1 was used for the variances.

Given the set of trained Gaussian distributions, one can subsequently determine for any candidate component the distribution with the highest output probability for its feature vector. In case the output probability is higher than a certain apriori fixed threshold value, a match is considered to have been found.

6.2.1.4 Graph Construction and Matching

For modeling the topology of the identified salient components, we construct the point Delaunay triangulation [122] using as input the centers of their bounding boxes. As an alternative to the Delaunay triangulation we have also considered and tested the Euclidean minimum spanning tree (EMST) applied to the same set of points. Although the EMST has the advantage that the number of edges is about 3 times lower than for a triangulation (thus the graph matching problem is substantially simplified), the appearance of the resulting tree is more heavily influenced by small position changes of the tree vertices. This phenomenon can be observed easiest when (some of) the sites are almost uniformly distributed - in such case, even small perturbations of the node positions can produce very different spanning trees, whereas the Delaunay triangulation remains almost unchanged.

Even with the increased structural stability provided by the Delaunay triangulation, the obtained graphs are still sensitive to the loss of internal vertices. For enabling a robust graph matching, we additionally assign a positive real weight to each edge. Assuming that the probability of a node to be missing from the candidate graph is the same for all vertices, one can readily see that the probability of certain edges to be absent from the resulting triangulation is higher than for others. Our goal is that the edge weights are selected in such a way that for each node, the sum of the weights of the adjacent edges will always be equal to a certain constant. At the same time we need to minimize the absolute differences between the weights of each pair of edges, so that no edges will have a (near) zero weight. This is important because edges, just like vertices may also be absent from the graph and this fact should be directly observable from the matching score.

Unfortunately, the problem posed in this manner is NP-hard, because of its equivalence to the search for a solution with a maximum number of zeros in an underdetermined linear system, which was shown to be NP-hard in [191]. Because of this, one must settle for a heuristic choice of a function in order to compute approximates of the ideal edge weights. Several weighting functions were evaluated on our training set and the function which performed best in our tests is:

$$\text{Weight}(e) = \frac{1}{1 + (\deg(v_1) - \deg_{min}) + (\deg(v_2) - \deg_{min})} \tag{6.1}$$

Here, $e = (v_1, v_2)$ is an edge in the graph between vertices v_1 and v_2, $\deg_{min}$ is the minimum vertex degree in the graph and $\deg(v_i)$ is the degree of vertex v_i.

By taking into account the computed edge weights, graph matching can now be readily performed on an edge-by-edge basis. The matching algorithm traverses the edges of the model graph in decreasing weight order and searches for the best correspondence within the candidate graph. If a possible correspondence is found, the total distance between the two graphs is incremented by the model edge weight multiplied with the modulus of

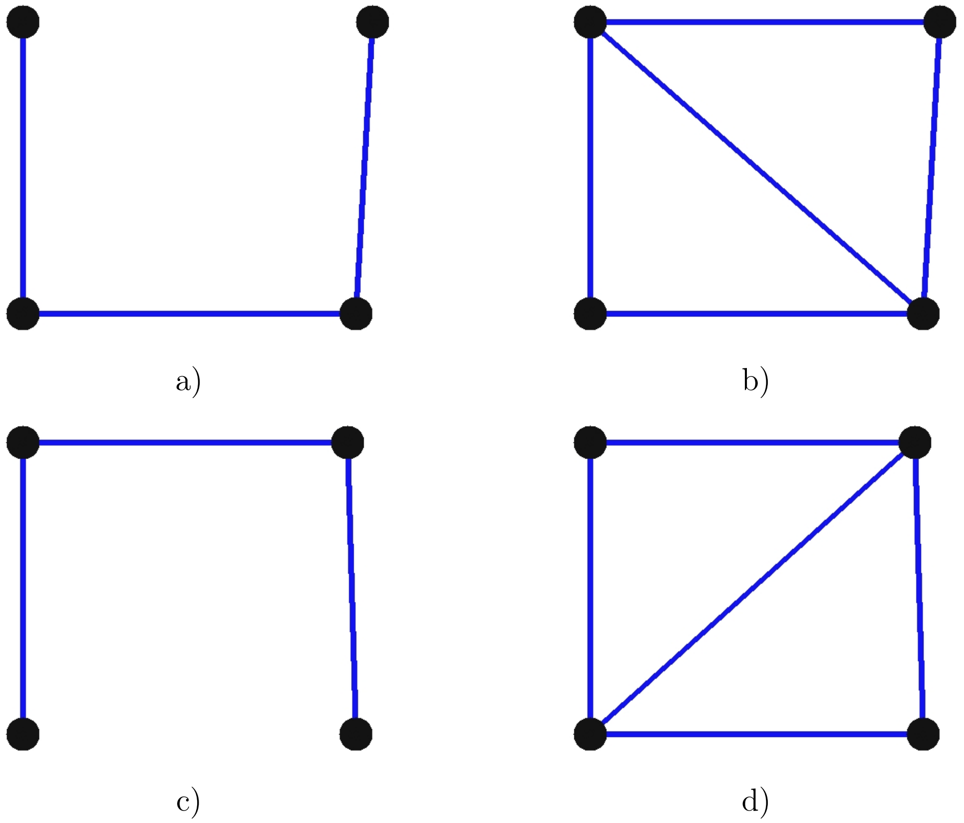


Figure 6.4: Topological model - stability of EMST vs. Delaunay triangulation in case of 4 points arranged as a nearly perfect square: (a),(b) EMST with bottom-right point slightly moved towards the inside and its corresponding Delaunay triangulation; (c),(d) EMST and Delaunay triangulation with the bottom-right point slightly moved towards the outside. Note the superior stability of the Delaunay triangulation, where the majority of the edges remain unchanged

the sinus of the angle between the two edge vectors. The total distance must not exceed a certain threshold ratio of the total sum of the edge weights of the model graph. The experiments described in the following section helped in determining a generally suitable range for the threshold ratio.

6.2.2 Experiments

Testing was carried out on a collection of 17 572 images from 1141 newspaper issues, scanned at 200 or 300 dpi. The document scans have been provided by two major German-language publishers. Five different front page models were trained and subsequently the front page detection algorithm was run on the entire document collection. We have tested two different threshold ratios for graph matching, as it was not clear what value would constitute a well-performing generic threshold.

In table 6.1, we have included not only the recognition rate, but also the precision and the recall, because the disparity between the number of front pages and regular pages make the recognition rate much less meaningful.

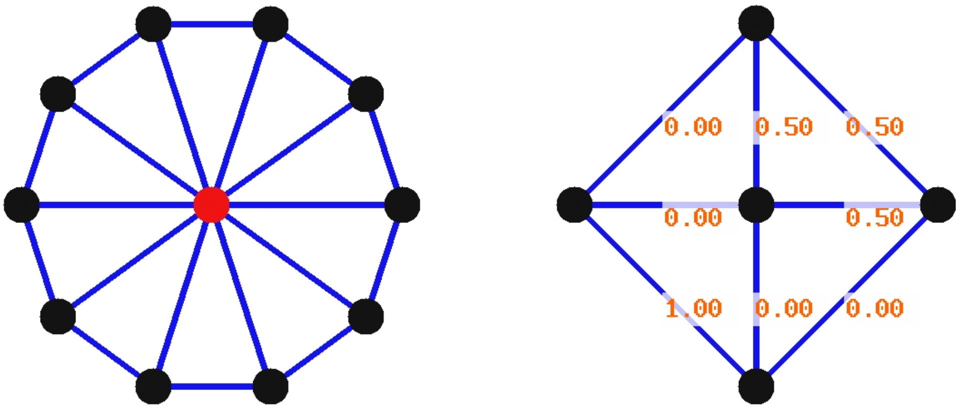


Figure 6.5: Left - The possibility of missing graph nodes (here: middle one) shows the necessity of having (near-) equal node weights (i.e. equal sums of incident edges); Right - Edges may also be missing in the candidate graph, thus an additional ideal requirement is to have equal edge weights

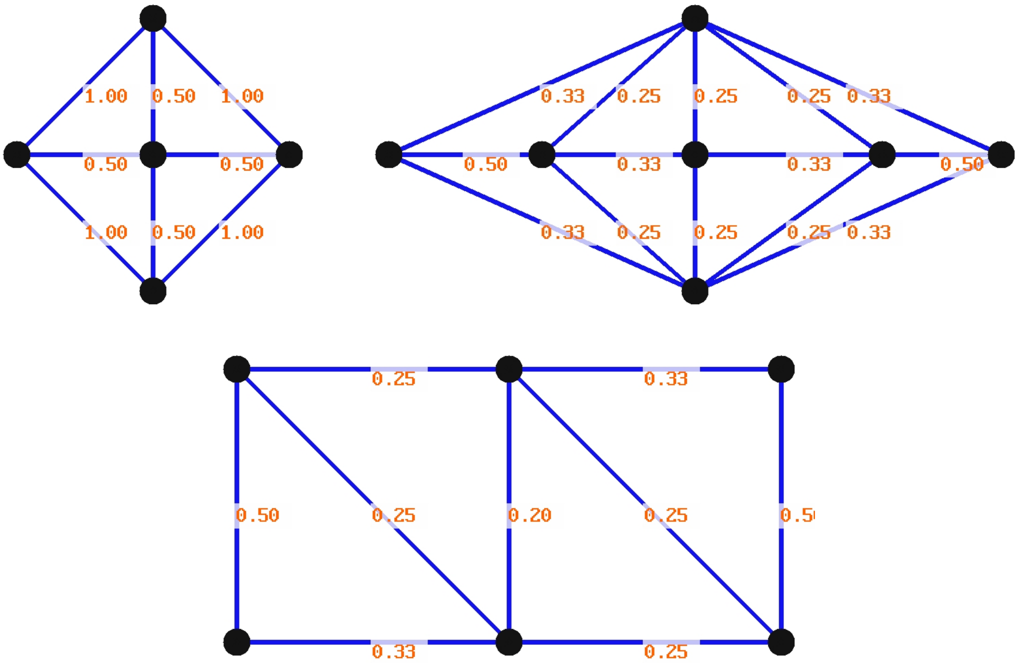


Figure 6.6: Sample graphs with edge weights computed using formula (6.1). Note the small edge weight differences and the nearly constant sums of incident edge weights for each node

From the obtained results, one can see that a 70% graph matching threshold θ is in general the better choice. It is important to notice that the 50% threshold, experimentally determined as being too high, generally implies that significantly more than half of the model graph's edges are present in a candidate graph, due to the additional penalties applied for all imperfect vector matches. The overall results obtained are encouraging and show that the described method can be applied with good results in digitization projects on a mass scale.

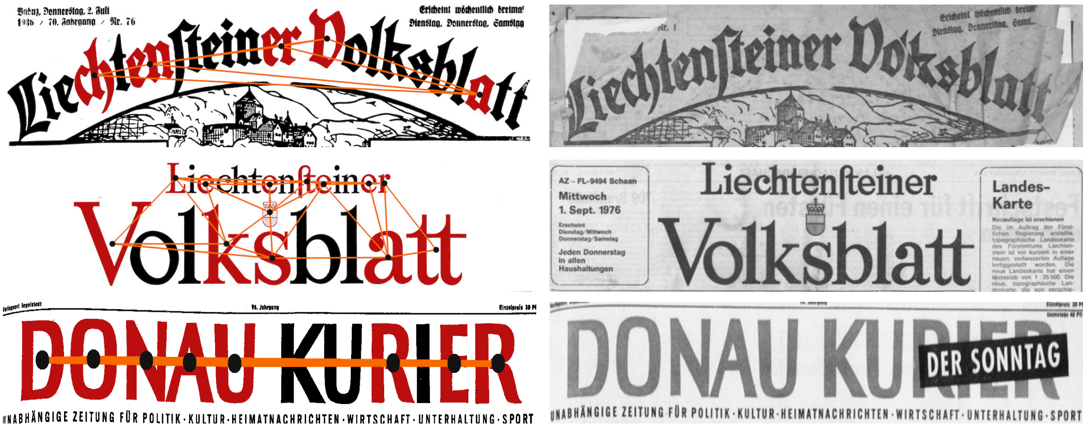

Liechtenſteiner Volksblatt
Liechtenſteiner
Volksblatt
Landes-
Karte
DONAU KURIER
DER SONNTAG
UNABHÄNGIGE ZEITUNG FÜR POLITIK · KULTUR · HEIMATNACHRICHTEN · WIRTSCHAFT · UNTERHALTUNG · SPORT

Figure 6.7: Three front page graph models (left), each along with a correctly recognized candidate, illustrating (top-to-bottom): resilence to limited distortions, scaling, partial occlusion

Model	θ	Pages	Issues	Recog. rate [%]	Recall [%]	Precision [%]
LV 1930	50%	1516	264	99.27	95.83	100
	70%			99.93	99.62	100
LV 1970	50%	7719	589	99.97	99.66	100
	70%			99.99	99.83	100
LV 1986	50%	3991	293	100	100	100
	70%			100	100	100
DK 1940	50%	2955	74	99.86	94.59	100
	70%			100	100	100
DK 1960	50%	1391	32	99.35	71.87	100
	70%			100	100	100

Table 6.1: Front page detection results obtained on a collection of 17 572 real-world newspaper scans. Note that the near-perfect precision and recall is actually a must-have for an automatic DIU system, as errors at this point would result in blatant inaccuracies to any human observer or post-processing step (i.e. entire missing/extra issues)

6.3 Complete Article Segmentation for Periodicals

A step up in difficulty from the front page detection problem, the current section introduces the logical layout analysis module of the Fraunhofer DIU system. The module is capable of performing a complete logical layout analysis for periodicals featuring complex layouts. More specifically, we present algorithms for extracting the exact layout columns, detecting framed boxes, constructing logically coherent regions, labeling them (e.g. body text, titles, captions, bylines, etc.), detecting a permissible reading order and segmenting a document page into articles. All algorithms described in the current section have been extensively used as part of the Fraunhofer DIU system in the processing of large-scale (i.e. > 500 000 pages) periodical collections. Note that many of the presented algorithms can be directly applied to books as well. No formal evaluation of the article segmentation results was performed on the complete document collections, because of the prohibitively large resource cost necessary for a manual evaluation. However, we present the evaluation results obtained on several subsets of the processed data, including both newspaper- and chronicle scans.

6.3.1 Methodology

In case of *complex layout documents*, the logical layout analysis phase must be able to cope with multiple columns and embedded commercial advertisements having a non-Manhattan layout. Most importantly however, the approach has to be flexible enough so as to be readily adaptable (or ideally adapt automatically) to the ever-changing layouts of each publisher.

The current section describes the general framework of the Fraunhofer DIU layout analysis modules, integrating a number of different algorithms. We require as input the complete page segmentation results, namely the list of all physical regions and their respective type. We do not require a fine-grained labeling of non-text regions, as our main focus lies on the textual regions and the separators. From the rough textual regions which represent the output of page segmentation we extract the text lines and compute a set of textual and logical attributes for each of them. By defining a generic dissimilarity function between text lines we can compute via dynamic programming the set of coherent logical regions which minimizes the overall merging costs. The obtained coherent text regions undergo a preliminary labeling into potential titles and captions by using a publisher-specific rule set. A compound distance value, computed using the similarity of the textual, logical and labeling attributes, along with the normalized physical distance and information about separators located in between is computed for each pair of coherent text regions. The minimum spanning tree (MST) is constructed from the set of computed edges and used to produce a complete article segmentation of the page. The most likely reading order is detected and used to merge matching article parts located on adjacent layout columns. The last step consists of assigning to each region its final logical type label by making use of all previously computed features for itself and all other text regions located within the same article.

6.3.1.1 Logical Layout Analysis Feature Extraction

As we have seen from the framework discussed in the previous section, a significant number of features must be computed as prerequisites for the different logical layout analysis steps. This section presents the methodology for computing all non-trivial features required by our algorithm. Note that not all features presented here are or even can be computed at the same time, however the location in the overall algorithm where they are employed should be clear from its description.

By making use of the page segmentation results, we are able to refine the estimation for the *dominant font* characteristics. In order to accomplish this we take into account only the connected components located within the input text regions and apply the character size detection algorithm described in section 3.1.3.1. Using the assumption that the dominant font of a document features (nearly) always non-slanted characters having a regular stroke width, we can now reliably label italic- and boldface text lines as such. In order to extract a better approximation for text lines, one can use any algorithm on the connected components from the text regions. We employ again the enhanced GTL approach described in section 5.2, but for example the constrained line finding algorithm of Breuel [59] may be used as well.

Computing the *stroke width* for a given text character is important for determining if it was emphasized as *boldface*. A generic feature, it is useful during the logical layout analysis of any kind of document, in order to detect titles, abstracts, captions, etc.. A simple procedure to compute the stroke width of a character as the width of the mean horizontal black run-length forming the respective character was proposed in [89]. Such an approximation is error-prone and generally not accurate enough for computing this feature. A slower, but much more exact and reliable alternative, is the computation of the *Euclidean distance transform*. Given an input binary image, the Euclidean distance transform will produce as result an image of the same size as the original, where the value at any location represents the shortest distance to a background pixel. In such a way, the stroke width of the considered character can simply be taken to be the maximum distance value within the transformed representation of the character. Note that the Euclidean distance transform and its approximations are closely related to the extraction of the character skeleton, a feature widely used in OCR systems. The Fraunhofer DIU system uses a probability value instead of a boolean in order to represent the "boldface" attribute in order to better cope with printing errors – e.g. older printing machines had problems controlling exactly the ink amount used for each glyph, erroneously resulting in visibly different stroke widths. The probability is computed with respect to the ratio of the x-height to the stroke width in the dominant font, which is considered to be equal to 0. More specifically, we consider a linear scale with the maximum probability clipped at the value $\frac{XHeight_{domfont}}{StrokeWidth_{domfont}} \times 0.65$, where the constant 0.65 was experimentally determined and performs well on a large range of documents.

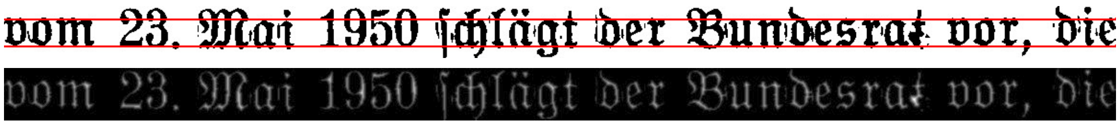

vom 23. Mai 1950 ſchlägt der Bundesrat vor, die
vom 23. Mai 1950 ſchlägt der Bundesrat vor, die

Figure 6.8: Top: text line showing the detected baseline and character x-height; Bottom: result of applying the Euclidean distance transform on the same text line for the purpose of determining the stroke width for each character (white: high distance values)

In general, Euclidean distance transform algorithms belong to one of two possible categories: approximate algorithms and exact algorithms. Approximate algorithms are generally much faster, but do not guarantee exact results, especially in case of wide, irregular filled surfaces. As part of the proposed system, two different Euclidean distance transform algorithms have been implemented: an approximate algorithm proposed by Danielsson [84] and an exact algorithm proposed by Maurer et al. [181]. The algorithm of Danielsson is based on vector propagation, i.e. while scanning over the image, the coordinates of the nearest background pixel are propagated between neighboring object pixels. For relatively thin components (stroke width less than 20 pixels) it provides exact results, and the error margin grows with the thickness of the considered component. The algorithm of Maurer et al. is based on partial Voronoi diagram construction, and by using a dimensionality reduction technique, recursively reduces the N-dimensional problem to an easily solvable 1-D problem. Both algorithms have a worst-case complexity of $O(N)$, but the hidden constant in the Maurer et al. method is much higher than in case of the approximate algorithm. This thesis does not contain a more detailed description of (Euclidean) distance transform algorithms, but a good reference for the interested reader is the PhD thesis of O. Cuisenaire [82], which deals with this issue in great detail. For practical purposes, using the algorithm of Danielsson for determining the stroke width of text characters represents a good compromise between speed and accuracy (see exact error analysis of Cuisenaire [82]).

In order to compute the *italic* property for a text line, we employ a computationally-efficient hierarchical approach. We consider at start a possible skew angle range of the main character axes of $R = [0°, 45°]$ and we set a target precision of $0.5°$. We logically divide the interval into N bins of equal width and compute the alignment premium score [217] for each bin K on the text line horizontally sheared by $\frac{|R|}{N}(K + 0.5)$ degrees. We set as constant row for the shear operations the vertical center of the robust baseline fit for the characters belonging to the text line. Postl defines the alignment premium score as the sum of squared differences between the adjacent bin values from the vertical projection of an image. Note that Postl initially proposed this measure for computing the global document skew. Next, we find the bin with the maximum alignment premium and apply the search process recursively on the respective interval. The process is repeated until the interval size falls below the pre-specified target precision. The value obtained as a result of this procedure represents the dominant skew angle of the main axes for the characters on the text line. We experimentally determined that a generic, fixed threshold of 4.5 degrees can safely be used for computing the boolean "italic" property.

Other important features describing a text line are the *x-height* and its *capital letter height*. As input data for computing both values we require the heights of the portions of all likely characters located above the *baseline*. As likely characters we consider those deviating less than 2 times from the median connected component height in a text line. We apply a k-means clustering [129] with $k = 2$ on the set of heights and consider the resulting lower height to correspond to the x-height, and the higher height to the capital letter height. Note that if the two obtained cluster centers are located very close or are identical, we are most likely dealing with a text line containing just small or large characters/ digits. As alternative one may also detect the two height values as the two tallest peaks from the smoothed histogram of the input set of heights (and considering a minimum height difference as search condition).

The features presented until now all have a highly local character – i.e. they can be computed even for small groups of characters. By contrast, we must now compute a few higher-level characteristics. One of these features is the set of closed (framed) regions. Most commonly, such regions represent coherent logical units (e.g. single articles, advertisements, tables, quotations, etc.) within a document. Therefore, *frame boxes* give us important information about the logical connections between page elements, for both text and non-text areas alike. The detection algorithms requires as input the lists of extracted vertical and horizontal separators and produces as output a list of all detected rectangular frames. We define a frame as a quadruple of separators, two vertical ones and two horizontal ones, forming a complete or nearly complete rectangle. Note that it is entirely possible to also detect frame boxes where one or more separators are broken. One can achieve this by constructing beforehand the set of all possible "virtual" separators formed by merging two or more separators of the same type located directly next to one another, where in between there exists no separator of the opposite type. Instead of just the initial set of separators one may also consider the constructed virtual separators in the following search procedure. The underlying idea of the search function is simple, yet effective: for each horizontal separator, the algorithm looks for a pair of vertical separators with approximately the same height, one located on its left side, the other on its right side. Finally, a suitable horizontal separator for closing the box at its top/bottom is searched for, and the frame is validated if such a separator is found. Only two parameters are necessary – the relative and absolute tolerance for the allowable (Euclidean) distance between the adjacent pairs of separators in the framed boxes. The most challenging part of the proposed algorithm is the search for the best matching vertical lines to the left/right of a given horizontal separator.

An exhaustive search through all possible pairs of vertical separators and closing horizontal separators, executed for each horizontal separator would result in a very slow execution speed, especially for large newspaper pages with a large number of separators (e.g. tables, advertisements). Therefore, we make use of spatial data structures in the form of binary space partitioning (BSP) trees.

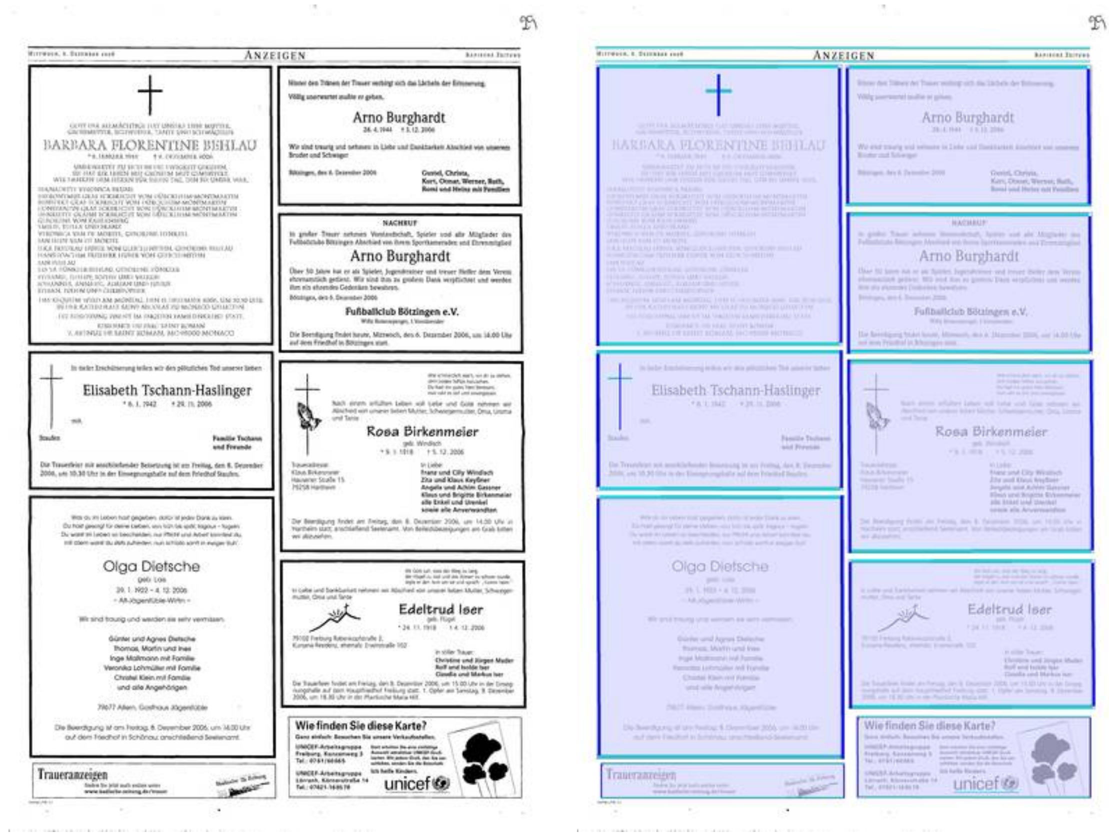


Figure 6.9: Left: original newspaper image; right: rectangular frames (darkened background) detected by the proposed algorithm from the sets of horizontal- (cyan) and vertical solid separators (blue)

Another powerful global document feature is given by the positions, widths and heights of the *layout columns*. For multi-column documents, such as journals and newspapers, knowledge about the layout columns is crucial for computing the logical labels of text regions, the extraction of tables and in the segmentation of articles. Furthermore, in the case of newspapers it is possible to automatically differentiate between advertisement pages and pages with editorial content solely based on the structure of the detected layout columns. The proposed layout column detection algorithm requires as input the sets of vertical and horizontal separators, as well as the bounding boxes of the identified text lines. As output, the algorithm produces a set of axis-aligned rectangular regions having the same height, corresponding to the layout columns present in the given page. In case no valid columns are detected, the algorithm will return the smallest axis-aligned rectangle containing all text lines – usually equivalent to the page's print area. As the first step, a binned histogram is created from the widths of the given text lines, using as bin radius the width of the average character from the dominant font. Thereafter, the histogram is traversed in descending order all peaks higher than a fixed threshold (default: 4) and taller than half the highest peak are saved as possible column widths. For each of the candidate column widths, a search procedure is called for determining if the respective average layout column width can produce a valid covering of the document page. The algorithm stops when the column coverage search procedure finds more than one column, or when all candidate column width have been investigated. The column coverage search procedure employs dynamic programming in the attempt to find the longest sequence of vertically overlapped vertical separators having between each consecutive pair a horizontal distance of approximately the specified column width. Experimentally we have determined that a relative tolerance for the column width difference of around 20% produces good results. Note that since in general we do not know if two vertical separators (or white spaces) do indeed exist at the left, respectively right side of the image, we initially add two such dummy separators to the original separator set. This is necessary for the search procedure, as the two separators act as beginning and end of page delimiters. In case the search function finds a sequence of less than 3 separators in length, it is considered to have been unsuccessful. Otherwise, two additional conditions must be satisfied in order to accept the corresponding column layout as valid:

- The layout columns span more than 20% of the print area width
- In case the layout column span less than 60% of the print area, they must contain most of the text lines on the page

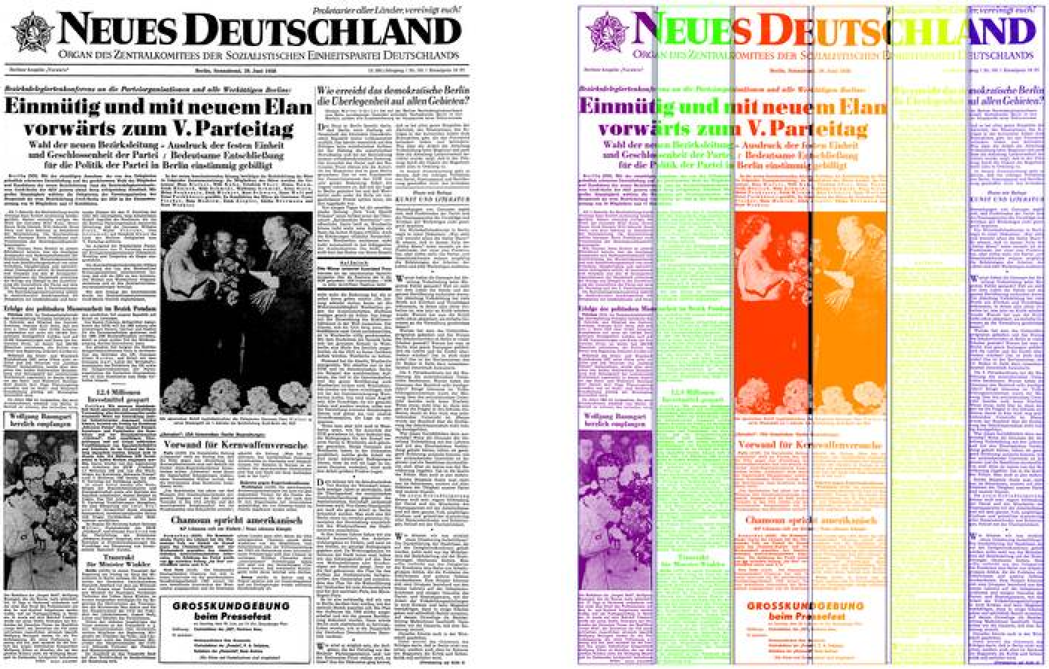

Proletarier aller Länder, vereinigt euch!
NEUES DEUTSCHLAND
ORGAN DES ZENTRALKOMITEES DER SOZIALISTISCHEN EINHEITSPARTEI DEUTSCHLANDS
Einmütig und mit neuem Elan vorwärts zum V. Parteitag
Wie erreicht das demokratische Berlin die Überlegenheit auf allen Gebieten?
Chamoun spricht amerikanisch
GROSSKUNDGEBUNG beim Pressefest

Figure 6.10: Left: newspaper image; right: determined layout columns for the editorial content

The two aforementioned conditions are derived from typical editorial layout rules regarding the size and placement of advertisements. They are well-suited for classical newspapers and journals, but may need to be adjusted for publications featuring a more free layout style. If the conditions are satisfied, the bounding boxes corresponding to each text column between two consecutive separators are created with a height equal to the print area height. The width of each text column is determined by taking the minimum and maximum x coordinate of the leftmost, respectively rightmost text line located between the two considered separators. If any solid horizontal separator intersects all created columns, we split each column vertically and keep as final result only the part containing the highest number of text lines. One must note that such situations occur frequently on pages containing both editorial text and advertisements. Finally, we compute an exact vertical line fit for the each layout column's left and right borders by computing a robust line fit for the start and endpoints of all contained text lines. The exact fits are crucial for the computation of accurate *indentation attributes* (i.e. centered, left-aligned, right-aligned) for each text line and region.

The interested reader may find a more detailed description of other features relevant for logical layout analysis in the recent work of Gao et al. [163].

6.3.1.2 Logical Region Building and Reading Order Detection

At this point we are in possession of the complete set of text lines on the document page, as well as the set of all (solid and white) separators. The previous section describes the exact modality of computing global document features, such as layout columns and framed boxes, as well as local features specific to each text line: baseline, x-height, capital height, boldface and italic indicators, alignment attributes, the set of intersected layout columns.

We can now use the aforementioned set of features for the text lines in order to define a *distance measure* between any pair of text lines. In effect, any distance measure can be used in the following as long as it is directly proportional to the distances between each corresponding pair of feature values. While indeed tempting, one must note that it is not directly possible to consider the sets of features as vectors and compute their dot product. This is because of multiple issues which may arise. One of the issues involves the font size distance (given by the pair x-height, capital height), which must account for lines featuring only capitals/small characters versus ones where both values could be determined. Another key point is the special treatment necessary for paragraph starts, i.e. indented text lines following non-indented ones must reflect in a higher distance than that used in the case of non-indented lines following indented ones.

The set of *coherent text regions* is constructed via dynamic programming internally using the defined distance measure. The optimization goal for the dynamic programming function is the minimization of the overall distance measure between the adjacent text lines, as well as the minimization of the intra-region variation of the inter-line distances. In addition we do not allow adjacent text lines be merged into the same region if between them there exists any separator. The modality for solving this minimization problem is very similar to that of chain matrix multiplication using a minimum number of multiplications. We shall not go into further detail, as many programming books exist which deal with the subject exhaustively. What is important to observe is that the applicability of a dynamic programming approach has as prerequisite a previous sorting of the text lines, hence the notion of "adjacency". In our case, the sorting of text lines is given by their reading order. Note that the creation of text regions defined in this manner is completely generic and results in an optimal partitioning of the page.

What remains to be discussed is the *detection of the reading order* between text regions. Indeed, this is not a trivial problem and may depend not only on the geometric layout of a document (which varies widely among publishers even for the same document type), but also on linguistic and semantic content. It is entirely possible that many allowable reading orders exists for a given sequence of text blocks. We are only interested in extracting one of these. For the reading order detection task we consider that each text line is approximated by its bounding box. This is an important simplification, as one may be able to use OCR results and linguistic modeling for obtaining much more accurate results (i.e. reducing the cardinality of the set of allowable reading orders). As general framework for reading order detection, we employ the method proposed by Breuel [61], enriched with information regarding the layout columns. In short, we compute a partial ordering between each pair of text lines and apply a topological sorting for determining an overall admissible ordering. Liang et al. [165] observe common failure causes for the algorithm of Breuel – they mention that Breuel's algorithm fails most times due to the unsuitability of their precedence relationships on multi-column layouts. We are able to remedy this problem by making use of the computed exact layout columns. The resulting algorithm can easily be adapted to different scripts (e.g. right-to-left, bottom-to-top). Interested readers can find a reading order detection algorithm using a different approach in the work of Aiello et al. [22].

6.3.1.3 Article Segmentation and Region Labeling

Beside the text regions and the set of separators on the document page, we require a preliminary labeling of the text regions into possible titles and captions. The labeling procedure in the Fraunhofer DIU system uses a rule-based approach, with generic as well as publisher-specific rules. Such hand-crafted rules are necessary due to the large style variations among publications, publishing houses and time epochs. In effect, this entails that for obtaining optimum performance the logical region labeling procedure must be specialized by the addition of publisher-specific rules.

As seen from the layout analysis module description at the start of the current section 6.3.1, the construction of a MST constitutes the key part of the Fraunhofer layout analysis module. We use a MST approach for generating an initial (over-)segmentation of a document page into articles. There exist previous approaches in document layout analysis relying on the MST [89, 140]. Ittner and Baird [140] construct the MST from the centers of the connected components in the document image, and by means of a histogram of slopes of the tree edges, they are able to detect the dominant orientation of the text lines. More interesting in our case is the algorithm of Dias [89] which uses the MST to compute the set of coherent text regions in a document. Dias constructs the MST from individual connected components and, using automatically determined inter-character (horizontal) and inter-line (vertical) spacing as splitting thresholds for the tree edges, produces as output a segmentation of the document page into text regions. The most important problems observed by Dias are the sensitivity of the MST to noise components and the fact that a single incorrectly split branch has the potential to produce a poor segmentation.

At this point, we are able to compute a *compound distance measure* between any two text regions. The distance measure is defined as a weighted mean of the Euclidean distance between their bounding boxes and a value directly reflecting the "logical distance" between the two text blocks. The logical distance between two text blocks is asymmetrical and directly influenced by the number and type of separators present between the two text blocks, as well as by their feature similarity (as defined for text region creation) and title/caption/regular labeling. The weights assigned to each of the components forming the logical distance can and must generally be adjusted so as to reflect the chosen layout style of the considered publisher. Note that the publisher-specific influence is actually much less important in this case than for the title/caption labeling. It is only relevant

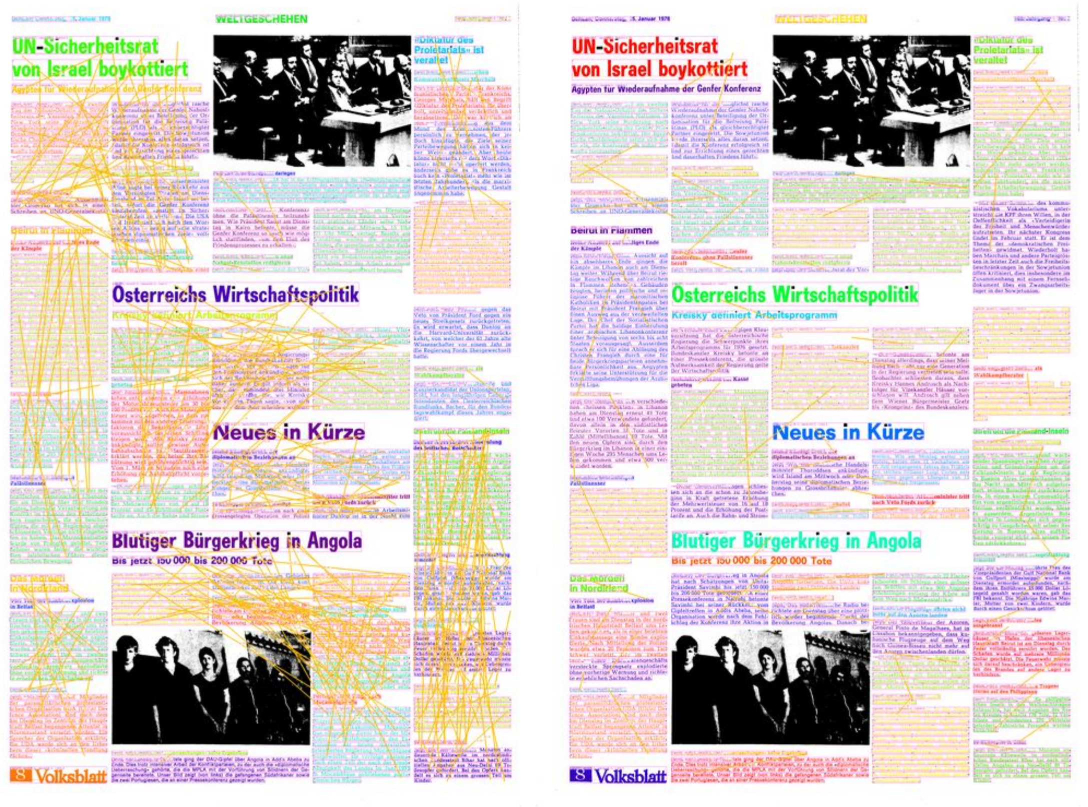

WELTGESCHEHEN
UN-Sicherheitsrat von Israel boykottiert
Ägypten für Wiederaufnahme der Genfer Konferenz
»Diktatur des Proletariats« ist veraltet
Beirut in Flammen
Österreichs Wirtschaftspolitik
Kreisky definiert Arbeitsprogramm
Neues in Kürze
Blutiger Bürgerkrieg in Angola
Bis jetzt 150 000 bis 200 000 Tote
Das Morden in Nordirland
Volksblatt
WELTGESCHEHEN
UN-Sicherheitsrat von Israel boykottiert
Ägypten für Wiederaufnahme der Genfer Konferenz
»Diktatur des Proletariats« ist veraltet
Beirut in Flammen
Österreichs Wirtschaftspolitik
Kreisky definiert Arbeitsprogramm
Neues in Kürze
Blutiger Bürgerkrieg in Angola
Bis jetzt 150 000 bis 200 000 Tote
Das Morden in Nordirland
Volksblatt

Figure 6.11: Sample visualization of the output of the proposed article segmentation algorithm on a newspaper image: left – initial graph edges; right – logical layout analysis result

for publishers making use of special editorial layouts, such as articles within articles or sections featuring specific layouts. A few of the generic rules for computing the compound distance are presented in the following. For example a regular text block located before a title block in reading order will have a high logical distance to it (a probable article ending is located between them). A provisional *hierarchy of titles* proved beneficial to use in our tests, as it allows the formulation of rules such as: a lower-level title located (in reading order) after a higher-level title with no other title in between has a low logical distance to it (as they are very likely part of the same article). Also, for logical distance computation purposes, the existence of (inter-column) separators between two regions results in a high penalty factor.

By using the compound distance measure between text regions, the MST of the document page can be constructed as the next step of the algorithm as a starting point for generating an *article (over-)segmentation*. Keeping in mind the noise sensitivity remarked by Dias [89] upon the construction of the MST, we can see that it is indeed greatly reduced in our case. This happens because we use higher-level page components, i.e. logical blocks instead of just connected components. Next, the obtained tree is split into a forest, each resulting tree ideally corresponding to an article (or a part of one). The splitting operation can be accomplished in a straightforward manner by "cutting" the edges with weights greater than a certain threshold value. The considered threshold value is closely related to the logical distance between blocks.

The final stage of the algorithm consists of the *merging of broken article parts* and the *complete labeling* of all regions within the articles. In order to merge the over-segmented set of articles we again make use of the reading order algorithm presented in the previous section. This time, we sort the articles in reading order by considering each article represented by the minimum area bounding box enclosing its regions. Next, we simply merge all adjacent articles where the set of detected titles (hierarchical order and textual properties) are compatible. We consider as additional restriction the presence of a solid horizontal separator between two articles. A factor allowing the merging of otherwise non-adjacent (but with compatible title hierarchies) articles is their belonging to the same frame box (see section 6.3.1.1). Vertical separators are intentionally disregarded at this point, since they were already taken into account as part of the layout column detection. Once the final article set is obtained, the complete set of logical labels may be assigned to the text and non-text regions. As for the previous labeling step, for optimum performance in case of large layout differences one must resort to publisher-specific sets of rules in combination with generic rules (the same ones as necessary for logical distance computation). Non-text regions are assigned to the articles by minimizing the Manhattan distance (main criterion) and maximizing overlap (as secondary criterion).

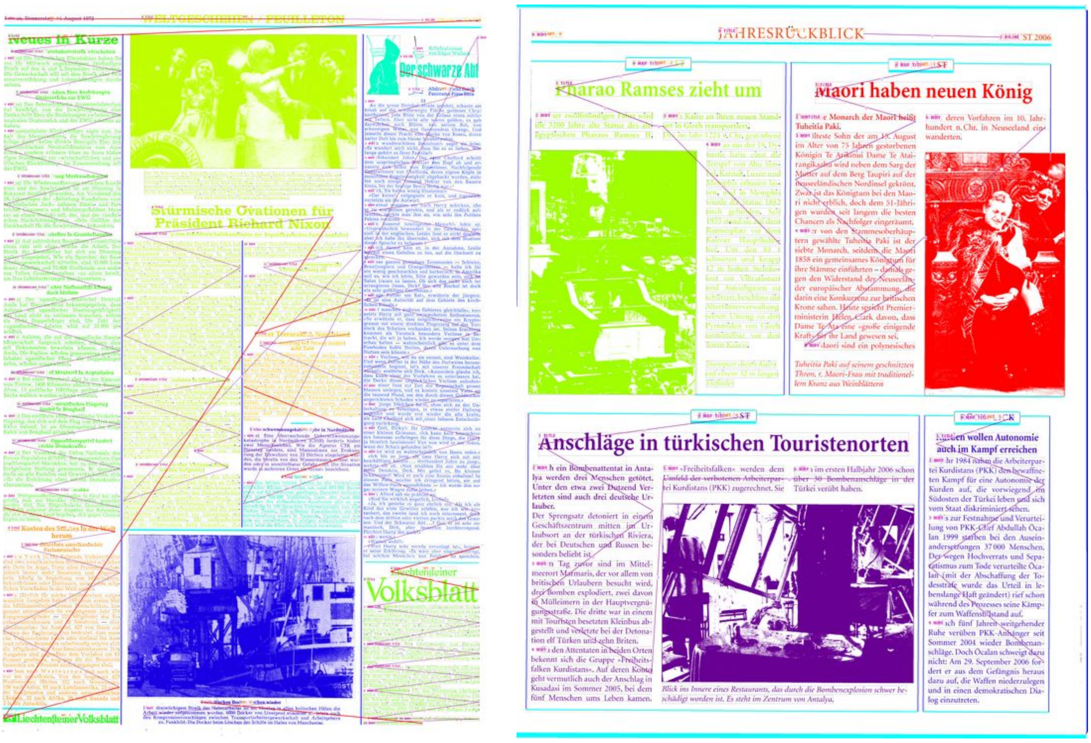

Neues in Kürze
Stürmische Ovationen für Präsident Richard Nixon
Der schwarze Abt
Volksblatt
Liechtensteiner Volksblatt
JAHRESRÜCKBLICK
Pharao Ramses zieht um
Maori haben neuen König
Anschläge in türkischen Touristenorten

Figure 6.12: Visualizations of article segmentation results on scanned images from a newspaper (left side) and a chronicle (right side). Articles are drawn in distinct colors and the line segments indicate the detected reading order

The layout analysis algorithm described in this section has the advantage of being light on computational power, robust to noise and easily adaptable to a wide variety of document layouts. Its main shortcoming however, is the need for manual adaptation of the logical distance measure and logical type labeling rules for each publisher or layout type in order to obtain optimal results. Also, the current algorithm does not need to take into account the text within each block, which is a useful property in cases where only certain parts of the document page are of interest. The application of machine learning algorithms (such as those described in section 6.4) for automating these tasks represents a most promising direction.

6.3.2 Experiments

In the current section we present the experimental results gathered as a result of human evaluation. The total size of the test sets was very close to 250 pages, coming from newspapers and chronicles published during the last 70 years. An important general observation is that all results for the logical labeling process as well as the article segmentations are highly dependent on the input page segmentation. The homogeneous physical regions wrongly labeled by the page segmentation module will inevitably cause errors for all subsequent DIU steps, as they cannot be re-labeled in the further steps. Consequently, all results presented in the following reflect *error sources from both the page segmentation and the logical layout analysis*. Note that even if one would know the exact errors made by the page segmentation in each case, the evaluation of the layout analysis results in isolation would still be inaccurate because of the interdependencies between logical regions. For example, a title region which was mislabeled as non-text by the page segmentation may cause its subtitle to be wrongly labeled as main title in the logical layout analysis step. Although we know that the nonexistence/mislabeling of the real title region is caused by the page segmentation, we cannot know whether in case the mislabeling would not have occurred that the title and subtitle would have both been correctly labeled as such or not.

Year	**Total articles**	**Complete articles**	**Articles with all titles labeled**	**Articles with all author(s) labeled**	**Flawless articles**
1950	10	6	7	5	4
1960	23	16	17	9	8
1970	128	88	79	69	43
1980	5	5	2	3	2
Total	166	115	105	86	57
Percent		**69,3%**	**63,3%**	**51,8%**	**34,3%**

Table 6.2: Article segmentation results on a random subset of scans from the German language newspaper *"Die Zeit"*. Note: "Titles" also include roof titles, subtitles and intermediary titles

The first two data sets contain multi-column, grayscale newspaper images from the German language newspapers "Die Zeit" and "Neues Deutschland", respectively. The "Die Zeit" set consisted of around 20 images containing 166 articles, while the "Neues Deutschland" image set contained around 132 pages and 1731 articles. The scanning resolution for the newspaper images varies between 300 and 400 dpi. The reader may see a typical layout example in figure 6.12. While the sample size is significantly lower for the newspaper "Die Zeit", we can see that the overall evaluation results do not differ considerably. This fact is significant because shows that the layout analysis algorithms processed are indeed able to generalize well to different publishers. The most significant difference ($\approx 15\%$) is visible in the percentage of complete articles. We have investigated this discrepancy and found out that it is caused by the more complex layout of the "Neues Deutschland" newspaper. The manually coded rule sets were unable to adapt to all its peculiarities, thus causing relatively many non-text regions assigned to the wrong articles as well as merging errors caused by the consideration of only a single possible reading order. Another interesting observation is that the total percentage of flawlessly segmented and logically labeled articles is relatively low in both cases (close to 35%). This adequately reflects the difficulty of the task and the considerable potential for improvement in the area of layout analysis. Note that for the second dataset we added a category for "good" articles, which denotes articles perfectly segmented and having all title types correctly labeled. This represents a better approximation of a baseline for real-life performance, since for obtaining a more accurate labeling/metadata extraction into an extended number of categories (e.g. authors, locations, bylines, leads, etc.) one may wish use other more powerful classification algorithms and features (see research area of named entity recognition).

Issue date	**Total articles**	**Complete articles**	**Articles with all titles labeled**	**Articles with all author(s) labeled**	**Flawless articles**	**Good articles**
Jun.1949	104	44	54	56	29	40
Oct.1952	61	33	29	39	17	26
Feb.1956	97	49	41	57	26	46
Nov.1964	99	51	64	64	45	48
Sep.1967	123	38	67	51	24	34
Jun.1970	119	59	85	77	42	53
Apr.1973	114	71	84	85	56	67
Apr.1976	121	88	110	96	75	86
Jul.1979	96	49	66	79	44	49
Oct.1982	98	66	80	78	61	63
Jan.1986	148	89	93	99	72	82
Apr.1989	143	85	92	93	64	80
Aug.1992	159	71	98	78	46	65
May 1993	249	143	170	123	69	117
Total	1731	936	1133	1075	670	856
Percent		**54,1%**	**65,5%**	**62,1%**	**38,7%**	**49,5%**

Table 6.3: Article segmentation results on a random subset of scans covering 14 issues of the German language newspaper *"Neues Deutschland"*, uniformly distributed throughout the years 1949–1993 and totaling 132 digitized pages. Note: the term "titles" actually denotes 4 different types of titles (regular-, roof-, sub- and intermediary titles), while "good" articles were considered to be those complete and with perfectly labeled titles, irrespective of the author labeling success

The third dataset used in our evaluation consists of 100 multi-column chronicle images printed in 2006. An example of the simple chronicle layout can be seen in figure 6.12. In contrast to the newspaper scans, the chronicle images originally featured a 24-bit color depth and a high scanning resolution of 600 dpi. In the test set there were 621 titles (incl. subtitles, roof titles and intermediary titles), and for the detection and labeling task the manual rule set achieved a precision of 86% and a recall of 96% (resulting in an F-measure of 90.7%). For the detection of captions on the test set containing 255 instances, the rule set was able to achieve an F-measure of 97.6% (precision 98.4% and recall 96.8%). These values are only significant to show that a relatively simple labeling rule set is able to perform quite well on known layouts, thus giving hope that such rule sets can be evolved in the future automatically through machine learning methods. Based on the results produced by these two manual rule sets, the article segmentation algorithm was able to correctly segment 85.2% of the 311 total articles present in the test set. While the vast majority of document images are segmented correctly, a few pages fail catastrophically, thus generating most of the observed errors (e.g. two pages were responsible for more than 75% of the title labeling errors). Article split errors were the most common, totaling 13.2% and most often these were generated as a direct consequence of errors during color reduction and/or page segmentation (i.e. split non-text regions, such as tables and bright halftone images). The large discrepancy between the success rate at article level between the newspaper- and the chronicle datasets is partially explained by the simpler page layout in the latter case. Most significantly, the majority of articles were printed within individual frame boxes, which we are able to detect reliably (near 100% rate in case of non-broken forming separators).

	# Instances	**# Correct**	**Precision**	**Recall**	**F-Measure**
Titles	621	596	86.0%	95.9%	90.7%
Captions	255	247	98.4%	96.8%	97.6%

	# Instances	**# Complete**	**Splits**	**Merges**	**Complete ratio**
Articles	311	265	41	6	85.2%

Table 6.4: Article segmentation results on a 100-page subset of scans from the annual chronicle 2006 (Jahreschronik). Note: the term “titles” actually denotes 4 different types of titles (regular-, roof-, sub- and intermediary titles)

The average test image size was around 4000 × 6000 pixels. In these conditions, the total processing time for the logical layout analysis (incl. feature computation, region creation, article segmentation and region labeling) on one document image was about $8 - 10$ seconds on a computer equipped with an Intel Core2Duo 2.66GHz processor and 2GB RAM.

6.4 Logical Region Labeling via Conditional Random Fields

As we have seen from the previous section, logical labeling plays the key role in the overall logical layout analysis process. At the same time, it is exactly this module that relies most heavily on publisher- or time period-dependent manual rule sets. Thus, intuition tells us that by automating the logical labeling we can obtain the most gains for LLA as a whole.

Document logical layout analysis can generally use two sources of information. On the one hand, as exploited in the previous sections, the layout of the page regions in the document often gives many clues about the relation of different logical units like headings, body text, references, captions, etc.. On the other hand the wording and the text content itself can be exploited to recognize the interrelation and semantics of text regions. One must note that until this point we have completely ignored the latter information source. This was necessary in order to maintain acceptable computational costs at the current standards. In contrast, our goal in the current section is to make use of all available information so as to get closer to a truly universal and effective algorithm.

Deterministic models for LLA, such as the MST-based approach previously presented, often can only handle a limited amount of noise or ambiguities. This is problematic, since document pages almost always exhibit a certain degree of degradation, be it due to printing, handling, paper aging, digitization process or other factors (see section 2.2). It is inevitable that the outputs of the geometric structure analysis are not always correct and so the LLA process must generally be able to cope with uncertain inputs. While the influence of noise can indeed be greatly reduced by smart deterministic algorithms (as done in section 6.3.1), it is even better to allow for the existence of noise (regions) by design.

For this purpose, the use of machine learning techniques has been identified by researchers as a promising new direction to take [179]. In the current section we take one step in this direction by introducing a machine learning approach which captures the meaning of document parts via hidden state variables. Probabilistic models are employed in order to describe the relation between these state variables. The variables are thus allowed to have a rich set of interactions with some observable features. Conditional Random Fields (CRFs) are used to estimate the parameters of these models from training data. Consequently, the models have the inherent potential to automatically and dynamically adapt to novel structures and layouts.

As a final observation, our choice of a supervised machine learning algorithm was far from arbitrary. The choice was motivated by the fact that layout composition rules are somewhat arbitrary and can wildly differ among publishers and over time, as can also be observed from figure 6.1. In other words, we believe that in our situation "teaching" these hidden composition rules by example is a much more efficient and natural way of accomplishing the learning task as opposed to providing artificial rewards (i.e. semi-supervised learning)/ completely uninformed search for patterns (unsupervised learning). Throughout the last 10 years, CRFs have been thoroughly investigated and successfully applied in many research areas, such as computer vision [130, 157], text processing [212, 263] and bioinformatics [170, 227].

6.4.1 Basic Model

Let us consider the problem that we want to identify title and author in the following text snippet:

The new bestseller:
Game of Thrones
by George Martin

For simplicity we may write the words of the snippet including newlines and mark them with **T** if they belong to the title, by **A** if they belong to the authors, and by **O** otherwise. This gives the two vectors x of words and y of unknown states:

y	O	O	O	O	T	T	T	O	O	A	A
x	The	new	bestseller:	\n	**Game**	**of**	**Thrones**	\n	by	George	Martin

Text data in documents has two characteristics: first, statistical dependencies exist between the words we wish to model and second, each word often has a rich set of features that can aid classification. For example, when identifying titles in documents we can exploit the format and font properties of the title itself, but the location and properties of an author and an abstract in the neighborhood can improve performance.

To infer the unknown states we represent the relation between sequences y and x by a conditional probability distribution $p(y|x)$. More specifically let the variables $y = (y_1, \ldots, y_n)$ represent the labels of the word that we wish to predict with a set Y of possible values. Let the input variables $x = (x_1, \ldots, x_n)$ represent the observed words and their properties. If $I = \{1, \ldots, n\}$ is the set of indices of y then we denote the subvector corresponding to the indices in $A \subset I$ by y_A. Let $\phi_A(x, y_A) > 0$ be a *factor function* with x and the subvectors y_A as arguments and let $\mathcal{C}$ be a set of subsets of $A \subset I$. Each $\phi_A(x, y_A)$ is a function taking into account the relation between the labels in the subvector y_A, which often are the adjacent labels in the sequence. Then we represent the conditional distribution by a product of factor functions:

$$p(y|x) = \frac{1}{Z(x)} \prod_{A \in \mathcal{C}} \phi_A(x, y_A)$$

Here $Z(x) = \sum_y \prod_{A\in\mathcal{C}} \phi_A(x, y_A)$ is a factor normalizing the sum of probabilities to 1.

The product structure enforces a specific dependency structure of the variables y_i. Consider the conditional distribution of y_i given all other variables. It may be written as:

$$p(y_i|y_{D(i)}, x) = \frac{p(y_i, y_{D(i)}, x)}{\sum_{y_i\in Y} p(y_i, y_{D(i)}, x)} = \frac{\prod_{B\in\mathcal{C}, i\in B} \phi_B(x, y_B)}{\sum_{y_i\in Y} \prod_{B\in\mathcal{C}, i\in B} \phi_B(x, y_B)} \qquad (6.2)$$

as the factor functions $\phi_A(x, y_A)$ where $i \notin A$ cancel. Therefore the conditional probability of y_i is completely determined if the values of x and the y_B are known for all B which contain i. The factor functions $\phi_A(x, y_A)$ describe the *interactions* between the argument variables. Obviously C determines the dependency structure of the components of y. A probability distribution of this form is known as a conditional random field [160, 259]. As dependencies among the input variables x do not need to be explicitly represented, rich, global input features x may be used. For example, in natural language tasks, useful features include neighboring words and word N-grams, prefixes, suffixes, capitalization, membership in domain-specific lexicons, and semantic information from sources such as WordNet.

Usually there exist a number of different *features* for the same variables x, y_A. For $A = \{i\}$ for instance $\phi_A(x, y_i)$ may cover the feature that word x_i is in bold and $y_i = T$, i.e. is a title word. If we have K_A features for A then we may write $\phi_A(x, y_A) = \exp(\sum_{k=1}^{K_A} \lambda_{A,k} f_{A,k}(x, y_A))$. Here $\lambda_{A,k}$ is a real-valued *parameter* determining the importance of the real-valued *feature function* $f_{A,k}(x, y_A)$. The exponentiation ensures that the factor functions are positive. This yields the representation:

$$p(y|x) = \frac{1}{Z(x)} \prod_{A\in\mathcal{C}} \exp\left(\sum_{k=1}^{K_A} \lambda_{A,k} f_{A,k}(x, y_A)\right) = \frac{1}{Z(x)} \exp\left(\sum_{A\in\mathcal{C}} \sum_{k=1}^{K_A} \lambda_{A,k} f_{A,k}(x, y_A)\right)$$

Often the feature functions are binary with value $f_{A,k}(x, y_A) = 1$ if the feature is present and $f_{A,k}(x, y_A) = 0$ otherwise. If $\lambda_{A,k} = 0$ the corresponding feature has no influence. For non-negative feature functions positive values for $\lambda_{A,k}$ indicate that the feature increases $p(y_A|x)$, while negative values decrease the conditional probability and have to be estimated from training data by maximum likelihood.

A common special case is a *linear chain conditional random field*, where only interactions between y_t and y_{t-1} are allowed. If in addition we only take into account the corresponding inputs x_t and x_{t-1} the feature functions have the form $f_{\{t-1,t\},k}(x_{t-1}, x_t, y_{t-1}, y_t)$. Therefore only the adjacent states y_{t-1} and y_t influence each other directly. Figure 6.13 shows a simplified example of such a linear chain.

In many cases it is safe to assume that the parameters do not depend on the particular t and hence $\lambda_{\{t-1,t\},k} = \lambda_{\{t,t+1\},k}$ for all t. This parameter tying drastically reduces the number of unknown parameters. More generally, we may partition $\mathcal{C} = \{C_1, \ldots, C_Q\}$ where each C_q is a set of all A whose parameters are tied. Then we get the representation:

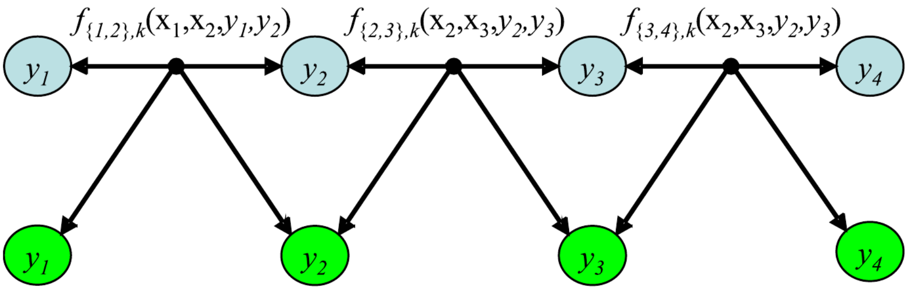

$f_{\{1,2\},k}(x_1,x_2,y_1,y_2)$
$f_{\{2,3\},k}(x_2,x_3,y_2,y_3)$
$f_{\{3,4\},k}(x_2,x_3,y_2,y_3)$
y_1
y_2
y_3
y_4
y_1
y_2
y_3
y_4

Figure 6.13: Example of a linear chain with four states and a single type of feature function

$$p(y|x;\lambda) = \frac{1}{Z(x)} \exp\left(\sum_{C_p \in \mathcal{C}} \sum_{A \in C_p} \sum_{k=1}^{K_A} \lambda_{p,k} f_{A,k}(x, y_A) \right) \tag{6.3}$$

We may estimate the unknown parameters according to the maximum likelihood criterion. Assume we have observed a number of independent and identically distributed observations $(x^{(1)}, y^{(1)}), \ldots, (x^{(N)}, y^{(N)})$, e.g. different documents which are already labeled with the states. Differentiating the log-likelihood function $\ell(\lambda) = \log \prod_n p(y^{(n)}|x^{(n)};\lambda)$ with respect to $\lambda_{p,k}$ yields:

$$\frac{\partial \ell(\lambda)}{\partial \lambda_{p,k}} = \sum_{n=1}^{N} \left[\sum_{A \in C_p} f_{A,k}(x^{(n)}, y_A^{(n)}) - \sum_{A \in C_p} \sum_{y_A \in Y_A} p(y_A|x^{(n)};\lambda) f_{A,k}(x^{(n)}, y_A) \right]$$

where Y_A is the set of all possible y_A and $p(y_A|x^{(n)};\lambda)$ is the probability of y_A given $x^{(n)}$ and the current parameter values λ.

The first sum contains the observed feature values for $f_{A,k}(x^{(n)}, y_A^{(n)})$ and the second sum consists of the expected feature values given the current parameter λ. If the gradient is zero both terms have to be equal. It can be shown that the log likelihood function is concave and hence may be efficiently maximized by second-order techniques such as conjugate gradient or L-BFGS [259]. To improve generalization a quadratic penalty term may be added which keeps the parameter values small.

Gradient training requires the computation of the marginal distributions $p(y_A|x^{(i)})$. In the case of a linear chain CRF this can efficiently done by the forward-backward algorithm requiring $2N$ steps. Networks with cycles require more effort as the exact computation grows exponentially with the diameter (see section 6.4.2 on loopy belief propagation).

If the parameters are known we have to determine the most likely state configuration for a new input $x^+ = (x_1^+, \ldots, x_n^+)$

$$y^* = \arg\max_y p(y|x^+;\lambda)$$

which in the case of linear chain models can be efficiently calculated by dynamic programming using the Viterbi algorithm. During prediction the linear-chain CRF takes into account the correlations between adjacent states, which for many problems increase the prediction quality.

6.4.2 Linear-Chain and Graph-Structured CRFs

Linear chain CRFs have already seen successful application in the extraction of structural information from scientific papers. In their header extraction task Peng and McCallum [212] consider the first part of a paper which has to be labeled with the following states: title, author, affiliation, address, note, email, date, abstract, introduction, phone, keywords, web, degree, publication number, and page. A second reference task labels the references at the end of a paper with the following states: author, title, editor, book title, date, journal, volume, tech, institution, pages, location, publisher, note. They used the following features:

- Local features describing the current word x_i: word itself, starts with capital letter, only capital letters, contains digit, only digits, contains dot, contains "–", acronym, capital letter and dot, matches regular expressions for phone number, zip code, URL, email.
- Layout features: word at beginning of a line, word in the middle of a line, word at the end of a line.
- External lexicon features: word in author list, word in date list (e.g. Jan., Feb.), word in notes.

On a training set with 500 headers they achieve an average F1 of 94% for the different fields, compared to 90% for SVMs and 76% for HMMs. For the reference extraction task trained on 500 articles they yield and F1-value of 91.5% compared to 77.6% for an HMM. A Gaussian prior was found to consistently perform best for the aforementioned task. Schneider [233] uses linear CRFs to extract information like conference names, titles, dates, locations, and submission deadlines from call for papers with the goal to compile conference calenders automatically. He models the sequence of words in a conference paper and uses the following layout features: first/last token in the line, first/last line in the text, line contains only blanks/punctuations, line is indented, in first 10/20 lines of the text. Using a training dataset of 128 conference papers they achieve an average F1-value of about 60–70% for the title, date and other fields of a conference paper.

These examples show us that linear CRFs can be used with a very good success rate for cases where the interdependencies between the logical elements are relatively simple. However, our real target for mass digitization consists of complex, multi-column documents. We will now discuss graph-like structures which allow for the modeling of more complex dependencies.

As seen from chapter 5, geometric layout analysis segments each document page into a set of regions and assign physical labels to them, such as text, line-art, photograph, separator, etc.. Since the location and shape of each region is readily available at the start of logical layout analysis, one may now go further and establish a set of basic spatial relationships between the elements, such as "touch", "below", "above", "right of", etc.. Especially in documents featuring a multi-column layout the sequence of paragraphs of an article in different columns or even on continuation pages is not unique. In the same way the assignment of tables, captions, figures and images located somewhere on the page to an article represents a challenging problem. For a subset of all possible object pairs relations r_j may be specified, e.g. *photograph* **left-of** *article*, *photograph* **belongs-to** *article*, *article* **below** *article*. Each object and each relation has an associated type $t(o_i)$ or $t(r_i)$. Depending on their type each object and each relation is characterized by type-specific attributes, e.g. topic, title, or x-y position. This yields for each type t a type-specific attribute vector $x_{o_i}^{t(o_i)}$ for an object or and attribute vector $x_{r_i}^{t(r_i)}$ for a relation. The following figure shows a small example network of relations between articles and images of a newspaper. A probabilistic relational network [263] represents a joint distribution over the attributes $x_{o_i}^{t(o_i)}$ and $x_{r_i}^{t(r_i)}$ of objects and relations.

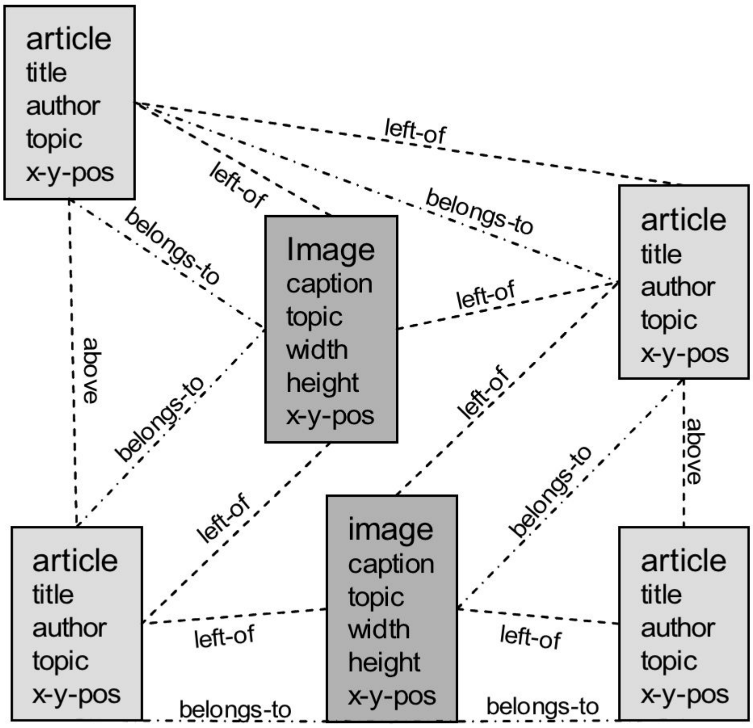


Figure 6.14: Toy example: Articles and images on periodical page can be characterized via a set of attributes. Different types of relations may exist between the pairs of logical entities. Source: [207]

Attributes of an object or relation can depend probabilistically on other attributes of the same or other objects or relations. For example the probability of image belonging to an article is higher if it is located close to the main article body. In the same way the probability of an image belonging to an article is higher if the topic of the caption and the topic of the article are similar. These dependencies can now be exploited.

In a linear chain CRF we had a generic dependency template between the states of successive states in the chain. This resulted in using the same parameters independent of the step index or the specific sentence. In the same way probabilistic relational models may define a number generic dependency templates depending on the types of the involved items. This approach of typing items and tying parameters across items of the same type is an essential component for the efficient learning of probabilistic relational models. It enables generalization from a single instance by decomposing the relational graph into multiple examples of each item type (e.g. all image objects), and building a joint model of dependencies between and among attributes of each type. The resulting probabilistic dependency network is a graph-structured CRF (equation 6.3), where parameters are tied in a specific way. For more details on this model one may consult the recent book chapter of Sutton and McCallum [259]. Many other variants of CRF models have been developed in recent years. Dynamic conditional random fields [260] are sequence models which allow multiple labels at each time step, rather than single labels as in linear-chain CRFs. Lumped label CRFs [206] allow to include observations, where only a subset of labels is observed and it is known that one of the labels in the subset is the true label. Finally, Markov logic networks [223] are a type of probabilistic logic network in which there are parameters for each first-order rule in a knowledge base. These first-order rules may, for example, be exploited to specify constraints between layout elements.

Parameter estimation for general CRFs is essentially the same as for linear-chains, except that computing the model expectations requires more general inference algorithms. Whenever the structure of the relationships between elements form an undirected graph, finding exact solutions require special graph transformations and eventually the enumeration of all possible annotations on the graph. This results in the exponential complexity of model training and inference. To make it tractable, several approximation techniques have been proposed for undirected graphs; these include variational and Markov Chain Monte Carlo methods. A number of alternatives exist:

- Gibbs sampling [103], where for each training example the labels are selected randomly according to the conditional distribution (equation 6.2). The required probabilities can be estimated from the resulting joint distribution of labels.
- Loopy belief propagation [258], performing belief propagation, which is an exact inference algorithm for tree-structured networks, ignoring part of the links.
- Pseudo-likelihood approaches [56] which instead of the predicted labels use the observed label values to predict a label from its environment.

Chidlovskii and Lecerf [76] use a variant of probabilistic relational models to analyze the structure of documents. They aim at annotating lines and pages in layout-oriented documents which correspond to the beginning of sections and section titles. While for a local network corresponding to linear chain CRFs they obtain a F1-value of 73.4%, switching to a graph-structured probabilistic relational network increases the F1-value to 79.5%. There exist other, more heuristic, models which take into account graph-structured dependencies. For example, Wisniewski and Gallinari [284] consider the problem of sequence labeling and propose a two step method. First the authors employ a local classifier for the initial assignment of elements without taking into account any dependencies. Then a relaxation process successively takes into account non-local dependencies to propagate information and ensure global consistency. They test their approach on a collection of 12000 course descriptions which have to be annotated with 17 different labels such as lecturer, title, start time or end time; each description contains between 4 and 552 elements to be extracted. For a CRF they report an F1-value of 78.7%, for a Probabilistic Context Free Grammar using maximum entropy estimators to estimate probabilities they yields 87.4% and the relaxation model arrives at an F1-value of 88.1%.

From these examples it is clear that graph-structured CRFs can be used successfully for more complex labeling problems and that they are in fact a more powerful tool than linear-structured CRF in this context. The remaining question however, is whether it is indeed practically feasible in mass digitization projects to obtain the necessary training data for both CRF models without a heavy reliance on human annotators. Ideally, for changing layouts the DIU system would only need to see a few examples for each label type – that is at most 5–10 annotated pages of printed material per layout. For both the linear and the graph-structured CRF models this amount of training data is most likely not enough. Thus, an interesting investigation direction is the artificial augmentation of the training sets using a set of principled algorithms, as discussed for newspaper material by Strecker et al. [254] and by Baird and Casey [52] for the generic case.

6.5 Conclusions

Logical layout analysis aims at extracting the (hierarchical) relations between physical components, as well as labeling and segmenting the physical regions into consistent logical regions, such as articles, sections, titles, footnotes, a.s.o.. When available, the logical document structure allows the application of a multitude of electronic document tools, including markup, hyperlinking, hierarchical browsing and component-based retrieval [256]. Most importantly, the higher-level structuring of the information allows humans to handle electronic documents in a manner which is more natural and efficient. In this chapter we talked about modern approaches for document logical structure recognition. In the first section we reviewed the state of the art in logical layout analysis and mentioned commercial OCR products working actively towards this goal.

As a gentler introduction to the problems posed by logical layout analysis, we introduced a novel, semi-automatic approach for front page detection targeted at large collections of periodicals. The proposed front page detection algorithm combines the advantages of statistical and structural pattern recognition methods. As presented in detail in section 6.2.1, statistical models are used for describing and detecting the salient parts of a front page, which are in turn connected into a structural model. In this way, the amount of noise (i.e. false graph vertices or edges) which must be dealt with during the creation and matching of the structural model is greatly reduced. We proposed a specialized graph edge weighting function, which is plays a key role in computing the distance scores between a model and each candidate. This allowed us to use a simple and efficient matching algorithm even for larger graphs, while still achieving a high degree of robustness. Our test set consisted of 17 572 images from 1141 newspaper issues printed during a time interval of around 60 years. Experimental results showed that the proposed approach is fast and robust enough for real-life mass digitization. As topic for future investigation remains the issue of automated salient component identification, as well as the question regarding the best features/feature sets to use in this context. Another promising research direction is the automatic detection of brand marks from scanned material, with direct applicability in media research and marketing.

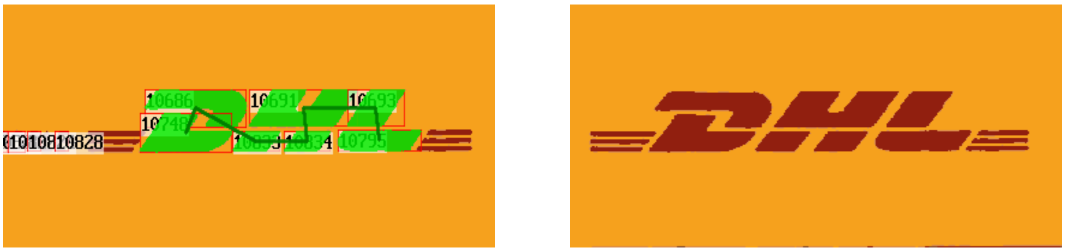


Figure 6.15: Example of model-based brand mark detection on the DHL logo, by using the front page detection algorithm proposed in section 6.2

Section 6.3 presented a practical approach for performing a complete logical layout analysis on heterogeneous periodical collections. It directly builds upon the results of the geometric layout analysis step in order to extract and label the logical zones from a document image, sort them in correct reading order and finally assemble the logical building blocks into complete articles. The proposed algorithm makes use of minimum spanning trees for the article segmentation task and relies on adaptable rule sets for the logical labeling tasks. Most importantly, it has already been extensively used in large (> 500 000 pages) research and industry digitization projects with good empirical results. Rigorous evaluations on a three real-life data sets consisting of complex layout newspapers and chronicles showed that the approach works reliably in practice where the layouts do not differ wildly. As this may not always be the case in mass digitization, the key area for future improvements was identified as being the logical labeling process.

Building upon this insight, we described a theoretical model through which the manual rule sets may be abolished in favor of automatic machine learning approaches. We introduced CRFs as a general probabilistic algorithm to model the uncertain relation between inputs and outputs. The model may be adapted to the geometrical document structure, which may be set up as a linear sequence or a general graph. It is noteworthy that via this model one may potentially be able to take a large number of interrelated features into account in order to infer a document's logical structure. While the introduced trainable model is much more promising than the rule-based alternatives, it also requires far more computational resources as well as more training data. Since generally not enough training data is available under a mass digitization scenario, we mentioned as possible solution the enrichment of manual ground truth with automatically generated samples. A start in this direction has recently been made by Strecker et al. [254] and Baird and Casey [52].

In the future traditional document structure recognition approaches must be enhanced by machine learning approaches, which are able to take into account publications as whole entities. Another pressing issue in the area of logical layout analysis is that there currently do not exist any standardized evaluation sets and corresponding benchmarks. As such, objectively comparing the results of different approaches is a challenging endeavor. This is a gap that needs to be filled in future research, as principled evaluation is the only way to convincingly demonstrate advances in logical layout analysis.

Chapter 7

Conclusions and Outlook

Part of the inhumanity of the computer is that, once it is competently programmed and working smoothly, it is completely honest.

– I. Asimov

In the preceding chapters, we discussed several methods to effectively enable the automatic processing of large, heterogeneous collections of books and periodicals. Nearly all methods introduced in this work have been implemented and extensively tested as part of the modular Fraunhofer DIU system in the processing of more than 1 000 000 document scans.

We proposed a generic framework for document image understanding systems, usable for practically any document types available in digital form. Starting the second chapter, we presented an automatic method for the *detection of color reference charts* from any kind of digitized materials. Color reference charts nowadays represent a minimal requirement in any professional digitization project [120]. The evaluation on a set of 239 real-life document- and photograph scans featuring partially occluded color targets in different orientations yielded a recall rate of 97.1%, which confirms the robustness of our algorithm. The proposed method enables a precise quality assessment and thus allows the implementation of a quick sanity check as the first stage of a mass digitization endeavor. For enabling the processing of highly degraded historical documents we described an algorithm which allows the exact *extraction of the foreground*. Unlike the vast majority of the scientific works concerning document image enhancement, the novel algorithm is capable of making use of full color information and does not require any prior training or apriori information about the document or the noise component. Because of the explicit modeling of the scanned document as the result of three independent processes (i.e. printing, paper- and ink degradation, noise) we are however limited to documents printed using a single ink color. The proposed FastICA-based algorithm is shown to provide an order of magnitude improvement in OCR performance over traditional document analysis approaches on a test set comprising around 43 000 characters. Potential for improvement was identified in the form of incorporating spatial proximity information into the approach to complement the purely color-based information currently exploited. As additional technique for improving the results of optical character recognition on old documents, we proposed a method for the effective *removal of letterpress printing artifacts.* Such artifacts were commonly found in documents produced via hot metal typesetting, which was the first non-manual printing technique widely used throughout the late 19^{th} and early 20^{th} century by many publishers around the globe. The stroke-like nature of the artifacts makes traditional filtering approaches unfeasible. The proposed method uses a simple, interchangeable decision tree classifier and font- and script-independent features computed from extracted directional single-connected chains. Most importantly, we were able to significantly relax the traditional assumptions about such artifacts in the document analysis community and show that a resolution-independent processing of prints exhibiting artifacts with a stroke width even higher than that of many thin characters is indeed possible. Our approach was evaluated on a 63 000 character dataset consisting of old newspaper excerpts containing both affected and clean regions via two independent OCR engines (ABBYY FineReader and Tesseract). For both engines a significant improvement in OCR quality was observed, as the edit distance decreased by an average of around 42%. Further research shall focus on the automatic detection of affected regions in order to allow a selective (and consequently much more effective) application of the algorithm, as well as on evaluating different combinations of classifiers and features for a more accurate artifact classification.

The third chapter saw the introduction of a generic framework for the *adaptation of global binarization algorithms into local ones*. The framework achieves the theoretically lowest bound on computational complexity (constant per pixel) and it allows the application of optimal thresholding techniques. It was validated by the implementation of two optimal global algorithms, well-known in the document analysis community for their good results in practice [115, 237]. Visual results on newspaper scans showing irregular degradations confirm the versatility of the discussed framework. We show that traditional implementations have a practical running time dozens to hundreds of times higher than that of the proposed framework (e.g. for a local window radius of 125 pixels our implementation is around 30 times faster). A further significant improvement in running time (in tests from around 48 seconds to under 1 second per DIN A4 scan) can be obtained via a sampling grid proportional to the detected dominant character height. With concern to the challenging problem of *color reduction for document images*, we presented an overview of available techniques and identified the deficiencies in the current state of the art in document analysis. We suggested possible improvements via the consequent use of vector operations and color models capable of reproducing prominent features of the human visual system. We believe such models are necessary due to the fact that printing techniques and layout styles currently in use have been optimized over the years specifically for human readers. In addition, we have discussed the problem posed by an objective and meaningful evaluation of color reduction results via the practical example of the system proposed by Rodriguez Valadez [224]. As future work we note that an ideal document color quantization framework must also be able to exploit style consistency over a set of related pages and intelligently combine the extracted color palettes in order to produce truly robust segmentation results.

As the next stage of a generic DIU system, we discussed the issue of skew and orientation detection. Global skew detection was selected as our focus, following the observation that it represents by far the most widespread skew type encountered in mass document digitization projects. Our original research contribution to the research area consists of two new algorithms for *global skew detection* falling in the category on nearest-neighbor approaches and employing the same framework. Important distinguishing features of the proposed algorithms are: the existence of a theoretical foundation, the discarding of the typical requirement regarding prior layout analysis and the absence of any explicit algorithm parameters. It is worth mentioning that despite its simplicity, the underlying Euclidean minimum spanning tree essentially approximates the text lines via piecewise linear functions, as opposed to the more error-prone state-of-the-art approach using a linear model. Finally, we introduced a way to compute and return meaningful confidence estimates for the global skew angles produced by both algorithms. The accuracy of the novel approaches was tested on a set of totaling about 110 000 real-life images. We included a comparison with other prominent research methods with regard to both skew- and orientation detection performance. As continuation to our work, we believe that methods for holistic skew estimation which are capable of integrating skew-related information from more sources represent a promising research direction. Some exploitable additional sources of information are for example the directions of solid separators, the edges of halftones and shading information.

We defined the task of *document page segmentation* in chapter 5 and discussed the delimitation between page segmentation and logical layout analysis. From the context of mass digitization, we introduced the flexible page segmentation approach implemented in the Fraunhofer DIU system. The core segmentation method is based on a version of the Jain and Yu's [142] generalized text line structure, enriched with dominant direction information. In contrast to existing approaches, we construct isothetic polygonal regions in a fully resolution-independent and noise-resilient manner by using as constraints the complete set of separators found in the document. We proposed a general-purpose solid separator detection method, obtained from the combination of two complementary research algorithms [112, 302]. White spaces are found via a robust version of Breuel's globally-optimal algorithm [59]. An extension of the region detection algorithm for arbitrary layouts (i.e. featuring text to non-text intersections) was introduced along with a variant allowing the segmentation of pages with multiple skew. Most importantly, in the latter case we pose no restrictions on the allowable skew angle difference. The described page segmentation algorithm was tested on three datasets containing a total of around 200 document scans with wide variations in layout complexity and image quality, having newspapers, modern magazines and historical documents as their respective focus. It is noteworthy that using the same parameter set and generic methods our approach was able to take the first place in the ICDAR 2009 page segmentation competition [38] and place third in the ICDAR 2011 historical document layout analysis competition [33] at a close distance to the winning algorithm. To our knowledge, the page segmentation method described in this work is the first approach able to obtain provably good results on a large, heterogeneous dataset resembling real-life mass digitization. As promising research directions we identified the ability to seamlessly process multi-thresholded document images (as resulting from generic color reduction algorithms) and the development of page segmentation algorithms capable of outputting an overall or per-region confidence values/intervals.

Logical layout analysis was the final subject discussed in the work at hand. This is in direct relation to the difficulty of the problem, since LLA algorithms need to be able to cope with the partially erroneous data produced by all earlier processing stages. As a simpler instance of LLA, we introduced a novel, semi-automatic approach for *front page detection* targeted at large collections of periodicals. The proposed algorithm combines the advantages of statistical and structural pattern recognition methods. Gaussian distributions using DCT-based feature descriptors represent the statistical models describing the salient parts of a front page, which are in turn connected into a topological model using the Delaunay triangulation. Following this approach, we greatly reduce the amount of noise (i.e. false graph vertices or edges) during the creation and matching of the structural model. A specialized graph edge weighting function was proposed in order to allow us the use of an efficient, yet still robust matching algorithm even for larger graphs. Our experimental results on a test set consisting of around 17 500 newspaper images from 1141 issues showed that the proposed approach is sufficiently fast and robust for real-life digitization projects. The issue of fully automated salient component identification, as well as an extended investigation on the best feature set to use in the context of front page detection remain as topics for forthcoming research.

A step up in difficulty, we described a practical approach for performing a *complete logical layout analysis* on periodical publications. The proposed method refines the textual regions produced by the page segmentation via a dynamic programming approach, sorts them in reading order using a topological sort and performs a preliminary labeling of titles and caption. For each pair of text regions we compute a compound distance measure comprising similarities of textual, logical and labeling attributes along with normalized physical distance and in-between separators. The minimum spanning tree is computed from the set of edges and is directly used to produce an article over-segmentation. The reading order is used to iteratively merge matching adjacent article parts. The last step consists of labeling each region with its final logical type label via a combined set of publisher-specific and generic rules using all previously computed features along with the added logical restrictions among regions located within same articles. We evaluate the approach experimentally with respect to both segmentation and labeling accuracy on three data sets comprising in total around 250 images (2200 articles) and consisting of complex layout newspapers and chronicles. In view of large collections of material showing significant layout variations, we identify the logical labeling process as the key area for future improvements. Following this observation, we sketched a *theoretical model* using conditional random fields through which the manual rule sets may be abolished in favor of automatic machine learning approaches. It is worth observing that via the CRF-based model one may in principle be able to take a larger number of interrelated features into account, i.e. combine the spatial and structural features with textual (OCR) ones. However, training the graph-structured CRF model requires far more computational resources as well as more training data. Since generally training data is relatively scarce and hard to produce, we mentioned as possible solution the enrichment of manual ground truth with automatically generated samples, as discussed by Strecker et al. [254] and Baird and Casey [52]. In the area of logical layout analysis, promising directions for future work are machine learning approaches treating publications as whole entities and the development of standardized and open evaluation sets and corresponding benchmarks.

Bibliography

[1] ABBYY FineReader. http://finereader.abbyy.com/. accessed 09-Feb-2012.

[2] CCS DocWorks. http://www.ccs-gmbh.de/en/products/docworks. accessed 16-Feb-2012.

[3] ColorChecker Classic. http://xritephoto.com/ph_product_overview.aspx?id=1192&catid=28&action=overview. accessed 13-Feb-2012.

[4] ColorChecker Digital SG. http://xritephoto.com/ph_product_overview.aspx?id=938&catid=28&action=overview. accessed 09-Feb-2012.

[5] ColorChecker Passport Camera Calibration Software. http://xritephoto.com/ph_product_overview.aspx?ID=1257. accessed 09-Feb-2012.

[6] Deutsche Digitale Bibliothek. http://www.deutsche-digitale-bibliothek.de/. accessed 21-Feb-2012.

[7] Europeana: think culture. http://www.europeana.eu/portal/. accessed 21-Feb-2012.

[8] Google Book Search. http://books.google.com/. accessed 09-Feb-2012.

[9] ImageWare - Complete solutions for digitisation projects. http://www.imageware.de/en/. accessed 09-Feb-2012.

[10] Impact - Improving access to text. http://www.impact-project.eu/about-the-project/concept/. accessed 09-Feb-2012.

[11] IRISDocument - OCR Server. http://www.irislink.com/c2-1600-189/IRISDocument-9---OCR-Server.aspx. accessed 16-Feb-2012.

[12] Metamorfoze Programme. http://www.metamorfoze.nl/programme. accessed 09-Feb-2012.

[13] Ocropus - The OCRopus(tm) open source document analysis and OCR system. http://code.google.com/p/ocropus/. accessed 09-Feb-2012.

[14] Omnipage. http://www.nuance.com/for-individuals/by-product/omnipage/index.htm. accessed 09-Feb-2012.

[15] Tesseract-OCR. http://code.google.com/p/tesseract-ocr. accessed 09-Feb-2012.

[16] The European Library. http://search.theeuropeanlibrary.org/portal/en/index.html. accessed 21-Feb-2012.

[17] Theseus - Contentus. http://theseus-programm.de/en/922.php. accessed 09-Feb-2012.

[18] P.K. Agarwal, H. Edelsbrunner, O. Schwarzkopf, and E. Welzl. Euclidean minimum spanning trees and bichromatic closest pairs. *Discrete and Computational Geometry*, 6(6):407–422, December 1991.

[19] H.K. Aghajan, B.H. Khalaj, and T. Kailath. Estimation of skew angle in text-image analysis by slide: subspace-based line detection. *Machine Vision and Applications*, 7:267–276, 1994.

[20] M. Agrawal and D. Doermann. Clutter noise removal in binary document images.

In *Proc. Int. Conf. Document Analysis and Recognition*, pages 556–560, 2009.

[21] M. Agrawal and D. Doermann. Stroke-like pattern noise removal in binary document images. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 17–21, 2011.

[22] M. Aiello, C. Monz, and L. Todoran. Combining linguistic and spatial information for document analysis. In *Proc. Int. Conf. Computer-Assisted Information Retrieval*, pages 266–275, 2000.

[23] O. Akindele and A. Belaid. Page segmentation by segment tracing. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 341–344, 1993.

[24] T. Akiyama and N. Hagita. Automated entry system for printed documents. *Pattern Recognition*, 23:1141–1154, 1990.

[25] B. Allier, N. Bali, and H. Emptoz. Automatic accurate broken character restoration for patrimonial documents. *Document Analysis and Recognition*, 8(4):246–261, 2006.

[26] N. Amamoto, S. Torigoe, and Y. Hirogaki. Block segmentation and text area extraction of vertically/horizontally written document. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 739–742, 1993.

[27] C. An, H. Baird, and P. Xiu. Iterated document content classification. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 252–256, 2007.

[28] A. Antonacopoulos. Local skew angle estimation from background space in text regions. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 684–688, 1997.

[29] A. Antonacopoulos and D. Bridson. Performance analysis framework for layout analysis methods. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 1258–1262, 2007.

[30] A. Antonacopoulos, D. Bridson, C. Papadopoulos, and S. Pletschacher. A realistic dataset for performance evaluation of document layout analysis. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 296–300, 2009.

[31] A. Antonacopoulos and B. Brough. Methodology for flexible and efficient analysis of the performance of page segmentation algorithms. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 451–454, 1999.

[32] A. Antonacopoulos and C.C. Castilla. Flexible text recovery from degraded typewritten historical documents. In *Proc. Int. Conf. Pattern Recognition*, pages 1062–1065, 2006.

[33] A. Antonacopoulos, C. Clausner, C. Papadopoulos, and S. Pletschacher. Historical document layout analysis competition. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 1516–1520, 2011.

[34] A. Antonacopoulos, B. Gatos, and D. Bridson. ICDAR2005 page segmentation competition. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 75–79, 2005.

[35] A. Antonacopoulos, B. Gatos, and D. Bridson. ICDAR2007 page segmentation competition. In *Proc. Int. Conf. Document Analysis and Recognition*, volume 2, pages 1279–1283, 2007.

[36] A. Antonacopoulos, B. Gatos, and D. Karatzas. ICDAR2003 page segmentation competition. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 688–692, 2003.

[37] A. Antonacopoulos, D. Karatzas, and D. Bridson. Ground truth for layout analysis performance evaluation. *Lecture Notes in Computer Science*, 3872:302–311, 2006.

[38] A. Antonacopoulos, S. Pletschacher, D. Bridson, and C. Papadopoulos. ICDAR2009 page segmentation competition. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 1370–1374, 7 2009.

[39] A. Antonacopoulos and R.T. Ritchings. Representation and classification of complex-

shaped printed regions using white tiles. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 1132–1135, 1995.

[40] H. Aradhye. A generic method for determining up/down orientation of text in roman and non-roman scripts. *Pattern Recognition*, 38(11):2114–2131, 2005.

[41] D. Arthur and S. Vassilvitskii. k-means++: the advantages of careful seeding. In *Proc. ACM-SIAM Symp. Discrete algorithms*, pages 1027–1035, 2007.

[42] Adobe Developers Assoc. *TIFF 6.0 Specification.* Adobe Systems Inc., 6 1992. accessed 12-Jul-2012.

[43] A. Atiquzzaman and M.W. Akhtar. A robust hough transform technique for complete line segment description. *Real-Time Imaging*, 1:419–426, 1995.

[44] L. Aurdal. Image segmentation, thresholding — an introduction. Norwegian Computing Center (Norsk Regnesentral), 9 2004.

[45] I. Avcibas, B. Sankur, and K. Sayood. Statistical evaluation of image quality measures. *J. Electronic Imaging*, 11(2):206–223, 2002.

[46] B.T. Ávila and R.D. Lins. A fast orientation and skew detection algorithm for monochromatic document images. In *Proc. ACM Symp. Document engineering*, pages 118–126. ACM, 2005.

[47] B.T. Ávila, R.D. Lins, and L.A.O. Neto. A new rotation algorithm for monochromatic images. In *Proc. ACM Int. Conf. Document Engineering*, 2005.

[48] A.D. Bagdanov and J. Kanai. Evaluation of document image skew estimation techniques. *Document Recognition III*, 2660:343–353, 1996.

[49] A.D. Bagdanov and J. Kanai. Projection profile based skew estimation algorithm for JBIG compressed images. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 401–405, 1997.

[50] H.S. Baird. The skew angle of printed documents. In *Proc. Conf. Soc. Photographic Scientists and Engineers*, volume 40, pages 21–24, 1987.

[51] H.S. Baird. Anatomy of a versatile page reader. *Proc. IEEE*, 80:1059–1065, 1992.

[52] H.S. Baird and Matthew R. Casey. Towards versatile document analysis systems. In *Proc. Int. Workshop Document Analysis Systems*, pages 280–290, 2006.

[53] H.S. Baird, S.E. Jones, and S.J. Fortune. Image segmentation by shape-directed covers. In *Proc. Int. Conf. Pattern Recognition*, pages 820–825, 1990.

[54] F.A. Baqai, J.-H. Lee, A.U. Agar, and J.P. Allebach. Digital color halftoning. *IEEE Signal Proc. Magazine*, 22(1):87–96, 2005.

[55] A. Belaïd, I. Falk, and Y. Rangoni. Xml data representation in document image analysis. In *Proc. Int. Conf. Document Analysis and Recognition*, volume 1, pages 78–82, 9 2007.

[56] J. Besag. Statistical analysis of non-lattice data. *The Statistician*, 24(3):179–195, 1975.

[57] A. Beutelspacher. *Kryptologie.* Vieweg, 2005.

[58] D.S. Bloomberg, G.E. Kopec, and L. Dasari. Measuring document image skew and orientation. In *Proc. SPIE Document Recognition II*, pages 302–316, 1995.

[59] T.M. Breuel. Two algorithms for geometric layout analysis. In *Proc. Workshop Document Analysis Systems*, volume 3697, pages 188–199, 2002.

[60] T.M. Breuel. An algorithm for finding maximal whitespace rectangles at arbitrary orientations for document layout analysis. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 66–70, 2003.

[61] T.M. Breuel. High performance document layout analysis. In *Symp. Document Image Understanding Technology, Greenbelt, Maryland*, 2003.

[62] S.S. Bukhari, F. Shafait, and T. Breuel. Dewarping of document images using

coupled-snakes. In *Proc. Int. Workshop Camera-Based Document Analysis and Recognition*, pages 34–41, 7 2009.

[63] S.S. Bukhari, F. Shafait, and T. Breuel. An image based performance evaluation method for page dewarping algorithms using sift features. In *Int. Workshop Camera-Based Document Analysis and Recognition*, pages 138–149. Springer, 9 2011.

[64] M. Cannon, J. Hochberg, and P. Kelly. Quality assessment and restoration of typewritten document images. *Int. J. Document Analysis and Recognition*, 2:80–89, 1999.

[65] A. Carbonaro and P. Zingaretti. A comprehensive approach to image-contrast enhancement. In *Proc. Int. Conf. Image Analysis and Processing*, pages 241–246, 1999.

[66] R. Cattoni, T. Coianiz, S. Messelodi, and C. Modena. Geometric layout analysis techniques for document image understanding: a review. Technical Report 9703-09, ITC-irst, 1998.

[67] F. Cesarini, M. Lastri, S. Marinai, and G. Soda. Encoding of modified x-y trees for document classification. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 1131–1135, 2001.

[68] A. Chalechale, A. Mertins, and G. Naghdy. Edge image description using angular radial partitioning. *Vision, Image and Signal Processing*, 151(2):93–101, 2004.

[69] F. Chang, C.-J. Chen, and C.-J. Lu. A linear-time component-labeling algorithm using contour tracing technique. *Comput. Vis. Image Underst.*, 93:206–220, 2004.

[70] H.-J. Chang, K.-C. Huang, and C.-H. Wu. Determination of sample size in using central limit theorem for weibull distribution. *Information and Management Sciences*, 17(3):31–46, 2006.

[71] H.-J. Chang, C.-H. Wu, J.-F. Ho, and P.-Y. Chen. On sample size in using central limit theorem for gamma distribution. *Information and Management Sciences*, 19(1):153–174, 2008.

[72] P.-R. Chang and C.-C. Chang. Color correction for scanner and printer using B-spline CMAC neural networks. In *Proc. Asia-Pacific Conf. Circuits and Systems*, pages 24–28, December 1994.

[73] J. Chen, M. K. Leung, and Y. Gao. Noisy logo recognition using line segment Hausdorff distance. *Pattern Recognition*, 36(4):943–955, April 2003.

[74] S. Chen and R.M. Haralick. An automatic algorithm for text skew estimation in document images using recursive morphological transforms. In *Proc. Int. Conf. Image Processing*, pages 139–143, 1994.

[75] S. Chen, S. Mao, and G.R. Thoma. Simultaneous layout style and logical entity recognition in a heterogeneous collection of documents. In *Proc. Int. Conf. Document Analysis and Recognition*, volume 1, pages 118–122, 2007.

[76] B. Chidlovskii and L. Lecerf. Stacked dependency networks for layout document structuring. *J. Univ. Comp. Sci.*, 14(18):2998–3010, 2008.

[77] C.K. Chow and T. Kaneko. Automatic boundary detection of the left ventricle from cineangiograms. *Computers and Biomedical Research*, 5:388–410, 1972.

[78] A. Cichocki and S. I. Amari. *Adaptive Blind Signal and Image Processing.* John Wiley and Sons, 2002.

[79] C. Clausner, S. Pletschacher, and A. Antonacopoulos. Scenario driven in-depth performance evaluation of document layout analysis methods. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 1404–1408, 2011.

[80] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):603–619, 2002.

[81] F.C. Crow. Summed-area tables for texture mapping. *Computer Graphics*, 18(3):207–212, July 1984.

[82] O. Cuisenaire. *Distance Transformations: Fast Algorithms and Applications to Medical Image Processing.* PhD thesis, Université Catholique de Louvain, Belgium, 1999.

[83] C. R. Dance. Perspective estimation for document images. In *Document Recognition and Retrieval IX*, pages 244–254. SPIE, 2001.

[84] P. E. Danielsson. Euclidean distance mapping. *Computer Graphics and Image Processing*, 14:227–248, 1980.

[85] A.K. Das, Chowdhuri S.P., and B. Chanda. A complete system for document image segmentation. In *Proc. Nat. Workshop Comp. Vision, Graphics and Image Processing*, pages 9–16, 2002.

[86] E.R. Davies. On the noise suppression and image enhancement characteristics of the median, truncated median and mode filters. *Pattern Recogn. Lett.*, 7(2):87–97, 1988.

[87] H. Déjean and J.-L. Meunier. On tables of contents and how to recognize them. *Int. J. Document Analysis and Recognition*, 12(1):1–20, 2009.

[88] C. di Ruberto. Recognition of shapes by attributed skeletal graphs. *Pattern Recognition*, 37(1):21–31, January 2004.

[89] A.P. Dias. Minimum spanning trees for text segmentation. In *Proc. Symp. Document Analysis and Information Retrieval*, 1996.

[90] M.B. Dillencourt, H. Sammet, and M. Tamminen. A general approach to connected-component labeling for arbitrary image representations. *J. ACM*, 39:253–280, 1992.

[91] D. Doermann. Page decomposition and related research. In *Proc. Symp. Document Image Understanding Technology*, pages 39–55, 1995.

[92] B. Dresevic, A. Uzelac, B. Radakovic, and N. Todic. Book layout analysis: Toc structure extraction engine. In Shlomo Geva, Jaap Kamps, and Andrew Trotman, editors, *Advances in Focused Retrieval*, pages 164–171. Springer, 2009.

[93] M.S. Drew and S. Bergner. Spatio-chromatic decorrelation for color image compression. *Sig. Proc.: Image Comm.*, 8:599–609, 2008.

[94] G.H. Dunteman. *Principal components analysis.* SAGE Publications, 1^{st} edition, 1989.

[95] S. Eickeler, S. Müller, and G. Rigoll. Recognition of JPEG compressed face images based on statistical methods. *Image and Vision Computing Journal*, 18(4):279–287, March 2000.

[96] F. Esposito, D. Malerba, and G. Semeraro. A knowledge-based approach to the layout analysis. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 466–471, 1995.

[97] F. Esposito, D. Malerba, G. Semeraro, S. Ferilli, O. Altamura, T.M.A. Basile, M. Berardi, M. Ceci, and N. Di Mauro. Machine learning methods for automatically processing historical documents: from paper acquisition to XML transformation. In *Proc. Int. Workshop Document Image Analysis for Libraries*, pages 328–335, 2004.

[98] K. Etemad, D. Doermann, and R. Chellappa. Multiscale segmentation of unstructured document pages using soft decision integration. *Pattern Recognition and Machine Intelligence*, 19:92–96, 1997.

[99] H. Ezaki, S. Uchida, A. Asano, and H. Sakoe. Dewarping of document image by global optimization. In *Proc. Int. Conf. Document Analaysis and Recognition*, pages 500–506, 2005.

[100] M.D. Fairchild. *Color Appearance Models.* John Wiley& Sons, 2^{nd} edition, 2005.

[101] M.D. Fairchild and G.M. Johnson. Meet iCAM: A next-generation color appearance model. In *IS&T/SID Color Imaging Conf.*, pages 33–38, 2002.

[102] J. Fan. Robust color image enhancement of digitized books. In *Pro. Int. Conf. Document Analysis and Recognition*, pages 561–565, 2009.

[103] J.R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proc. Annual Meeting Assoc. Computational Linguistics*, pages 363–370, 2005.

[104] Hans Fischer. *A history of the central limit theorem. From classical to modern probability theory.* Sources and Studies in the History of Mathematics and Physical Sciences. Springer New York, 2011.

[105] S. Fischer. Digital image processing: Skewing and thresholding. Master's thesis, University of New South Wales, Sydney, Australia, 2000.

[106] R. Fisher, S. Perkins, A. Walker, and E. Wolfart. Spatial filters - Laplacian/Laplacian of Gaussian. `http://homepages.inf.ed.ac.uk/rbf/HIPR2/log.htm`. accessed 09-Feb-2012.

[107] R.W. Floyd and L. Steinberg. An adaptive algorithm for spatial grey scale. *Proc. Soc. Inf. Display*, 17:75–77, 1976.

[108] Per-Erik Forssén. Maximally stable colour regions for recognition and matching. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2007.

[109] R. Furmaniak. Unsupervised newspaper segmentation using language context. In *Proc. Int. Conf. Document Analysis and Recognition*, volume 2, pages 619–623, 2007.

[110] D. Gao and Y. Wang. Decomposing document images by heuristic search. In *Proc. Int. Conf. Energy Minimization Methods in Comp. Vis. and Pat. Rec.*, pages 97–111, 2007.

[111] M.R. Garey, D.S. Johnson, and H.S. Witsenhausen. The complexity of the generalized lloyd-max problem. *IEEE Trans. Information Theory*, 28(2):255–256, 1982.

[112] B. Gatos, D. Danatsas, I. Pratikakis, and S. J. Perantonis. Automatic table detection in document images. In *Proc. Int. Conf. Advances in Pattern Recognition*, pages 609–618, 2005.

[113] B. Gatos, S.L. Mantzaris, and A. Antonacopoulos. First international newspaper contest. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 1190–1194, 2001.

[114] B. Gatos, S.L. Mantzaris, S.J. Perantonis, and A. Tsigris. Automatic page analysis for the creation of a digital library from newspaper archives. *Digital Libraries*, 3:77–84, 2000.

[115] B. Gatos, K. Ntirogiannis, and I. Pratikakis. ICDAR 2009 document image binarization contest (DIBCO 2009). In *Proc. Int. Conf. Document Analysis and Recognition*, pages 1375–1382, 2009.

[116] B. Gatos, I. Pratikakis, and S. J. Perantonis. Adaptive degraded document image binarization. *Pattern Recognition*, 39(3):317–327, March 2006.

[117] M. Gervautz and W. Purgathofer. A simple method for color quantization: octree quantization. In A.S. Glassner, editor, *Graphics gems*, pages 287–293. Academic Press Professional, Inc., 1990.

[118] S. Gold and A. Rangarajan. Graph matching by graduated assignment. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 239–244, 1996.

[119] R.E. Gonzalez, R.C. and Woods. *Digital Image Processing.* Prentice Hall, 2^{nd} edition, 2002.

[120] Still Image Working Group. *Technical Guidelines for Digitizing Cultural Heritage Materials: Creation of Raster Image Master Files.* Federal Agencies Digitization Guidelines Initiative (FADGI), 8 2010.

[121] M. Grundland and N.A. Dodgson. Decolorize: Fast, contrast enhancing, color to grayscale conversion. *Pattern Recognition*, 40(11):2891–2896, 2007.

[122] L. Guibas and J. Stolfi. Primitives for the manipulation of general subdivisions and

the computation of Voronoi diagrams. *ACM Trans. Graphics*, 4(2):74–123, April 1985.

[123] J. Ha, R. Haralick, and I. Phillips. Document page decomposition by the bounding-box projection technique. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 1119–1122, 1995.

[124] J. Ha, R. Haralick, and I. Phillips. Recursive X-Y cut using bounding boxes of connected components. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 952–955, 1995.

[125] K. Hajdar, O. Hitz, and R. Ingold. Newspaper page decomposition using a split and merge approach. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 1186–1189, 2001.

[126] R. Haralick. Document image understanding: Geometric and logical layout. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 385–390, 1994.

[127] C. Harris and M. Stephens. A combined corner and edge detector. In M. Matthews, editor, *Alvey vision conference*, volume 15, pages 147–151. Manchester, UK, 1988.

[128] P.E. Hart, N.J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *Systems, Science and Cybernetics*, 4(2):100–107, July 1968.

[129] J.A. Hartigan. *Clustering algorithms.* John Wiley and Sons, Inc., New York, 1975.

[130] X. He, R.S. Zemel, and M.A. Carreira-Perpi. Multiscale conditional random fields for image labeling. In *Proc. Int. Conf. Comp. Vision and Pattern Recognition*, pages 695–702, 2004.

[131] P. Heckbert. Color image quantization for frame buffer display. *SIGGRAPH Comput. Graphics*, 16(3):297–307, 7 1982.

[132] J.D. Hobby and T.K. Ho. Enhancing degraded document images via bitmap clustering and averaging. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 394–400, 1997.

[133] Q. Huang, W. Gao, and W. Cai. Thresholding technique with adaptive window selection for uneven lighting image. *Pattern Recogn. Lett.*, 26(6):801–808, 2005.

[134] J.J. Hull. Document image skew detection: survey and annotated bibliography. *Document Analysis Systems*, 2:40–64, 1998.

[135] A. Hyvaerinen. Fast and robust fixed-point algorithms for independent component analysis. *Neural Networks*, 3:626–634, 1999.

[136] A. Hyvaerinen, J. Karhunen, and E. Oja. *Independent Component Analysis.* John Wiley and Sons, 2001.

[137] International Color Consortium. *Specification ICC.1:2010-12 (Profile version 4.3.0.0).* ICC, 2010. accessed 13-Feb-2012.

[138] International Commission on Illumination. Cie colorimetry - part 4: 1976 l*a*b* colour space. Joint ISO/CIE Standard ISO 11664-4:2008(E)/CIE S 014-4/E:2007, CIE, 2008.

[139] ISO/IEC JTC1/SC29/WG11/N3321. MPEG-7 visual part of experimentation model version 5, March 2000.

[140] D. J. Ittner and H. S. Baird. Language-free layout analysis. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 336–340, 1993.

[141] A.K. Jain and K. Karu. Learning texture discrimination masks. *Pattern Analysis and Machine Intelligence*, 18:195–205, 1995.

[142] A.K. Jain and B. Yu. Document representation and its application to page decomposition. *Pattern Analysis and Machine Intelligence*, 20(3):294–308, 1998.

[143] A.K. Jain and Y. Zhong. Page segmentation using texture analysis. *Pattern Recog-*

nition, 29:743–770, 1996.

[144] I. T. Jolliffe. *Principal Component Analysis.* Springer-Verlag New York, 2^{nd} edition, 2002.

[145] J. Kanai, T.A. Nartker, S.V. Rice, and G. Nagy. Performance metrics for document understanding systems. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 424–427, 1993.

[146] J. Kanai, S.V. Rice, T.A. Nartker, and G. Nagy. Automatic evaluation of OCR zoning. *Pattern Analysis and Machine Intelligence*, 17(1):86–90, 1995.

[147] R. Kasturi, L. O'Gorman, and V. Govindaraju. Document image analysis: A primer. *Sadhana - Special Issue Indian Language Document Analysis and Understanding*, 27:3–22, 2002.

[148] D. Keysers, F. Shafait, and T. Breuel. Document image zone classification - a simple high-performance approach. In *Proc. Int. Conf. Computer Vision Theory and Applications*, pages 44–51, 2007.

[149] I.-K. Kim, D.-W. Jung, and R.-H. Park. Document image binarization based on topographic analysis using a water flow model. *Pattern Recognition*, 35:265–277, 2002.

[150] K. Kise, A. Sato, and M. Iwata. Segmentation of page images using the area Voronoi diagram. *Computer Vision and Image Understanding*, 70(3):370–382, 1998.

[151] J. Kittler and J. Illingworth. Minimum error thresholding. *Pattern Recognition*, 19:41–47, 1986.

[152] R. Klette and P. Zamperoni. *Handbook of image processing operators.* Wiley & Sons, 1996.

[153] I. Konya. Development of a newspaper image understanding system. Master's thesis, RWTH Aachen, 11 2006.

[154] I. Konya, S. Eickeler, and C. Seibert. Fast seamless skew and orientation detection in document images. In *Proc. Int. Conf. Pattern Recognition*, pages 1924–1928, 2010.

[155] I. Konya, C. Seibert, S. Eickeler, and S. Glahn. Constant-time locally optimal adaptive binarization. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 738–742, 2009.

[156] F. Kronenberg. Logo recognition from color documents using region-based shape descriptors. Master's thesis, RWTH Aachen, 11 2008.

[157] S. Kumar and M. Hebert. Man-made structure detection in natural images using a causal multiscale random field. In *Proc. Int. Conf. Comp. Vision and Pattern Recognition*, pages 119–126, 2003.

[158] M. Kuwahara, K. Hachimura, S. Eiho, and M. Kinoshita. Processing of riangiocardiographic images. In K. Preston and M. Onoe, editors, *Digital Processing of Biomedical Images*, pages 187–202. 1976.

[159] V. Lacroix. Automatic palette identification of colored graphics. In Jean-Marc Ogier, Wenyin Liu, and Josep Lladós, editors, *Graphics Recognition. Achievements, Challenges, and Evolution*, volume 6020 of *Lecture Notes in Computer Science*, pages 61–68. Springer Berlin/Heidelberg, 2010.

[160] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. Int. Conf. Machine Learning*, pages 282–289, 2001.

[161] K. Lee, Y. Choy, and S. Cho. Geometric structure analysis of document images: A knowledge-based approach. *Pattern Analysis and Machine Intelligence*, 22:1224–1240, 2000.

[162] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and re-

versals. *Soviet Physics Doklady*, 10(8):707–710, 1966.

[163] L.Gao, Z. Tang, X. Lin, and R. Qiu. Comprehensive global typography extraction system for electronic book documents. In *Proc. Int. Workshop Document Analysis Systems*, pages 615–621, 2008.

[164] J. Liang, R. Rogers, R.M. Haralick, and I.T. Phillips. UW-ISL document image analysis toolbox: An experimental environment. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 984–988, 1997.

[165] P. Liang, M. Narasimhan, M. Shilman, and P.A. Viola. Efficient geometric algorithms for parsing in two dimensions. In *Proc. Int.Conf. Document Analysis and Recognition*, pages 1172–1177, 2005.

[166] J. Lidke, C. Thurau, and C. Bauckhage. The snippet statistics of font recognition. In *Proc. Int. Conf. Pattern Recognition*, pages 1868–1871, 2010.

[167] X. Lin. Quality assurance in high volume document digitization: A survey. In *Proc. Int. Conf. Document Image Analysis for Libraries*, pages 312–319. IEEE, 2006.

[168] M. Liu, I. Konya, J. Nandzik, N. Flores-Herr, S. Eickeler, and P. Ndjiki-Nya. A new quality assessment and improvement system for print media. *EURASIP J. Advances in Signal Processing*, 109, 2012.

[169] W. Liu and D. Dori. From raster to vectors: Extracting visual information from line drawings. *Pattern Analysis & Applications*, 2:10–21, 1999.

[170] Y. Liu, J. Carbonell, P. Weigele, and V. Gopalakrishnan. Segmentation conditional random fields (SCRFs): A new approach for protein fold recognition. *J. Comput. Biology*, 13(2):408–422, 2006.

[171] R.P. Loce and E.R. Dougherty. *Enhancement and Restoration of Digital Documents.* SPIE Optical Engineering Press, 1997.

[172] H. Lu, A.C. Kot, and Y.Q. Shi. Distance-reciprocal distortion measure for binary document images. *IEEE Signal Processing Letters*, 11(2):228–231, 2 2004.

[173] J. Lu. *Signal recovery and noise reduction with wavelets.* PhD thesis, Dartmouth College, USA, 1993.

[174] Y. Lu and C.L. Tan. Improved nearest neighbor based approach to accurate document skew estimation. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 503–507, 2003.

[175] R. Lukac, B. Smolka, K. Martin, K.N. Plataniotis, and A.N. Venetsanopoulos. Vector filtering for color imaging. *IEEE Signal Processing Magazine*, 22(1):74–86, jan. 2005.

[176] S. Mandal, S.P. Chowdhuri, A.K. Das, and B.. Chanda. Automated detection and segmentation of form document. In *Proc. Int. Conf. Advances in Pattern Recognition*, pages 284–288, 2003.

[177] S. Mao and T. Kanungo. Empirical performance evaluation methodology and its application to page segmentation algorithms. *Pattern Analysis and Machine Intelligence*, 23:242–256, 2001.

[178] S. Mao, A. Rosenfeld, and T. Kanungo. Document structure analysis algorithms: A literature survey. In *Document Recognition and Retrieval X*, volume 5010, pages 197–207. SPIE, 2003.

[179] S. Marinai and H. Fujisawa, editors. *Machine Learning in Document Analysis and Recognition.* Springer, 2008.

[180] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. In *Proc. British Machine Vision Conf.*, pages 384–393, 2002.

[181] C.R. Maurer, R. Qi, and V. Raghavan. A linear time algorithm for computing exact euclidean distance transforms of binary images in arbitrary dimensions. *Pattern*

Analysis and Machine Intelligence, 25(2):265–270, 2003.

[182] C.S. McCamy, H. Marcus, and J.G. Davidson. A color-rendition chart. *J. Appl. Photogr. Eng.*, 2(3):95–99, 1976.

[183] G. McLachlan and T. Krishnan. *The EM Algorithm and Extensions.* John Wiley & Sons, 1997.

[184] B.T. Messmer. *Efficient Graph Matching Algorithms.* PhD thesis, University of Bern, Switzerland, 1995.

[185] J.-L. Meunier. Automated quality assurance for document logical analysis. In *Proc. Int. Conf. Pattern Recognition*, pages 253–256, 2010.

[186] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *Computer Vision*, 65(1–2):43–72, 11 2005.

[187] P.E. Mitchell and H. Yan. Newspaper document analysis featuring connected line segmentation. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 1181–1185, 2001.

[188] N. Moroney, M.D. Fairchild, R.W.G. Hunt, C. Li, M.R. Luo, and T. Newman. The ciecam02 color appearance model. In *Proc. IS&T/SID Color Imaging Conference*, pages 23–27, 2002.

[189] K. Murakami and T. Naruse. High speed line detection by Hough transform in local area. In *Proc. Int. Conf. Pattern Recognition*, volume 3, pages 467–470, 2000.

[190] G. Nagy and S. Seth. Hierarchical representation of optically scanned documents. In *Proc. Int. Conf. Pattern Recognition*, pages 347–349, 1984.

[191] B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM J. Computing*, 24(2):227–234, 1995.

[192] W. Niblack. *An Introduction to Digital Image Processing.* Prentice-Hall, Englewood Cliffs, NJ, 1986.

[193] N. Nikolaou and N. Papamarkos. Color reduction for complex document images. *Int. J. Imaging Syst. Technol.*, 19:14–26, 2009.

[194] D. Niyogi and S.N. Srihari. Knowledge-based derivation of document logical structure. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 472–475, 1995.

[195] D. Niyogi and S.N. Srihari. An integrated approach to document decomposition and structural analysis. *Imaging Systems and Technology*, 7:330–342, 1996.

[196] R. Nock and F. Nielsen. Statistical region merging. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(11):1452–1458, 11 2004.

[197] Univ. of Chicago. *The Chicago Manual of Style.* University of Chicago Press, Chicago, 13^{th} edition, 1982.

[198] L. O'Gorman. The document spectrum for page layout analysis. *Pattern Analysis and Machine Intelligence*, 15(11):1162–1173, 1993.

[199] H.-H. Oh and S.-I. Chien. Improvement of binarization method using a water flow model for document images with complex backgrounds. *Lecture Notes in A. I.*, 3157:812–822, 2004.

[200] Y. Ohta, T. Kanade, and T. Sakai. Color information for region segmentation. *Comp. Graphics and Image Proc.*, 13:222–241, 1980.

[201] O. Okun, D. Doermann, and M. Pietikainen. Page segmentation and zone classification: The state of the art. Technical Report LAMP-TR-036, CAR-TR-927, CS-TR-4079, University of Oulu, 1999.

[202] O. Okun, D. Doermann, and M. Pietikainen. Page segmentation and zone classification: a brief analysis of algorithms. In *Proc. Int. Congress Information Science Innovations*, pages 98–104, 2001.

[203] M. Orlowski. A new algorithm for the largest empty rectangle problem. *Algorithmica*, 5(1):65–73, 1990.

[204] N. Otsu. A threshold selection method from gray-level histograms. *Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[205] L. O'Gorman. Image and document processing techniques for the rightpages electronic library system. In *Proc. Int. Conf. Pattern Recognition*, pages 260–263, 1992.

[206] G. Paaß and F. Reichartz. Exploiting semantic constraints for estimating supersenses with CRFs. In *Proc. Int. Conf. Data Mining*, pages 485–496, 2009.

[207] G. Paaß and I. Konya. Machine learning for document structure recognition. In Alexander Mehler, Kai-Uwe Kühnbergerand, Henning Lobin, Harald Lüngen, Angelika Storrer, and Andreas Witt, editors, *Modeling, Learning and Processing of Text Technological Data Structures*, volume 370 of *Studies in Computational Intelligence.* Springer, 2011.

[208] U. Pal and B.B. Chaudhuri. An improved document skew angle estimation technique. *Pattern Recognition Letters*, 17:899–904, 1996.

[209] N. Papamarkos. A technique for fuzzy document binarization. In *Proc. ACM Symp. Document Engineering*, pages 152–156, 2001.

[210] M.B. Patel, J.J. Rodriguez, and A.F. Gmitro. Image classification based on focus. In *Proc. Int. Conf. Image Processing*, pages 397–400, 2008.

[211] T. Pavlidis and J. Zhou. Page segmentation and classification. *Graphical Models and Image Processing*, 54:484–496, 1992.

[212] F. Peng and A. McCallum. Information extraction from research papers using conditional random fields. *Inf. Process. Management*, 42:963–979, July 2006.

[213] W. B. Pennebaker and J. L. Mitchell. *JPEG Still Image Data Compression Standard.* Van Nostrand Reinhold, New York, 1993.

[214] S. Perreault and P. Hebert. Median filtering in constant time. *Image Processing*, 16(9):2389–2394, 2007.

[215] I.T. Phillips. User's reference manual for the UW english/technical document image database III. Technical report, Seattle University, Washington, 1996.

[216] S. Pletschacher and A. Antonacopoulos. The page (page analysis and ground-truth elements) format framework. In *Proc. Int. Conf. Pattern Recognition*, pages 257–260, 2010.

[217] W. Postl. Detection of linear oblique structures and skew scan in digitized documents. In *Proc. Int. Conf. Pattern Recognition*, pages 687–689, 1986.

[218] I. Pratikakis, B. Gatos, and K. Ntirogiannis. ICDAR 2011 document image binarization contest (DIBCO 2011). In *Proc. Int. Conf. Document Analysis and Recognition*, pages 1506–1510, 2011.

[219] W.H. Press, S.A. Teukolsky, W.T. Vettering, and B.P. Flannery. *Numerical Recipes in C++. The Art of Scientific Computing.* Cambridge University Press, 3^{rd} edition, 2007.

[220] Y. Rangoni and A. Belaïd. Document logical structure analysis based on perceptive cycles. In *Proc. Int. Workshop Document Analysis Systems*, pages 117–128. Springer, 2006.

[221] K. Rasche, R. Geist, and J. Westall. Re-coloring images for gamuts of lower dimension. *Comput. Graph. Forum*, pages 423–432, 2005.

[222] L. Ratinov, D. Roth, D. Downey, and M. Anderson. Local and global algorithms for disambiguation to wikipedia. In *Proc. Assoc. Computational Linguistics*, pages 1375–1384, 2011.

[223] M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62:107–

136, 2006.

[224] A. Rodriguez Valadez. Perceptually tuned color reduction for document images. Master's thesis, RWTH Aachen University, 2011.

[225] J. Sadri and M. Cheriet. A new approach for skew correction of documents based on particle swarm optimization. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 1066–1070, 2009.

[226] A. Said and W.A. Pearlman. A new, fast, and efficient image codec based on set partitioning in hierarchical trees. *IEEE Trans. Circuits and Systems for Video Technology*, 6(3):243–250, 6 1996.

[227] K. Sato and Y. Sakakibara. RNA secondary structural alignment with conditional random fields. *Bioinformatics*, 21(2):237–242, 2005.

[228] J. Sauvola and M. Pietikainen. Skew angle detection using texture direction analysis. In *Proc. Scandinavian Conf. Image Analysis*, pages 1099–1106, 1995.

[229] J. Sauvola and M. Pietikainen. Adaptive document image binarization. *Pattern Recognition*, 33:225–236, 2000.

[230] J. Sauvola, T. Seppanen, S. Haapakoski, and M. Pietikainen. Adaptive document binarization. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 147–152, 1997.

[231] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *Int. J. Computer Vision*, 37(2):151–172, 2000.

[232] F. R. Schmidt, D. Farin, and D. Cremers. Fast matching of planar shapes in sub-cubic runtime. In *IEEE Int. Conf. Computer Vision*, pages 1–6, 2007.

[233] K.-M. Schneider. Information extraction from calls for papers with conditional random fields and layout features. *J. Artif. Intell. Rev.*, 25:67–77, 2006.

[234] T.B. Sebastian, P.N. Klein, and B.B. Kimia. Recognition of shapes by editing their shock graphs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(5):550–571, 2004.

[235] S. Setlur, A. Lawson, V. Govindaraju, and S.N. Srihari. Large scale address recognition systems. truthing, testing, tools, and other evaluation issues. *Int. J. Document Analysis and Recognition*, 4(3):154–169, 2002.

[236] R. Sewall Hunter. Photoelectric color-difference meter. *J. Optical Society of America*, 48(12):985–993, 1958.

[237] M. Sezgin and B. Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *J. Electronic Imaging*, 13(1):146–165, January 2004.

[238] F. Shafait. Document image dewarping contest. In *Int. Workshop Camera-Based Document Analysis and Recognition*, pages 181–188, 2007.

[239] F. Shafait, D. Keysers, and T. M. Breuel. Performance comparison of six algorithms for page segmentation. In *Proc. Int. Workshop Document Analysis Systems*, pages 368–379. Springer, 2006.

[240] F. Shafait, D. Keysers, and T.M. Breuel. Pixel-accurate representation and evaluation of page segmentation in document images. In *Proc. Int. Conf. Pattern Recognition*, pages 872–875, 2006.

[241] F. Shafait, D. Keysers, and T.M. Breuel. Efficient implementation of local adaptive thresholding techniques using integral images. In *Proc. Conf. Document Recognition and Retrieval*, page 681510, 2008.

[242] F. Shafait, J. van Beusekom, D. Keysers, and T.M. Breuel. Document cleanup using page frame detection. *Int. J. Document Analysis and Recognition*, 11(2):81–96, 10 2008.

[243] G. Sharma. *Digital color imaging handbook*. Electrical engineering and applied signal processing series. CRC Press, 2003.

[244] G. Sharma, W. Wu, and E. N. Dalal. The CIEDE2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. *Color Research & Applications*, 30:21–30, 2005.

[245] Z. Shi and V. Govindaraju. Character image enhancement by selective region-growing. *Pattern Recognition Letters*, 17(5):523–527, 1996.

[246] C. Shih and R. Kasturi. Generation of a line-description file for graphics recognition. In *Proc. SPIE Conf. Applications of A. I.*, pages 568–575, 1988.

[247] P. Shirley, M. Ashikhmin, M. Gleicher, S. Marschner, E. Reinhard, K. Sung, W. Thompson, and P. Willemsen. *Fundamentals of Computer Graphics.* A.K. Peters, Ltd., 2^{nd} edition, 2005.

[248] E.H.B. Smith. An analysis of binarization ground truthing. In *Proc. Int. Workshop Document Analysis Systems*, pages 27–34, 2010.

[249] R. Smith. An overview of the tesseract ocr engine. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 629–633, 2007.

[250] R.W. Smith. Hybrid page layout analysis via tab-stop detection. In *Proc. Int.Conf. Document Analysis and Recognition*, pages 241–245, 2009.

[251] A.L. Spitz. Correcting for variable skew in document images. *Int. J. Document Analysis and Recognition*, 6:192–200, 2004.

[252] S.N. Srihari. Document image understanding. In *Proc. IEEE Comp. Soc. Fall Joint Comp. Conf.*, pages 87–96, 1986.

[253] S.N. Srihari and V. Govindaraju. Analysis of textual images using the Hough transform. *Machine Vision and Applications*, 2(3):141–153, June 1989.

[254] T. Strecker, J. van Beusekom, S. Albayrak, and T.M. Breuel. Automated ground truth data generation for newspaper document images. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 1275–1279, 2009.

[255] K. Subr, M. Majumder, and S. Irani. Greedy algorithm for local contrast enhancement of images. In *Proc. Int. Conf. Image Analysis and Processing*, pages 171–179, 2005.

[256] K. Summers. Near-wordless document structure classification. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 462–465, 1995.

[257] H.M. Sun. Page segmentation for manhattan and non-manhattan layout documents via selective CRLA. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 116–120, 2005.

[258] C. Sutton and A. McCallum. Collective segmentation and labeling of distant entities in information extraction. In *ICML Workshop Statistical Rel. Learning*, 2004.

[259] C. Sutton and A. McCallum. *Introduction to Relational Statistical Learning*, chapter An Introduction to Conditional Random Fields for Relational Learning, pages 93–128. MIT Press, 2007.

[260] C. Sutton, A. McCallum, and K. Rohanimanesh. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. *J. Mach. Learn. Res.*, 8:693–723, May 2007.

[261] T. Tajbakhsh and R.R. Grigat. Semiautomatic color checker detection in distorted images. In *Proc. Signal Processing, Patt. Recognition and Applications*, 2008.

[262] Y. Tang, H. Ma, X. Mao, D. Liu, and C. Suen. A new approach to document analysis based on modified fractal signature. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 567–570, 1995.

[263] B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *Proc. Conf. Uncertainty in Artificial Intelligence*, 2002.

[264] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In

Proc. Int. Conf Computer Vision, pages 839–846, 1 1998.

[265] A. Tonazzini, L. Bedini, and E. Salerno. Independent component analysis for document restoration. *Int. J. Document Analysis and Recognition*, 7:17–27, 2004.

[266] A. Tonazzini, I. Gerace, and F. Martinelli. Multichannel blind separation and deconvolution of images for document analysis. *Image Processing*, 19:912–925, 2010.

[267] A. Tonazzini, E. Salerno, and L. Bedini. Fast correction of bleed-through distortion in grayscale documents by a blind source separation technique. *Int. J. Document Analysis and Recognition*, 10:17–25, 2007.

[268] P.E. Trahanias, D. Karakos, and A.N. Venetsanopoulos. Directional processing of color images: theory and experimental results. *IEEE Trans. Image Processing*, 5(6):868–880, jun 1996.

[269] O. D. Trier and A. K. Jain. Goal-directed evaluation of binarization methods. *Pattern Analysis and Machine Intelligence*, 17(12):1191–1201, Dec 1995.

[270] O. D. Trier, A. K. Jain, and T. Taxt. Feature-extraction methods for character-recognition: A survey. *Pattern Recognition*, 29(4):641–662, April 1996.

[271] C.-M. Tsai and H.-J. Lee. Efficiently extracting and classifying objects for analyzing color documents. *Mach. Vision Appl.*, 22(1):1–19, 2011.

[272] W. H. Tsai and K. S. Fu. Subgraph error-correcting isomorphisms for syntactic pattern recognition. *Systems, Man and Cybernetics*, 13(1):48–62, 1983.

[273] S. Tsujimoto and H. Asada. Major components of a complete text reading system. *Proc. IEEE*, 80(7):1133–1149, 1992.

[274] M. Tuceryan and A.K. Jain. *The Handbook of Pattern Recognition and Computer Vision*, chapter Texture Analysis, pages 207–248. World Scientific Publishing Co., 1998.

[275] E. Ukkonen. Algorithms for approximate string matching. *Information and Control*, 64:100–118, 1985.

[276] J. van Beusekom, D. Keysers, F. Shafait, and T.M. Breuel. Example-based logical labeling of document title page images. In *Proc. Int. Conf. Document Analysis and Recognition*, volume 2, pages 919–923, 2007.

[277] J. van Beusekom, F. Shafait, and T.M. Breuel. Resolution independent skew and orientation detection for document images. In *Document Recognition and Retrieval*, pages 1–10, 2009.

[278] L. Vincent. Google book search: Document understanding on a massive scale. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 819–823, 2007.

[279] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 511–518, 2001.

[280] F. Wahl, K. Wong, and R. Casey. Block segmentation and text extraction in mixed text/image documents. *Computer Vision, Graphics, and Image Processing*, 20:375–390, 1982.

[281] F.M. Wahl and K.Y. Wong. An efficient method of running a constrained run length algorithm (CRLA) in vertical and horizontal directions on binary image data. Technical Report RJ3438, IBM Research Laboratory, San Jose, 1982.

[282] Y. Wang, R. Haralick, and I.T. Phillips. Zone content classification and its performance evaluation. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 540–544, 2001.

[283] Y. Wang, I.T. Phillips, and R.M. Haralick. Document zone content classification and its performance evaluation. *J. Pattern Recognition*, 39(1):57–73, 1 2006.

[284] G. Wisniewski and P. Gallinari. Relaxation labeling for selecting and exploiting

efficiently non-local dependencies in sequence labeling. In *Proc. Eur. Conf. Principles and Practice of Knowledge Discovery in Databases*, pages 312–323, 2007.

[285] C. Wolf, J. Jolion, and F. Chassaing. Text localization, enhancement and binarization in multimedia documents. In *Proc. Int. Conf. Pattern Recognition*, volume 4, pages 1037–1040, 2002.

[286] K.Y. Wong, R.G. Casey, and F.M. Wahl. Document analysis system. *IBM J. Research and Development*, 26:647–656, 1982.

[287] M. Wu, R. Li, B. Fu, W. Li, and Z. Xu. A model based book dewarping method to handle 2d images captured by a digital camera. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 158–162, 2007.

[288] S. Wu and A. Amin. Automatic thresholding of gray-level using multi-stage approach. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 493–497, 2003.

[289] Y. Wu, C. Bauckhage, and C. Thurau. The good, the bad, and the ugly: Predicting aesthetic image labels. In *Proc. Int. Conf. Pattern Recognitionscientific computing*, pages 1586–1589. IEEE, 2010.

[290] D. Wueller, H. van Dormolen, and V. Jansen. *Universal Test Target Technical Specification.* National Library of the Netherlands, 3 2011. accessed 09-Feb-2012.

[291] D. Xi and S.-W. Lee. Reference line extraction from form documents with complicated backgrounds. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 1080–1084, 2003.

[292] R. Xu and D. Wunsch. Survey of clustering algorithms. *IEEE Trans. Neural Networks*, 16(3):645–678, 2005.

[293] X. Xu, Z. Sun, B. Peng, X. Jin, and W. Y. Liu. An online composite graphics recognition approach based on matching of spatial relation graphs. *Int. J. Document Analysis and Recognition*, 7(1):44–55, March 2004.

[294] Y. Xue. Uniform color spaces based on ciecam02 and ipt color difference equation. Master's thesis, Rochester Institute of Technology, 11 2008.

[295] H. Yan. Skew correction of document images using interline cross-correlation. *CVGIP: Graphical Models and Image Processing*, 55:538–543, 1993.

[296] B.A. Yanikoglu and L. Vincent. Pink panther: a complete environment for groundtruthing and benchmarking document page segmentation. *Pattern Recognition*, 31(9):1191–1204, 1998.

[297] S.D. Yanowitz and A.M. Bruckstein. A new method for image segmentation. *Computer Vision, Graphics and Image Processing*, 46:82–95, 1989.

[298] B. Yu and A. Jain. A robust and fast skew detection algorithm for generic documents. *Pattern Recognition*, 29:1599–1629, 1996.

[299] B. Yu, X. Lin, Y. Wu, and B. Yuan. Isothetic polygon representation for contours. *CVGIP Image Understanding*, 56:264–268, 1992.

[300] L. Zhang, A.M. Yip, and C.L. Tan. Photometric and geometric restoration of document images using inpainting and shape-from-shading. In *Proc. Conf. Artificial Intelligence*, pages 1121–1126, 2007.

[301] Y. Zheng, H. Li, and D. Doermann. A model-based line detection algorithm in documents. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 44–48, 2003.

[302] Y. Zheng, C. Liu, X. Ding, and S. Pan. Form frame line detection with directional single-connected chain. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 699–703, 2001.

[303] H. Zhou and Z. Liu. Page frame segmentation for contextual advertising in print on demand books. In *Proc. Int. Workshop Comp. Vis. and Pattern Recognition*, pages

17–22, 2009.

[304] G. Zhu and D. Doermann. Automatic document logo detection. In *Proc. Int. Conf. Document Analysis and Recognition*, volume 2, pages 864–868, 2007.