

Institut für Kartographie und Geoinformation

Beschriftung von Bildschirmkarten in Echtzeit – Konzept und Struktur –

Inaugural-Dissertation

zur

Erlangung des Grades

Doktor-Ingenieur

(Dr.-Ing.)

der

Hohen Landwirtschaftlichen Fakultät

der

Rheinischen Friedrich-Wilhelms-Universität

zu Bonn

vorgelegt am 9. Juli 2003

von

Dipl.-Inform. Ingo Petzold

aus Bonn

Referent: Prof. Dr. Lutz Plümer
Korreferent: Prof. Dr. Dieter Morgenstern
Tag der mündlichen Prüfung: 21. November 2003
Gedruckt bei: _____

Zusammenfassung

Diese Arbeit stellt neuartige Konzepte zur Beschriftung von punkt-, linien- und flächenhaften Objekten in Karten vor. Der Fokus liegt dabei auf Bildschirmkarten. Im Unterschied zu traditionellen Papierkarten müssen für Bildschirmkarten die Funktionalitäten Zoomen (freie Maßstabswahl), Scrollen (Änderung des Kartenausschnitts) und Zusammenführen und Beschriften heterogener Daten (freie Kombination von Beschriftungslayern/-themen) unterstützt werden. Jede dieser Aktionen verursacht eine Neubeschriftung, die in Echtzeit erfolgen muss. Obwohl die Auflösung und Größe von Bildschirmkarten geringer ist als die von traditionellen Karten, ist die Beschriftung von Bildschirmkarten sehr aufwendig. Für die Informatik ist dieses Problem np-hard; das bedeutet, dass es vermutlich keinen wirklich schnellen Algorithmus gibt. Um trotzdem eine Beschriftung in Echtzeit zu erzielen, müssen Heuristiken eingesetzt werden. Gleichzeitig müssen kartographische Gestaltungsregeln in sehr schnellen Algorithmen integriert werden, um gut lesbare Bildschirmkarten zu erhalten.

Das Konzept zur Erreichung dieser Ziele integriert explizit kartographische Anforderungen, wie die Beschriftung in bevorzugter Position, die Minimierung des Informationsverlustes und die Berücksichtigung topologischer Trennlinien. Die Beschriftung in Echtzeit wird durch die Trennung in zwei Phasen erzielt. In die erste Phase, die vor einer Interaktion des Nutzers stattfindet, erfolgt die Verlagerung möglichst vieler zeitintensiver Berechnungen. Das führt zu einer Entlastung der zweiten Phase und somit zu einer schnelleren Beschriftung für den Nutzer. Die erste Phase wird als Präprocessingphase und die zweite Phase als Interaktionsphase bezeichnet.

Beide Phasen sind eng gekoppelt. Für die effiziente Unterstützung der Funktionen Scrollen und Zoomen erfolgt in der Präprocessingphase die Generierung einer Datenstruktur, auf die die Interaktionsphase aufbaut. Aus dieser maßstabsunabhängigen Datenstruktur, dem reaktiven Konfliktgraphen, werden passend zu jedem spezifischen Maßstab und Kartenausschnitt alle nötigen Informationen für den Beschriftungsprozess effizient abgeleitet.

Eine wichtige neuartige Heuristik, die für dieses Konzept entwickelt wurde, ist dabei das Kriterium zur Bestimmung der Beschriftungsschwierigkeit. Es ermöglicht eine Vorselektion der zu beschriftenden Objekte in Abhängigkeit von Maßstab und Objektpriorität in der Präprocessingphase. Dazu wurden verschiedene Kriterien entwickelt, analysiert und evaluiert.

Im Unterschied zu traditionellen, automatischen Beschriftungskonzepten ist in dem entwickelten Konzept die Beschriftung nicht auf vorgegebene diskrete Beschriftungspositionen beschränkt, sondern auf einen Beschriftungsraum, wie es den kartographischen Beschriftungsmethoden entspricht. Diese Modellierung wird mit „sliding labels“ bezeichnet.

Im Rahmen dieser Arbeit wurde ein komponentenbasiertes System entwickelt, das die wesentlichen Konzepte erfolgreich umsetzt und evaluiert. Die Umsetzung der Beschriftung punkthafter Objekte erfolgte nahezu vollständig. Die implementierte Visualisierungskomponente, die kartographisch angemessen Kartenobjekte darstellt, ermöglicht eine visuelle Verifikation der umgesetzten Verfahren, inklusive der Visualisierung der Konfliktgraphen. Eine Schnittstellenkomponente, die sich am OGC-Konzept der „Simple Features“ orientiert, ermöglicht den Datenimport von kommerziellen GIS.

Abstract

This work presents new concepts for labeling of point, line and area objects. The focus is on screen maps. In contrast to traditional paper maps, screen maps have to provide the functionalities of scrolling (changing the map clipping), zooming (free choice of scale), and integrating and labeling of heterogeneous data (free combination of labeling-layers/-themes). Each action causes a new labeling, which has to be performed in real-time. Although resolution and size of screen maps are lower than those of traditional maps, the labeling of screen maps is still very time-consuming. From a computer science perspective, this problem is NP-hard, which means that most likely there exists no fast algorithm. Nevertheless, to achieve a labeling in real time, heuristics must be used. Simultaneously, consideration of cartographic rules is essential in order to receive well readable screen maps.

To obtain these targets, the concept explicitly integrates cartographic rules, such as labeling in preferred position, minimization of information loss and consideration of topological separation lines. The labeling in real-time is achieved by dividing the labeling process into two phases. Calculations are shifted in the first phase as far as possible. Thus, the number of time consuming operations in the time critical second phase is reduced and leads to faster labeling for the user. The first phase is called the preprocessing-phase and the second phase is called the interaction-phase.

Both phases are linked closely. In order to efficiently support scrolling and zooming functions in the second phase, a data-structure is generated during the first phase. This scale independent data-structure is called reactive conflict graph and allows an efficient derivation of all information necessary for the labeling process for any scale and map clipping.

An essential new heuristic, which was developed for this concept, is a criterion to measure labeling difficulty. It allows a pre-selection of objects to be labeled in dependence of the object's scale and priority during the preprocessing-phase. Thus, different criteria are developed, analyzed and evaluated.

In contrast to traditional automatic labeling algorithms, the developed labeling concept is not restricted to discrete positions. Instead, sliding labels in labeling spaces are used, which satisfies cartographic principles.

The feasibility of the main concepts is demonstrated by the successful implementation and evaluation of a component-based system. The consideration of point labeling is realized completely. The implemented visualization component enables verification of the results of the labeling process, in particular by presenting the underlying reactive and static conflict graphs as well as the cartographic visualization of the map objects. An interface component provides data-import from commercial GIS, based on the OGC-concept of "Simple Features".

Danksagung

Ich möchte mich an dieser Stelle bei allen bedanken, die mir bei dieser Arbeit durch Diskussionen, Literatursuche, Korrekturlesen und moralische Unterstützung geholfen haben. Mein Dank gilt insbesondere Gerd, der sicher ein sehr guter Professor werden wird, und Ralf für ihre ewige Geduld, das nicht endende Korrekturlesen, Rummeckern, Diskutieren und natürlich auch für die große moralische Unterstützung ;-) Unterstützt haben mich ebenso Anne, Annette, Claudia, Doris, Ellen, Etienne, Germaine, Gundula, Jona, Micha, Rainer, Rolf, Stefan, Thomas, Viktor und meiner Familie.

Meinem Betreuer Prof. Dr. Lutz Plümer möchte ich mich für die vielen Anregungen, (harten) Diskussionen, konstruktiven Kritiken, seine intensive Betreuung bedanken, und das er mir immer wieder Freiräume zum Experimentieren eingeräumt hat. Durch die uneingeschränkte Nutzung von Ressourcen, wie Rechner, Drucker, Literatur, etc. hat er wesentlich zum Gelingen dieser Arbeit beigetragen. Mein Dank gilt natürlich nicht weniger Professor Dr. Dieter Morgenstern für die vielen produktiven Diskussionen, Anregungen und seine Unterstützung, sowohl inhaltlich, insbesondere bei kartographischen Problemstellungen, als auch für die Nutzung von Ressourcen des Instituts.

Der Deutschen Forschungsgemeinschaft gilt abschließend mein Dank für die Förderung des Kooperationsprojekts „Beschriftung von Bildschirmkarten in Echtzeit“, aus dem diese Arbeit hervorgegangen ist.

Finally, thanks to Caroline, Olivia and Sherwin for all their support and proofreading of all the English abstracts and papers.

Bonn, den 9.7.2003

Für Petra...

Inhaltsverzeichnis

1	Einleitung	1
2	Schrift in Bildschirmkarten – Aspekte der Kartographie und der Informatik.....	5
2.1	Kartographische Aspekte.....	5
2.1.1	Kartenobjekte.....	5
2.1.1.1	Schriftobjekte	6
2.1.1.2	Beschriftung punkthafter Objekte	6
2.1.1.3	Beschriftung linienhafter Objekte	7
2.1.1.4	Beschriftung flächenhafter Objekte.....	8
2.1.2	Maßstab und Darstellungsgrößen	9
2.1.3	Auswahl von zu beschriftenden Objekten	10
2.2	Begriffe und Methoden aus der Informatik	11
2.2.1	Komplexität und O-Notation	11
2.2.2	Datenstrukturen	13
2.2.2.1	Graph.....	13
2.2.2.2	Geometrische Datenstrukturen	13
2.2.3	Komplexität der überlappungsfreien Beschriftung.....	17
3	Computerbasierte Verfahren der Kartenbeschriftung.....	21
3.1	Vorgehensweise bei der automatischen Textplatzierung	21
3.2	Auswahlalgorithmen.....	22
3.3	Textplatzierungssysteme	25
4	Vorausberechnung von Konflikten.....	27
4.1	Konfliktgraph – Überblick und Idee.....	27
4.1.1	Verwendung des Konfliktgraphen.....	29
4.1.2	Historie 30	
4.1.3	Bezugssystem	31
4.2	Definition des reaktiven Konfliktgraphen.....	32

4.3	Generierung des reaktiven Konfliktgraphen	33
4.3.1	Punkthafte Objekte	33
4.3.1.1	Beschriftungsraum.....	33
4.3.2	Beschriftungskonflikt: punkthafte – punkthafte Objekt	38
4.3.2.1	Schnittmaßstab	39
4.3.2.2	Deselektionsmaßstab	40
4.3.3	Beschriftungsschwierigkeit und Deselektion: punkthafte Objekt.....	42
4.3.3.1	Knotengrad – Anzahl der Konflikte	43
4.3.3.2	Größter freier Winkel	48
4.3.3.3	Freie Beschriftungsfläche.....	52
4.3.3.4	Zusammenfassung – Verfahren zur Bestimmung der Beschriftungsschwierigkeit	69
4.3.4	Linienhafte Objekte	72
4.3.4.1	Beschriftungsraum.....	73
4.3.4.2	Segmentierung des Beschriftungsraums.....	75
4.3.5	Beschriftungskonflikt: linienhaftes – linienhaftes Objekt	77
4.3.5.1	Schnittmaßstab	78
4.3.5.2	Deselektionsmaßstab	79
4.3.6	Beschriftungsschwierigkeit und Deselektion: linienhaftes Objekt.....	79
4.3.7	Beschriftungskonflikt: punkthafte – linienhaftes Objekt	82
4.3.7.1	Schnittmaßstab	83
4.3.7.2	Deselektionsmaßstab	91
4.3.8	Erweiterung Beschriftungsschwierigkeit: punkthafte Objekt.....	91
4.3.9	Flächenhafte Objekte	92
4.3.10	Beschriftung von Siedlungsflächen	95
4.3.11	Erzeugung des reaktiven Konfliktgraphen	96
4.3.11.1	Laufzeitanalyse.....	100
4.4	Speicherung des reaktiven Konfliktgraphen	101
4.5	Geometrietypwechsel und Generalisierung – Maßstabsunabhängigkeit des reaktiven Konfliktgraphen.....	103
4.6	Integration von Informationsverlusten in den Konfliktgraphen.....	105
4.7	Topologisch korrekte Lage von Schrift zum zu beschriftenden Objekt.....	107
4.8	Zusammenfassung	109

5 Vermeidung von Konflikten	111
5.1 Ableitung des statischen aus dem reaktiven Konfliktgraphen	112
5.1.1 Laufzeitanalyse	115
5.2 Auswahlalgorithmus	116
5.2.1 Platzierung der Beschriftung punkthafter Objekte	120
5.2.2 Platzierung der Beschriftung linienhafter Objekte	126
5.2.3 Nachbehandlung nicht beschrifteter Objekte	127
5.2.4 Laufzeitanalyse	129
5.3 Feinplatzierung	132
5.3.1 Topologisch korrekte Lage von Schrift zum zu beschriftenden Objekt	132
5.3.2 Berücksichtigung von Informationsverlusten	134
5.4 Zusammenführung von unterschiedlichen Beschriftungen des gleichen Kartenausschnitts	137
6 Komponenten des Prototypen	141
6.1 Überblick	141
6.2 Offene Schnittstellenkomponente	144
6.3 Klassifizierung und Signaturzuordnung	145
6.4 Visualisierungskomponente	145
6.5 Effiziente geometrische Zugriffsstrukturen – R-Baum	146
6.6 Vorausberechnung von Konflikten – Konfliktgraphen	146
6.7 Vermeidung von Konflikten – Beschriftung	148
7 Resümee und Ausblick	151
Glossar	155
Literaturverzeichnis	159
Index	165

Kapitel 1

Einleitung

Geoinformationssysteme (GIS) finden in der Wissenschaft und in der Praxis eine immer größere Verbreitung. Sie werden in den Geowissenschaften genutzt und zur Routenplanung, zur Navigation und zur Unterstützung von politischen, ökonomischen und ökologischen Entscheidungen zu Hilfe genommen. In allen Einsatzgebieten kommt der Visualisierung eine hohe Bedeutung zu.

Bildschirmkarten sind die visuelle Schnittstelle zwischen dem GIS und dem Nutzer. Sie entstehen als Visualisierung von Antworten auf vorausgegangene Anfragen an GIS und dienen dazu, die Generierung spezifischerer Anfragen zu unterstützen. Sie unterstützen den navigierenden Zugriff auf die Datenbasis eines GIS. Im Kontext interaktiver GIS geht es darum, informationsreiche Bildschirmkarten in einer hohen kartographischen Qualität in Echtzeit zu erzeugen. Die zunehmende Bedeutung von Bildschirmkarten wird überdies daran deutlich, dass in den letzten Jahren vermehrt Tendenzen zu erkennen sind, traditionelle, statische Papierkarten durch Bildschirmkarten zu verdrängen. Die qualitativen Erwartungen, die an eine Bildschirmkarte gestellt werden, orientieren sich an denen von traditionellen Papierkarten.

Eine exakte Umsetzung von Papierkarten auf Bildschirmkarten ist jedoch auf Grund der verschiedenartigen Medien und den neu hinzukommenden Erwartungen des Nutzers nur bedingt möglich. Im Unterschied zu Papierkarten haben Bildschirmkarten eine geringere Auflösung, was zu einer Erhöhung der Mindestgröße für Grafik und Schrift führt. Gleichzeitig ist die Kartenfläche durch die Bildschirmgröße eingeschränkt. Dies verschärft die Konkurrenz zwischen Kartengrafik und Kartenschrift.

Die Bildschirmkarte ist für den Nutzer nicht nur Informationsquelle, sondern auch Interaktionsmedium mit dem GIS. Er kann sich von einer unspezifischen Anfrage schrittweisen Zugriff auf die von ihm gewünschten Informationen verschaffen. Zwei Eigenschaften sind dabei wesentliche Voraussetzung. Der Nutzer muss die Informationen der Bildschirmkarte in kurzer Zeit erschließen können. Diese kognitiven und ergonomischen Anforderungen werden durch die oben genannte Adaption traditioneller kartographischer Kartengestaltungsregeln berücksichtigt. Des Weiteren ist eine schnelle Erzeugung von Bildschirmkarten notwendig, so dass der Nutzer keine oder nur sehr geringe Wartezeiten hat. Die Zeit, die der Rechner insgesamt für die Generierung einer Karte benötigt, sollte deshalb nicht wesentlich höher sein als die vom Anwender für die Aufnahme und Verarbeitung dieser Informationen benötigte Zeit. Lange Antwortzeiten ermüden den Anwender. Jedoch darf diese Antwortzeit nicht auf Kosten der Qualität und der Informationsdichte verkürzt werden: je mehr relevante Objekte die einzelne Karte enthält, desto weniger Anfragen benötigt der Anwender, um zum gewünschten Ergebnis vorzudringen. Daher sollte die Generierung einer Bildschirmkarte in

Echtzeit erfolgen, was gleichzeitig ein wesentliches zusätzliches Qualitätsmerkmal des GIS darstellt.

Zum Ausgleich der technischen Beschränkungen und zur Unterstützung der Interaktion sollten Bildschirmkarten die Funktionen *Zoomen* (freie Maßstabswahl), *Scrollen* (Änderung des Kartenausschnitts) und *Zusammenführen von thematischen Karten* unterstützen.

Durch die technischen Beschränkungen ist die Zahl der darzustellenden Objekte bei Bildschirmkarten im Vergleich zur klassischen Karte klein. Dies erhöht die Bedeutung der Schrift als Informationsträger, zumal sie gleichzeitig zur Orientierung dient. Schrift unterstützt außerdem die Differenzierung von Kartenobjekten durch ihren darzustellenden Text und kann durch Wahl von Schriftgröße, -form und -farbe zusätzlich qualitative und quantitative Angaben vermitteln. Durch die korrekte Lage kann die Schrift bei Flüssen und Grenzen darüber hinaus eine zusätzliche topologische Information über die Lage des zu bezeichnenden Objekts ausdrücken.

Als wichtige Erkenntnisquelle für die Kartenbeschriftung können die von Imhof aufgestellten Regeln [IMH62] dienen. Diese sind jedoch keine formalen Regeln im Sinne der Informatik. Imhof beschreibt die richtige Vorgehensweise viel mehr kasuistisch. Dabei wird zwischen der Beschriftung von punkt-, linien- und flächenhaften Objekten sowie der von Siedlungsflächen unterschieden. Im Gegensatz zu Signaturen haben Schriftobjekte keine feste geometrische Lage. Sie besitzen in Abhängigkeit der Beschriftungsklasse des zu beschriftenden Objekts Freiheitsgrade bei der räumlichen Platzierung.

Neben der Berücksichtigung der Beschriftungsregeln zwischen dem zu beschriftenden Objekt und der zugehörigen Beschriftung existieren weitere Regeln, die die Kombination der Schriftobjekte untereinander und mit Kartenobjekten betreffen. Ein wichtiges Ziel bei der Schriftplatzierung ist die Vermeidung von Überlappungen zwischen Schriftobjekten, da dies zu Unlesbarkeit führt. Existiert keine überlappungsfreie Beschriftung, müssen Schriftobjekte in Abhängigkeit der Wichtigkeit der zugehörigen zu beschriftenden Objekte und ihres Konfliktpotentials weggelassen werden. In diesem Kontext stellt sich die Frage nach der Beschriftungsschwierigkeit von zu beschriftenden Objekten. Im Unterschied zur traditionellen Papierkarte ist bei Bildschirmkarten der Zwang, alles Wesentliche darzustellen, durch die temporäre Lebenszeit und den Einsatz multimedialer Effekte vermindert.

Die Erzeugung überlappungsfreier Beschriftungen ist aus Sicht der Informatik ein komplexes kombinatorisches Problem. Bei manueller Beschriftung entfällt ca. die Hälfte der Zeit zur Erstellung einer Karte auf die Platzierung; es werden 20 bis 30 Schriftobjekte pro Stunde von einem Kartographen platziert [COO88]. Das deutet auf die enorme Zeitressource hin, die auch die automatische Beschriftung benötigt. Im Kontext von Bildschirmkarten entsteht dabei der Zielkonflikt zwischen dem Wunsch nach Beschriftung in Echtzeit und der Forderung nach hoher Qualität. Verschärfend kommt bei Bildschirmkarten die Unterstützung der freien Maßstabswahl hinzu.

Für die automatische Beschriftung von Bildschirmkarten in Echtzeit ergeben sich folgende Ziele:

- Die Entwicklung von Algorithmen und Datenstrukturen für die Beschriftung in Echtzeit, die die Funktionen Zoomen, Scrollen und Zusammenführen unterstützen.
- Die systematische Kopplung von Beschriften und Weglassen zur Erzielung einer hohen Informationsdichte.
- Die Adaption und Integration kartographischer Gestaltungsregeln für die Lösung des Zielkonflikts zwischen hoher Informationsdichte und guter Lesbarkeit.

Zur Erreichung dieser Zielsetzung wird in dieser Arbeit ein Verfahren entwickelt, das in zwei Phasen geteilt ist. Aufgabe der ersten Phase ist der Aufbau einer reaktiven Datenstruktur, die zur Laufzeit – in der zweiten Phase – genutzt wird. Kapiteln 4 kann der ersten Phase und Kapitel 5 der zweiten zugeordnet werden. Kapitel 6 beschreibt ein komponentenbasiertes System, das die wesentlichen entwickelten Konzepte erfolgreich umsetzt und evaluiert.

Wesentliche Grundlagen, die zum Verständnis der entwickelten Konzepte erforderlich sind, werden in Kapitel 2 beschrieben. Kapitel 3 gibt einen kurzen Überblick über bisher existierende, automatische Kartenbeschriftungsverfahren. Die Arbeit schließt mit einem Resümee und Ausblick.

Diese Arbeit ist im Rahmen des Kooperationsprojekts „Beschriftung von Bildschirmkarten in Echtzeit“ von Prof. Dr. Lutz Plümer und Prof. Dr. Dieter Morgenstern entstanden, das von der Deutschen Forschungsgemeinschaft zwei Jahre gefördert wurde.

Kapitel 2

Schrift in Bildschirmkarten – Aspekte der Kartographie und der Informatik

Dieses Kapitel behandelt die Grundlagen aus der Kartographie und der Informatik, die für das Verständnis der weiteren Arbeit, insbesondere für Kapitel 4 und 5, nötig sind.

2.1 Kartographische Aspekte

Eine ausführliche und umfassende Darstellung kartographischer Grundlagen, der traditionellen Herstellung von Karten, der kartographischen Modellierung und der Generalisierung können in [HGM02] gefunden werden. Die Auswirkungen, Anpassungen und Anforderungen an Bildschirmkarten werden in [LUT97] und [NEU01] beschrieben.

Die Beschriftung von Karten behandelt das Standardwerk [IMH62]. Es werden Regeln beschrieben, wie Karten zu beschriften sind und durch Gegenüberstellung von guten und schlechten Beschriftungen erläutert. Eine Formalisierung der Kartenbeschriftung, die Erzeugung und Bewertung von Beschriftungspositionen auf Basis vorhandener Karten, werden detailliert in [ELL01] behandelt.

2.1.1 Kartenobjekte

Karten bestehen aus den Grundelementen Punkt, Linie und Fläche sowie den zusammengesetzten Zeichen Signatur, Diagramm, Halbton und Schrift [HGM02]. Aus Sicht der Schriftplatzierung und dem in den folgenden Kapiteln entwickelten Verfahren kann die Granularität der Betrachtung in Abhängigkeit der Geometrie auf punkthafte, linienhafte und flächenhafte Objekte sowie zu platzierende Objekte vergrößert werden, siehe die nachfolgenden Abschnitte (vgl. [IMH62]). Es tritt dabei insoweit kein Informationsverlust auf, da die Objekte alle relevanten Informationen als Attribute mit sich führen. Die Visualisierung der Objekte, die keine Schrift sind, wird im Folgenden nicht weiter behandelt und als gegeben vorausgesetzt. Sie können u.a. in [HGM02], [LUT97] und [NEU01] nachgelesen werden.

Eine Teilmenge der punkt-, linien- und flächenhaften Objekte sind die zu beschriftenden Objekte. Diese sind vorgegeben bzw. aus den Attributen der Objekte ableitbar, genauso wie der zu platzierende Text oder das Diagramm. Die in dieser Arbeit entwickelten Verfahren sind auf die Platzierung von Text bezogen, können jedoch auch für die Platzierung von Diagrammen verallgemeinert werden. Innerhalb der Klassen der punkt-, linien- und flächenhaften Objekte wird, wie in den nachstehenden Abschnitten beschrieben, eine weitere Unter-

teilung der Beschriftung vorgenommen, bevor auf die Abhängigkeiten des Maßstabs eingegangen wird.

Neben der Vermeidung von Schriftüberlappung, die das zentrale Thema dieser Arbeit ist, muss der Informationsverlust, der durch Verdeckung von nicht Schriftobjekten durch Schriftobjekte entsteht, wenn möglich vermieden sonst klein gehalten werden. Dies wird in Abschnitt 4.6 und insbesondere 5.3.2 beschrieben und behandelt. Außerdem muss die Schrift auf der topologisch korrekten Seite einer Trennlinie platziert werden, wenn vorhanden (4.7 und detailliert in 5.3.1).

2.1.1.1 Schriftobjekte

Das zu platzierende Schriftobjekt wird in dieser Arbeit durch eine Labelbox approximiert, bevor es bei der Visualisierung der Karte wieder durch das Schriftobjekt ersetzt wird. Dabei bieten sich verschiedene Approximationen an (Abbildung 1), die sich hinsichtlich der Geometrie unterscheiden. In dieser Arbeit wird die in Abbildung 1 b) dargestellte Approximation verwendet (s. Abschnitt 4.3.1.1.1), da sie eine geometrisch einfache Form ist und dadurch Geschwindigkeitsvorteile bei der Berechnung erzielt, vgl. Kapitel 4 und 5. Dies vereinfacht die Platzierung, stellt jedoch keine Einschränkung dar.

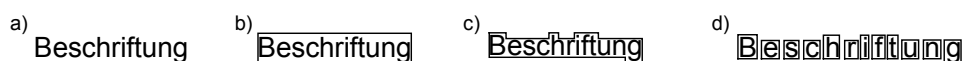


Abbildung 1 Beschriftungsobjekt (a)) und mögliche Approximationen durch Labelboxen (b) bis d)).

Der hier vorgestellte Ansatz ist für beliebige Schriftart, -höhe, -farbe, Sperrung etc. geeignet. Daher werden diese hier nicht weiter behandelt sondern als bekannt vorausgesetzt. [MUS93], [IMH62], [FRE10] gehen vertiefend auf dieses Thema ein.

2.1.1.2 Beschriftung punkthafter Objekte

In dieser Arbeit werden Objekte mit kleiner flächenhafter Ausdehnung als punkthaft bezeichnet. Die Beschriftung dieser Objekte erfolgt achsenparallel in einem vorgegebenen Abstand um sie herum. Dazu kann eine lokale Bewertung zwischen zu beschriftendem Objekt und Schrift angegeben werden (Abbildung 2 a) und b)). Diese in vielen Arbeiten ([ECM96], [PET96] und [ELL01]) quantifizierten Aussagen sind aus [IMH62] abgeleitet.

Daneben gibt es weitere spezielle Beschriftungsarten für punkthafte Objekte, die hier aber nicht explizit behandelt werden, da das in dieser Arbeit entwickelte Verfahren sie nur kurz streift: Abschnitt 4.3.1.1.2 geht kurz auf die Beschriftung von Gipfeln und Küstenorte ein. Der Ansatz eines Verfahrens für die Beschriftung von Siedlungsflächen wird in Abschnitt 4.3.10 dargestellt.

Wie oben angedeutet und in Abbildung 2 a) und b) dargestellt, sind die meisten Arbeiten im Bereich der automatischen Textplatzierung auf diskrete Beschriftungspositionen eingeschränkt. Das in dieser Arbeit entwickelte Verfahren ist allgemeiner und lässt ein Kontinuum an Beschriftungspositionen zu. Dazu wird ausgenutzt, dass alle Beschriftungspositionen innerhalb eines Beschriftungsraums liegen. Dieser kann, wie in Abbildung 2 a) und b) dargestellt, durch ein achsenparalleles Rechteck approximiert werden. Eine formale Definition erfolgt in 4.3.1.1.

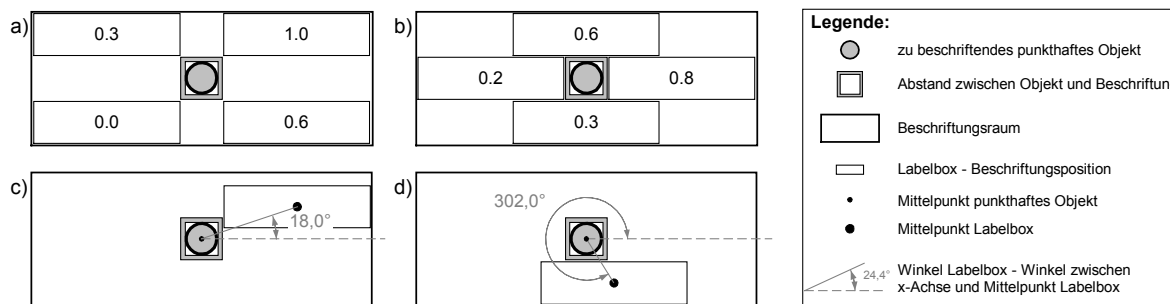


Abbildung 2 Lokale Bewertung von Beschriftungspositionen punkthafter Objekte in Abhängigkeit von ihrer Lage. In a) und b) sind diskrete Beschriftungspositionen mit Bewertungen von 0.0 (schlecht) bis 1.0 (sehr gut) dargestellt. Bewertungen von nicht diskreten (kontinuierlichen) Beschriftungspositionen zeigen c) und d).

Für jede Beschriftungsposition kann der Winkel zwischen der positiven x-Achse, dem Mittelpunkt des zu beschriftenden Objekts und dem Mittelpunkt der Beschriftungsposition bestimmt werden (Abbildung 2 c) und d)) [ELL01]. Über den Winkel kann die Beschriftungsposition bewertet werden. Dazu werden die Bewertungen von diskreten Beschriftungspositionen, die die Extrema bilden, in eine Funktion umgesetzt und zwischen ihnen interpoliert:

$$\text{Beschriftungspositionsbewertung}(\text{Winkel}) = \begin{cases} 1.0 & : \text{Winkel} = 25^\circ \\ 0.3 & : \text{Winkel} = 90^\circ \\ 0.7 & : \text{Winkel} = 155^\circ \\ 0.0 & : \text{Winkel} = 205^\circ \\ \text{interpoliere} & : \text{sonst} \end{cases} \quad (2.1)$$

Dies lässt sich wie folgt präzisieren:

$$\text{Beschriftungspositionsbewertung}(\text{Winkel}) = \begin{cases} 0,00556x - 0,86111 & : 0^\circ \leq \text{Winkel} < 25^\circ \\ -0,01077x + 1,26923 & : 25^\circ \leq \text{Winkel} < 90^\circ \\ 0,00615x - 0,253846 & : 90^\circ \leq \text{Winkel} < 155^\circ \\ -0,014x + 2,87 & : 155^\circ \leq \text{Winkel} < 205^\circ \\ 0,00556x - 1,138889 & : 205^\circ \leq \text{Winkel} \leq 360^\circ \end{cases} \quad (2.2)$$

2.1.1.3 Beschriftung linienhafter Objekte

Die Beschriftung linienhafter Objekte erfolgt in einem kleinen Abstand entlang des zu beschriftenden Objekts (Abbildung 3 a)). Für die Beschriftungsposition kann dabei keine einfache und eindeutige Bewertung analog zu der für punkthafte Objekte angegeben werden, da sie sich aus mehreren gewichteten Kriterien zusammensetzt, wie am Ende des Abschnitts dargestellt. Die Beschriftung linienhafter Objekte ist detailliert in [IMH62] und [ELL01] dargestellt und wird hier nicht weiter vertieft.

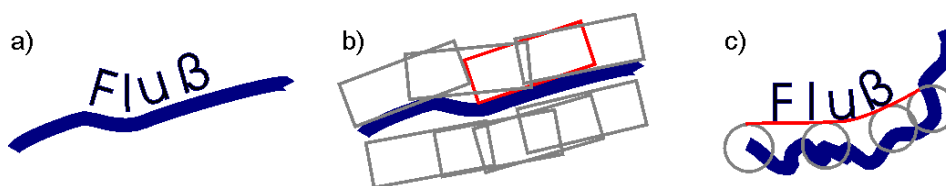


Abbildung 3 Beschriftung linienhafter Objekte a). In b) ist das Verfahren von [ECM96] dargestellt; die ausgewählte Beschriftungsposition ist rot dargestellt. c) zeigt das Verfahren von [ELL01] und [KNI98] (rote Standlinie).

Existierende Verfahren für die Beschriftung linienhafter Objekte werden u.a. in [ECM96], [KNI98] und [ELL01] behandelt. Das Verfahren aus [ECM96] platziert in einem festen Abstand Beschriftungen entlang des ganzen zu beschriftenden Objekts und wählt anschließend die aus, die die beste Bewertung hat (Abbildung 3 b)). Die Verfahren von [KNI98] und [ELL01] sind komplexer und erzielen bessere Beschriftungsergebnisse, da sie Splines (Kurvensegmente) verwenden und so ästhetische Aspekte berücksichtigen. Dadurch werden Knicke vermieden und ein harmonischer Verlauf der Schrift sichergestellt (Abbildung 3 c)).

Im Verlauf dieser Arbeit wird zunächst in der Präprocessingphase, Kapitel 4, analog zu der Beschriftung punkthafter Objekte, von einem Beschriftungsraum ausgegangen. Die Beschriftung linienhafter Objekte erfolgt parallel zum Objekt. Der Beschriftungsraum bildet einen Schlauch um das Objekt, in dem das zu beschriftende linienhafte Objekt, der Abstand der Schrift zum Objekt sowie die Menge der möglichen Beschriftungspositionen, d.h. die Höhe der Schrift plus einem Sicherheitsraum für die Knicke, liegt. Dies wird in Abschnitt 4.3.4.1 formalisiert und behandelt.

Um dem Titel der Arbeit gerecht zu werden und in Abschnitt 5.2.2 beschrieben ist, muss die Beschriftung schnell erfolgen. Daher erfolgt in der Interaktionsphase in Kapitel 5 zunächst eine schnelle d.h. einfache Beschriftung innerhalb des Beschriftungsraums, die, bei ausreichend verbleibender Zeit, durch Verfahren von [ELL01] und [KNI98] nachgebessert werden kann.

Um eine Bewertung für die Beschriftungspositionen linienhafter Objekte analog zu der von punkthaftern Objekten zu erhalten, kann die in [ECM96] entwickelte Bewertungsmethode, die sich aus mehreren gewichteten Bewertungsfunktionen zusammensetzt, verwendet werden. Einige exemplarische Bewertungsfunktionen sind:

- Zentrierung: Verhältnis der Entfernung zwischen Labelbox und Anfang des linienhaften Objekts zur gesamten Länge des Objekts:

$$centerness = 1 - |2 * distance(labelbox, start\ of\ lineObject) / length(lineObject) - 1|$$

- Flachheit: Die eingeschlossene Fläche zwischen Labelbox und dem linienhaften Objekt:

$$flatness = 1 / \text{Fläche zwischen Labelbox und linienhaften Objekt}$$

Die Bewertung erstreckt sich von 0.0 (schlecht) bis 1.0 (sehr gut), analog zu der der Beschriftung punkthafter Objekte.

2.1.1.4 Beschriftung flächenhafter Objekte

Bei flächenhaften Objekten können zwei Fälle unterschieden werden: Im ersten Fall haben die Objekte eine Grafik, die ihre geometrische Begrenzung festlegt. Dann kann die Beschriftung horizontal erfolgen (Abbildung 4 a)) oder die Geometrie des zu beschriftenden Objekts widerspiegeln. Im zweiten Fall ist die Geometrie der Objekte nicht durch Grafik repräsentiert. Dann wird die Geometrie ausschließlich durch Schrift repräsentiert (Abbildung 4 b)).

In [IMH62] und insbesondere [ELL01] kann mehr zu dieser Thematik gefunden werden. Verfahren zur Beschriftung flächenhafter Objekte, die die Geometrie des Objekts widerspiegeln, werden in [PET96] und [PEP97] behandelt und die achsenparallele Beschriftung wird detailliert in [DOE01] und [DPP03] vorgestellt.

Die Anwendung des in dieser Arbeit verwendeten Konzepts des Beschriftungsraums gestaltet sich bei flächenhaften Objekten schwierig, da der Beschriftungsraum gleich der zu beschriftenden Fläche ist und er somit Konflikte mit den Beschriftungsräumen aller zu beschriftender Objekte innerhalb der Fläche hat (vgl. Kapitel 4). In Abschnitt 4.3.9 werden daher kurz Ver-

fahren skizziert, die zur Beschriftung flächenhafter Objekte die Methodik zur Beschriftung linienhafter Objekte anwendet.

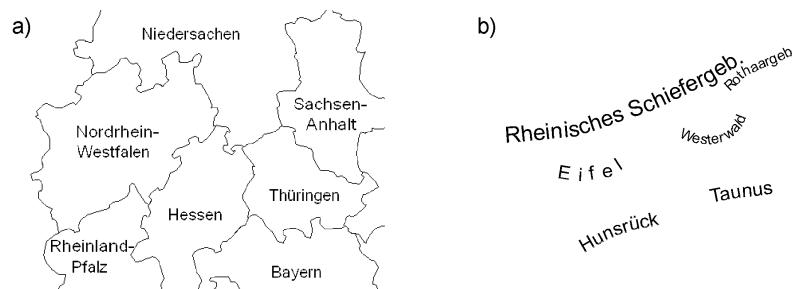


Abbildung 4 Beschriftung flächenhafter Objekte mit a) und ohne Begrenzung b): In a) ist eine achsenparallele Beschriftung dargestellt und in b) spiegelt die Schrift die Geometrie des zu beschriftenden Objekts wider.

2.1.2 Maßstab und Darstellungsgrößen

Schrift- und Signaturgrößen sind abhängig von Mindestgrößen, visueller Unterscheidung von Klassen, Komplexität der Grafiken usw. und – nach kartographischer Theorie – maßstabsunabhängig. Abweichend von dieser Theorie kann in manchen traditionellen Karten eine Verkleinerung dieser Objekte, bei einer Verkleinerung des Maßstabs, oder eine Vergrößerung dieser Objekte, bei einer Vergrößerung des Maßstabs, festgestellt werden.

In [HEB98] werden empirisch Karten gleichen Typs in unterschiedlichen Maßstäben untersucht, die Größenänderungen ermittelt und in einer Funktion formalisiert. Die Änderung der Größe eines Objekts im Referenzmaßstab $fixScale$ ($objectWidth_{fixScale}$, $objectHeight_{fixScale}$) für die Darstellung im Zielmaßstab $scale$ ($objectWidth_{scale}$, $objectHeight_{scale}$) lässt sich wie folgt bestimmen:

$$objectWidth_{scale} = objectWidth_{fixScale} \left(\frac{scale}{fixScale} \right)^{value} \quad (2.3)$$

$$objectHeight_{scale} = objectHeight_{fixScale} \left(\frac{scale}{fixScale} \right)^{value} \quad (2.4)$$

dabei ist $value$ ein Wert aus dem Intervall $[0..0,15]$ mit dem die Vergrößerung bzw. Verkleinerung justiert werden kann. Soll keine Vergrößerung oder Verkleinerung erfolgen, also der kartographischen Theorie Rechnung getragen werden, wird der Wert $value$ auf Null gesetzt. Durch den Einsatz dieser Funktion in dem in den Kapiteln 3 und 4 entwickelten Konzept, kann eine Universalität erreicht werden.

Maßstabsänderung und Auswirkung auf Schrift und Signatur sind in Abbildung 5 dargestellt. Die Ausgangskarte (Referenz) mit Signaturierung und Beschriftung in der für den Maßstab angemessenen Größe ist in Abbildung 5 c) dargestellt. Erfolgt eine „einfache“, d.h. fotografische, Verkleinerung oder Vergrößerung der Karte mit Schrift und Signatur ist das Ergebnis eine „schlechte“ Karte b) und d). Beschriftete und signaturierte Karten mit der in [HEB98] entwickelten Skalierung sind in Abbildung 5 a) und e) dargestellt. Die Schrift ist etwas kleiner bzw. größer als die Referenzbeschriftung in c).

Da Bildschirmkarten nicht auf einen festen Kartentyp, wie z.B. die Topographische Karte 1:50.000, eingeschränkt sind und der Darstellungsmaßstab frei wählbar ist, existieren keine expliziten Zeichenvorschriften, sondern nur Mindestgrößen, die auch bei kleinen Maßstäben einzuhalten sind [HGM02], [MAL98].

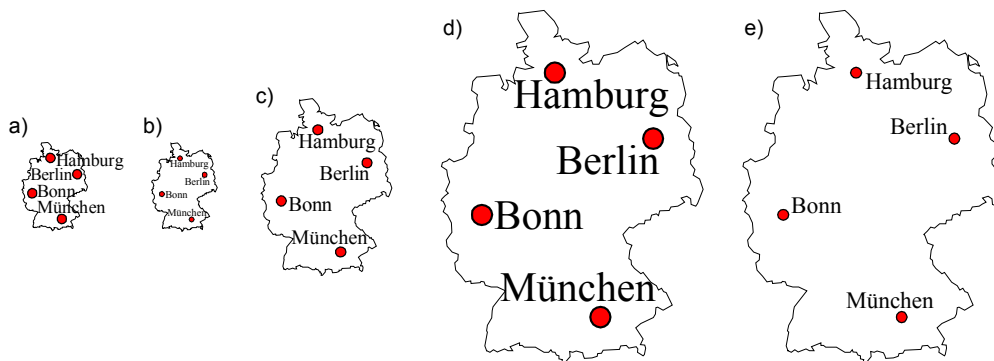


Abbildung 5 Maßstabsänderung und ihre Auswirkung auf Schrift und Signatur: Ausgangskarte (Referenz) c); fotografische Verkleinerung und Vergrößerung b) und d); Anwendung der in [HEB98] entwickelten Skalierung a) und e).

Bei der Verwendung der von [HEB98] entwickelten Funktion mit einem *value*-Wert größer Null kann es zu einer Unterschreitung der Mindestgrößen kommen. Um diese Mindestgrößen in dem in Kapitel 4 entwickelten Konzept des reaktiven Konfliktgraphen, in dem die von [HEB98] entwickelte Funktion integriert ist, für *value*-Werte größer Null sicherzustellen, erhält jeder reaktive Konfliktgraph einen ausgezeichneten Maßstab. Dies ist der Maßstab, ab und unterhalb dem mindestens ein Objekt die Mindestgröße unterschreitet und somit bildet er die untere Grenze für den Maßstabsbereich, in dem der reaktive Konfliktgraph verwendbar ist. Für kleinere Maßstäbe muss folglich ein zusätzlicher reaktiver Konfliktgraph generiert werden. Dies kann auf zwei Arten erfolgen: die Objektgröße wird im gleichen Referenzmaßstab vergrößert oder durch eine entsprechende Wahl der Objektgröße, des Referenzmaßstabs und des Setzen des *value*-Wertes auf Null wird ein weiteres Verkleinern der Objekte unterhalb des ausgezeichneten Maßstabs verhindert. Diese Einschränkung des reaktiven Konfliktgraphen hinsichtlich der Berücksichtigung von Mindestgrößen wird im Folgenden als bekannt vorausgesetzt und nicht weiter erwähnt. Die Behandlung reaktiver Konfliktgraphen, die für ein vorgegebenes Maßstabsintervall gültig sind, wird in Abschnitt 4.5 im Kontext von Geometriypwechsel und Generalisierung behandelt.

2.1.3 Auswahl von zu beschriftenden Objekten

Im letzten Abschnitt wird die Skalierung von Schrift in Abhängigkeit vom Maßstab diskutiert und in Abbildung 5 dargestellt. Bei dem Vergleich der Abbildung 5 a), c) und e), in denen eine dem Maßstab angemessene Schriftgröße gewählt wurde, fällt auf, dass die Karte in a) überladen mit Schriftobjekten ist, die Anzahl der Schriftobjekte in Karte c) angemessen ist und die Karte in e) zu wenige Schriftobjekte hat. Es muss folglich eine angemessene Auswahl an zu platzierenden Schriftobjekten bzw. an zu beschriftenden Objekten getroffen werden. Dieses geschieht in Abhängigkeit vom Maßstab der Karte und der zum Maßstab zugehörigen Schriftobjektgröße zu einem Referenzmaßstab.

In [PIT64] wird ein Auswahlgesetz beschrieben, dass eine Aussage über die Anzahl der darzustellenden Objekte für einen (Folge-)Maßstab in Abhängigkeit eines Ausgangsmaßstabs macht. Zum Nachweis werden Karten von Atlanten in verschiedenen Maßstäben hinsichtlich ihrer Anzahl an Bahnlinien, Inseln, Siedlungen, Wäldern, etc. auf den Einklang mit dem entwickelten Auswahlgesetz erfolgreich untersucht.

Eine Erweiterung auf die Auswahl von Schriftobjekten wird ebenfalls in diesem Artikel behandelt. Die Anzahl der Schriftobjekte im Folgemaßstab wird dabei in Abhängigkeit der

Anzahl der Objekte im Ausgangsmaßstab bestimmt. Zusätzlich fließen die Schriftflächen im Ausgangs- und Folgemaßstab ein:

$$n_F = n_A \frac{f_A}{f_F} \left(\frac{M_A}{M_F} \right)^2 \quad (2.5)$$

n_A ≡ Anzahl der Objekte im Ausgangsmaßstab

n_F ≡ Anzahl der Objekte im Folgemaßstab

M_A ≡ Maßstabszahl des Ausgangsmaßstabs

M_F ≡ Maßstabszahl des Folgemaßstabs

f_A ≡ Durchschnittsfläche eines Buchstaben im Ausgangsmaßstab f_F ≡ Durchschnittsfläche eines Buchstaben im Folgemaßstab

„[Das Auswahlgesetz] stellt die quantitative Seite der Generalisierung auf eine gesetzmäßige Basis, indem es für die Objektauswahl und für die Formvereinfachung exakte Zahlenwerte vermittelt“ [PIT64]. Es wird dabei keine Aussage über die Auswahl konkreter Objekte getroffen sondern eine rein normative. Die für die Schriftplatzierung notwendige Berücksichtigung der geometrischen Verteilung der zu platzierenden Schriftobjekte bzw. der zu beschriftenden Objekte, um eine überlappungsfreie Beschriftung zu erzielen, erfolgt nicht. Das Auswahlgesetz ist somit nur zur Verifikation von Karten geeignet.

Andere automatische Verfahren, die zu einem Maßstab die zu platzierenden Schriftobjekte bzw. die zu beschriftenden Objekte liefert, sind nicht bekannt. Zu festen Maßstäben, wie den topographischen Karten, existieren Musterblätter, die jedoch nur bedingt für Bildschirmkarten mit freier Maßstabswahl geeignet sind. Daher wird in Kapitel 4 und 5 ein neues Verfahren entwickelt, das auf Basis von Beschriftungsschwierigkeiten entscheidet welche Objekte in welchen Maßstäben versucht werden sollen zu beschriften. Basis bildet dabei das von Imhof beschriebene Vorgehen: „Man arbeite im Dispositionsentwurf vom Dichtezentrum aus nach außen, beginne somit im Innern dieses Zentrums und schiebe möglichst viele Namen, so gut es geht, d.h. ohne die Forderung nach eindeutiger Zugehörigkeit zu missachten, nach außen.“ [IMH62]

Wie aus der Diskussion deutlich wird existiert eine Kopplung von Schriftplatzierung und Generalisierung. Die Untersuchung wie die Schriftplatzierung den Generalisierungsvorgang beeinflusst, wie groß dieser Einfluss ist und wie eine Integration in den Generalisierungsprozess erfolgen kann, ist nicht Ziel dieser Arbeit. Diese Aspekte erfordern weitere Forschungsarbeit. Der in dieser Arbeit entwickelte reaktive Konfliktgraph kann bei der Generalisierung hilfreich sein (vgl. Abschnitt 4.5).

2.2 Begriffe und Methoden aus der Informatik

In den folgenden Abschnitten wird eine kurze Einführung in die Komplexitätstheorie und *O-Notation*, die für die Laufzeitanalyse notwendig ist, sowie in die verwendeten Datenstrukturen gegeben. Abschließend erfolgt eine kurze Darstellung der Komplexität der Schriftplatzierung.

2.2.1 Komplexität und O-Notation

Die Ausführungen dieses Abschnitts sind [OTW93], [CLS01] und [MAN89] entnommen und können dort vertiefend nachgelesen werden.

Eine wichtige Eigenschaft eines Algorithmus ist sein Verbrauch an Ressourcen. Die entscheidenden Maße hierfür sind die benötigte Rechenzeit und der zur Ausführung des Algorithmus benötigte Speicherplatz.

Laufzeit und Speicherplatzbedarf werden dabei in der Regel in Abhängigkeit von der Größe der Eingabe gemessen. Man unterscheidet zwischen dem Verhalten im besten Fall (Best-Case), im Mittel (Average-Case) und im schlechtesten Fall (Worst-Case). Ausschlaggebend für einen Algorithmus und die Vergleichbarkeit verschiedener Algorithmen, die das gleiche Problem lösen, ist die Worst-Case-Laufzeit. Die Average-Case-Laufzeit, die augenscheinlich sinnvoller erscheint, ist weniger geeignet, da es in der Regel unklar ist, worüber der Durchschnitt gebildet werden soll. Sie ist technisch schwerer durchzuführen.

Bei der Ermittlung der Laufzeit und des Speicherplatzbedarfs im Worst-Case kommt es auf das asymptotische Verhalten in Abhängigkeit von der Größe der Eingabe an, d.h. für das Verhalten bei großen Werten spielen additive oder multiplikative Konstanten keine Rolle. Um solche Größenordnungen, also Wachstumsordnungen von Funktionen, auszudrücken, hat sich die *O-Notation* zur Abschätzung des Wachstums von Funktionen nach oben etabliert.

Anstelle der formal korrekten Beschreibung: „für die Laufzeit $T(n)$ eines Algorithmus in Abhängigkeit von der Problemgröße n gilt $\forall n : T(n) \leq c_1 n + c_2$ mit den zwei Konstanten c_1 und c_2 “ sagt man: „ $T(n)$ ist von der Größenordnung n “ oder „ $T(n)$ ist $O(n)$ “ oder „ $T(n)$ ist in $O(n)$ “ und schreibt: „ $T(n) = O(n)$ “ oder „ $T(n) \in O(n)$ “.

Formal wird für eine Funktion f die Klasse der Funktionen $O(f)$ wie folgt definiert¹:

$$O(f) = \{g \mid \exists c_1 > 0 : \exists c_2 > 0 : \forall n \in \mathbb{N}^+ : g(n) \leq c_1 f(n) + c_2\}$$

Dabei werden nur Funktionen mit nicht negativen Werten betrachtet, weil negative Laufzeiten und negativer Speicherplatzbedarf keinen Sinn machen.

Im Wesentlichen werden bei der Effizienzmessung von Algorithmen in Abhängigkeit von n folgende Funktionen bzw. Ausdrücke benutzt, die disjunkte Klassen bilden (sortiert nach der Stärke des Wachstums):

$\log n$	logarithmisches Wachstum
n	lineares Wachstum
$n \log n$	n-log-n Wachstum
n^2, n^3, \dots	quadratisches, kubisches,... Wachstum
$2^n, 3^n, 4^n, \dots$	exponentielles Wachstum

In den in dieser Arbeit entwickelten Algorithmen in den Kapiteln 4 und 5 werden Sortierverfahren verwendet und mit einer Laufzeit von $O(n \log n)$ abgeschätzt. Bekannte Sortierverfahren wie Mergesort und Heapsort im Worst-Case sowie Quicksort im Average-Case garantieren diese Laufzeiten. Details über die Verfahren und Laufzeitabschätzungen können in [MAN89], [CLS01], [OTW93], [CLR90] nachgelesen werden und werden im Folgenden als bekannt vorausgesetzt.

Probleme werden in Abhängigkeit der Laufzeit ihrer zugehörigen Lösungsalgorithmen in zwei grundlegende Klassen, **P** und **NP**, eingeteilt [CLS01]. Formal wird mit **P** die Menge aller Probleme bezeichnet, die ein deterministischer Algorithmus in Polynomialzeit, d.h. in der Ordnung $O(p(m))$ (für ein Polynom p), löst. **NP** ist die Menge aller Probleme, die ein nichtdeterministischer Algorithmus in Polynomialzeit löst. Offensichtlich ist $\mathbf{P} \subseteq \mathbf{NP}$, denn jeder deterministische Polynomialzeitalgorithmus ist auch ein nichtdeterministischer

¹ Die übliche und in dieser Arbeit verwendete Schreibweise $O(n)$, $O(n^2)$, usw. ist formal nicht ganz korrekt, da die Variable n als gebundene Variable gekennzeichnet werden müsste, d.h. anstelle von $O(n^2)$ müsste geschrieben werden: $O(f)$ mit $f(n)=n^2$.

Polynomialzeitalgorithmus. Ein Problem ist NP-vollständig, wenn es einerseits selbst in NP liegt und wenn sich seine Komplexität um nicht mehr als ein Polynom von der Komplexität jedes anderen Algorithmus in NP unterscheidet. NP-vollständige Probleme sind komplexitätsmäßig betrachtet, die schwierigsten Probleme in NP.

Anschaulich können Algorithmen in der Klasse P in vertretbarer Zeit gelöst werden, wohingegen die NP-vollständigen nach heutigem Kenntnisstand exponentielle Laufzeit haben und somit nicht in vertretbarer Zeit gelöst werden können.

Um auch Lösungen für diese Probleme in akzeptabler Laufzeit zu finden, werden heuristische Algorithmen eingesetzt. Heuristische Verfahren basieren auf Strategien, die auf Hypothesen und Vermutungen aufbauen und die mit höherer Wahrscheinlichkeit, jedoch ohne Garantie, das Auffinden einer Lösung beschleunigen.

2.2.2 Datenstrukturen

2.2.2.1 Graph

Ein Graph G besteht aus einer Menge von Knoten N (nodes) und Kanten E (edges), d.h. $G=(N,E)$ mit $G\subseteq N\times E$. Eine Kante e ist ein Tupel von Knoten $(u,v)\in E$ mit $u,v\in N$; u wird auch Anfangs- und v Endknoten genannt; u und v sind *adjazent* zueinander. Ein Knoten u ist *inzident* zur Kante $e\in E$, wenn $e=(u,w)$ oder $e=(w,u)$ gilt, d.h. er ist Anfangs- oder Endknoten der Kante. Ist $u=v$, so nennt man die Kante auch Schlinge. Der Knotengrad eines Knotens n ist die Anzahl seiner inzidenten Kanten.

Mehr zum Thema Graphen kann in [OTW93], [CLS01], [WOR95] gefunden werden.

2.2.2.2 Geometrische Datenstrukturen

Für den effizienten Zugriff auf geometrische Daten werden geometrische Datenstrukturen benötigt. Bekannte Strukturen, wie der B-Baum, der AVL-Baum, das Hashing, etc., sind unzureichend, wenn typische geometrische (mehrdimensionale) Operationen unterstützt werden sollen. So eine Operation ist z.B. die Ermittlung aller geometrischen Objekte, die in einen gegebenen Bereich fallen.

Die in dieser Arbeit verwendeten geometrischen Datenstrukturen, Segment-, Intervall- und R-Baum, werden in den nächsten Abschnitten kurz vorgestellt. Details und eine Übersicht zu geometrischen Datenstrukturen können in [SAM90], [BKO97], [OTW93], [WOR95] gefunden werden.

2.2.2.2.1 Segment-Baum

Der Segment-Baum [SAM90] ist eine Datenstruktur zur Speicherung von Intervallen zur effizienten Beantwortung der Anfrage, ob ein Anfragepunkt in einem Intervall liegt und wenn ja in welchem Intervall. Er dient zur Speicherung eindimensionaler Objekte in einer eindimensionalen Datenstruktur, wie z.B. auf die x-Achse projizierte geometrische Objekte. Im Kontext der Arbeit ist sie eine Möglichkeit zur Speicherung der Konfliktmaßstäbe zwischen zu beschriftenden Objekten (Abschnitt 4.4). In [SAM90], [OTW93] und [BKO97] können Details zum Segment-Baum gefunden werden.

Im Folgenden wird kurz die Idee des Segment-Baums an einem Beispiel für Intervalle aus \square skizziert (Abbildung 6 a)). A [3, 6], B [6, 8], C [5, 9], D [10, 12], E [6, 15] sind die zu speichernden Intervalle mit Namen. Die Elemente der Menge der Intervallgrenzen, mehrfach vorkommende Intervallgrenzen werden nur einmal berücksichtigt, bilden die Blätter des

Intervall-Baums, d.h. 3, 5, 6, 8, 9, 10, 12 und 15 sind die Blätter des Beispiels. Die Blätter des Segment-Baums sind aufsteigend von links nach rechts sortiert. Er ist ein ausgeglichener binärer (Blatt-)Suchbaum ([OTW93], [CLR90], [CLS01]). Diese Eigenschaft muss beim Einfügen und Löschen sichergestellt werden. Zusätzlich umfasst eine Liste an jedem Knoten alle Intervallnamen, deren Intervall Obermenge des Intervalls ist, die dem Teilbaum unter dem Knoten entspricht. Der Teilbaum repräsentiert somit ein Teilintervall (Segment). Innerhalb dieses Teilbaums ist der Intervallname in keiner Liste eines Knotens vorhanden.

Intervalle werden so durch ein oder mehrere Segmente (Teilintervalle) repräsentiert. In dem Beispiel entspricht das Intervall B einem Segment, wohingegen Intervall A in zwei Segmente aufgeteilt ist. Eines der Segmente von A ist ein Blattknoten.

Die Abfrage des Segment-Baums wird analog zu der des binären Suchbaums bearbeitet. Zusätzlich werden auf dem Pfad von der Wurzel zu dem Blattknoten alle Intervallnamen aufgesammelt. Dies sind genau die Intervalle, in denen der angefragte Wert liegt.

Die Verallgemeinerung des im Beispiel auf Intervall aus \square beschränkten Segment-Baums auf Intervalle aus \square ist offensichtlich.

Das Einfügen eines Intervalls kann mit $O(\log n)$ abgeschätzt werden. Die Intervalle, die den Anfragepunkt p enthalten, können in $O(\log n + k)$ bestimmt werden, wobei k die Anzahl der ermittelten Intervalle ist. Der Speicherbedarf eines Segment-Baums mit n Intervallen kann mit $O(n \log n)$ abgeschätzt werden. ([SAM90], [OTW93] und [BKO97])

2.2.2.2 Intervall-Baum

Der Intervall-Baum ist eine Speicherstruktur für Intervalle, die alternativ zum Segment-Baum verwendet werden kann und seine Motivation daher analog ist. Er dient ebenfalls der Speicherung von Intervallen zur effizienten Beantwortung des Aufspießproblems, d.h. liegt ein Anfragepunkt in einem Intervall und wenn ja in welchem Intervall. In [OTW93], [BKO97] und [SAM90] können Details zum Intervall-Baum gefunden werden.

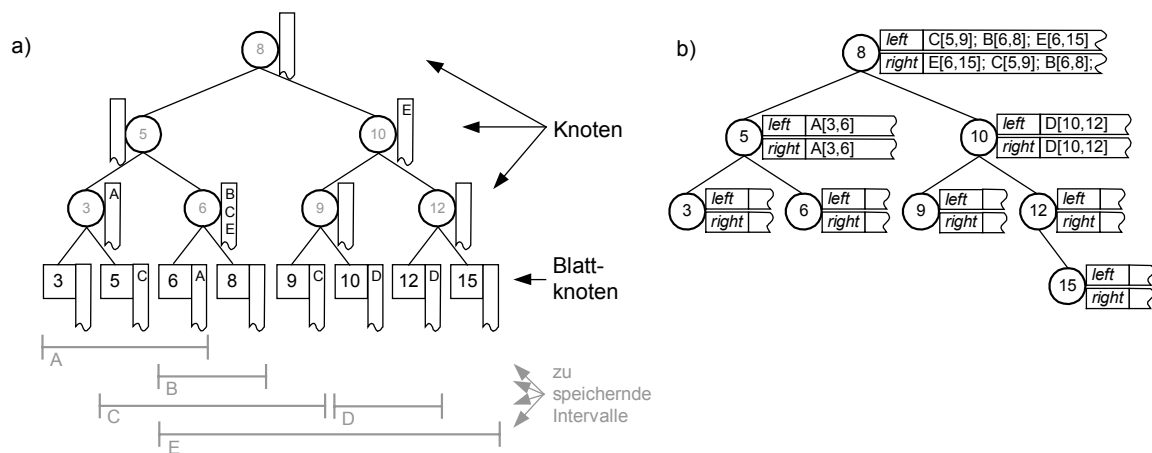


Abbildung 6 Speicherung der Intervalle A [3, 6], B [6, 8], C [5, 9], D [10, 12], E [6, 15] in einem Segment-Baum a) und in einem Intervall-Baum b).

Der Intervall-Baum wird im Folgenden anhand eines Beispiels skizziert (Abbildung 6 b)). Dabei werden, für einen anschaulichen Vergleich, die gleichen Intervalle aus \square verwendet, wie für das Segment-Baum Beispiel im letzten Abschnitt: A [3, 6], B [6, 8], C [5, 9], D [10, 12], E [6, 15]. Der Intervall-Baum ist ein ausgeglichener binärer Suchbaum ([OTW93], [CLR90],

[CLS01]) bestehend aus den Elementen der Menge der Intervallgrenzen, mehrfach vorkommende Intervallgrenzen werden nur einmal berücksichtigt, d.h. 3, 5, 6, 8, 9, 10, 12 und 15 bilden die Knoten des Baums. Jedes zu speichernde Intervall wird an dem Knoten abgelegt, der der Wurzel am nächsten ist (Pfadlänge) und dessen Schlüssel im Intervall liegt. Offensichtlich wird jedes Intervall an genau einem Knoten gespeichert. An den Blattknoten sind keine Intervalle gespeichert, wenn die untere Intervallgrenze (echt) kleiner als die obere ist.

Die Speicherung der Intervalle an den Knoten erfolgt (redundant) in zwei Listen *left* und *right*. Die *left*-Liste ist eine nach den unteren Intervallgrenzen aufsteigend sortierte Intervallliste und die *right*-Liste ist eine nach den oberen Intervallgrenzen absteigend sortierte Intervallliste.

Die Intervalle B, C und E sind an der Wurzel gespeichert, da der Knotenschlüssel „8“ innerhalb der Intervalle liegt und der Wurzel am nächsten ist, d.h. die minimalste Tiefe hat. Die zum Wurzel-Knoten gehörige *left*-Liste ist aufsteigend nach den unteren Intervallgrenzen sortiert und die *right*-Liste absteigend nach den oberen Intervallgrenzen.

Die Abfrage des Intervall-Baums wird analog zu der des binären Suchbaums bearbeitet. Dabei wird ein Pfad von der Wurzel bis zu dem Knoten durchlaufen, dessen Schlüssel gleich dem Anfragepunkt ist oder es wird an einem Blattknoten gestoppt. Von den auf dem Pfad liegenden Knoten werden entweder die *left*- oder die *right*-Liste durchlaufen. Ist der Anfragepunkt kleiner als der Schlüssel des gerade betrachteten Pfadknotens, wird die *left*-Liste durchlaufen, andernfalls die *right*-Liste. Bei dem Durchlauf der Liste werden alle Intervalle ausgegeben bis zu dem Intervall, dass vom Anfragepunkt nicht mehr aufspießt wird.

Für den Anfragepunkt 12 wird der Pfad mit den Knotenschlüsseln „8“, „10“ und „12“ durchlaufen. Bei dem Knoten mit Schlüssel „8“ wird die *right*-Liste durchlaufen, da der Anfragepunkt größer ist als der Knotenschlüssel und das Intervall E [6, 15] ausgegeben. Der Durchlauf der Liste wird abgebrochen, da der Anfragepunkt das folgende Intervall C [5, 9] nicht aufspießt. Bei dem Folgeknoten im Pfad, der den Schlüssel „10“ hat, wird ebenfalls die *right*-Liste durchlaufen, da auch hier der Anfragepunkt größer ist als der Knotenschlüssel und das Intervall D [10, 12] ausgegeben. Die Listen des letzten Knotens im Pfad mit dem Schlüssel „12“ sind leer. Die Intervalle E [6, 15] und D [10, 12] werden somit erkannt.

Für den Anfragepunkt 3 wird der Pfad mit den Schlüsseln „8“, „5“ und „3“ durchlaufen. Bei dem Knoten mit dem Schlüssel „8“ wird die *left*-Liste durchlaufen, da der Suchpunkt kleiner als der Knotenschlüssel „8“ ist. Da der Anfragepunkt schon das erste Intervall der *left*-Liste nicht aufspießt, wird der Durchlauf der Liste abgebrochen und kein Intervall ausgegeben. Bei dem Knoten mit dem Schlüssel „5“ wird die *left*-Liste durchlaufen und das Intervall A [3, 6], das vom Anfragepunkt aufspießt wird, ausgegeben.

Der Aufbau eines Intervall-Baums mit n Intervallen beträgt $O(n \log n)$. Die Intervalle, die den Anfragepunkt p enthalten, können in $O(\log n + k)$ bestimmt werden, wobei k die Anzahl der ermittelten Intervalle ist. Der Speicherbedarf bei n Intervallen beträgt $O(n)$. ([SAM90], [OTW93] und [BKO97])

Die Laufzeit für den Aufbau der Datenstruktur sowie die Extraktion der Intervalle zu einem Anfragepunkt haben bei dem Segment- und dem Intervall-Baum die gleiche Größenordnung. Der Speicherbedarf ist dagegen bei dem Segment-Baum $O(n \log n)$, während er $O(n)$ beim Intervall-Baum ist. Im Unterschied zum Segment-Baum wird beim Intervall-Baum jedes Intervall nur an einem Knoten gespeichert.

2.2.2.2.3 R-Baum

Der R-Baum [GUT84] (s. auch [SAM90]) ist eine geometrische Datenstruktur zur Speicherung und zum Zugriff auf (zunächst) zweidimensionale geometrische Objekte, die durch ein Rechteck (*minimum bounding rectangle*) approximiert werden (Abbildung 7).

Der R-Baum basiert auf dem B-Baum ([OTW93], [CLS01]), wobei im Unterschied zum B-Baum die Anzahl der Söhne jedes inneren Knotens gleich der Anzahl der Schlüsseleinträge ist. Ein R-Baum hat die Ordnung (m, M) , wenn jeder innere Knoten mit Ausnahme der Wurzel zwischen $m \leq \lceil M/2 \rceil$ und M Einträgen hat. Die Blätter enthalten die zu speichernden Objekte, deren Geometrie durch ihr kleinstes umgebende Rechteck approximiert wird. Die inneren Knoten repräsentieren die Vereinigung der approximierten geometrischen Objekte ihrer Teilbäume. Sie werden durch das kleinste umgebende Rechteck der Geometrien ihrer Teilbäume approximiert.

Die Effizienz der Datenstruktur ist dabei von der Aufteilungsstrategie überlaufender Knoten während des Einfügens abhängig. Dabei gibt es im wesentlichen zwei Strategien, die Minimierung der Gesamtfläche und die Minimierung der Schnittfläche.

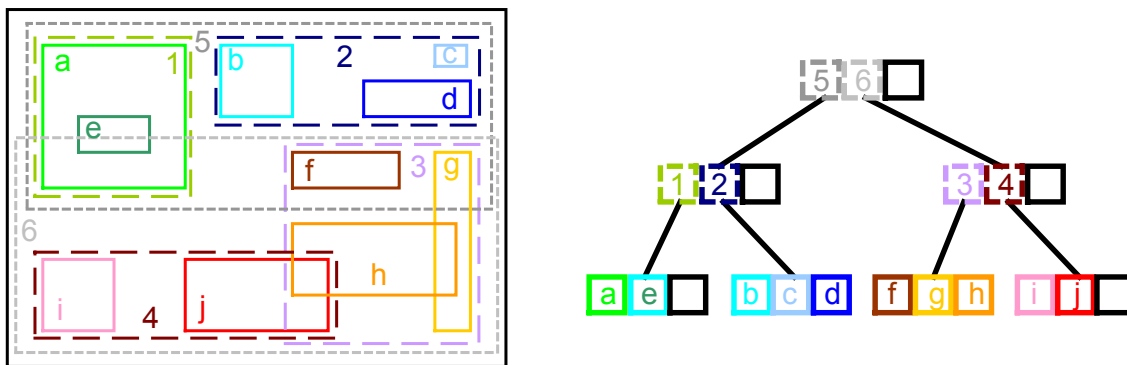


Abbildung 7 Der R-Baum für ein zweidimensionales Beispiel – links die zu speichernden Objekte a bis j und rechts die Speicherstruktur des R-Baums.

Der hier vorgestellte zweidimensionale R-Baum kann auf beliebige Dimensionen erweitert werden [SAM90]. Für den dreidimensionalen Fall werden anstelle der kleinsten umgebenden Rechtecke kleinste umfassende Quader verwendet. Analog werden die Methoden für zweidimensionale Geometrien durch die für dreidimensionale ersetzt.

Der Speicherbedarf für den R-Baum beträgt $O(n)$ [SAM90]. Der Aufbau des R-Baums mit n Objekten beträgt im Worst-Case $O(n \log n)$. Die Bestimmung der Objekte, die in einem Anfragebereich enthalten sind oder ihn schneiden, kann im Worst-Case mit $O(n)$ abgeschätzt werden. Eine genauere Abschätzung und auch die Bestimmung einer Average-Case-Laufzeit sind wegen der Abhängigkeit von der Ausdehnung der Objekte, von der Reihenfolge in der die Objekte eingefügt werden und von der Aufteilungsstrategie überlaufender Knoten beim Einfügen schwierig.

Packed R-Tree [ROL85], R^+ -Baum [SRF87], R^* -Baum [BKS90], X-Baum [BKK96] und TV-Baum [LJF95] sind Verwandte des R-Baums, die einige Optimierungen und verbesserte Strategien aufweisen.

2.2.3 Komplexität der überlappungsfreien Beschriftung

In diesem Abschnitt wird gezeigt, dass schon das Beschriftungsproblem eingeschränkt auf eine überlappungsfreie Beschriftung punkthafter Objekte eine sehr hohe, d.h. exponentielle² Laufzeit verursacht [MAS93]. Um dennoch Ergebnisse in angemessener Zeit zu erzielen, werden Heuristiken eingesetzt (vgl. Abschnitt 2.2.1).

Das Beschriftungsproblem für die Beschriftung punkthafter Objekte, das in der Literatur auch mit *simple admissible point-feature label placement* (SLP) bezeichnet wird, stellt sich wie folgt:

Gegeben ist dabei eine Menge von punkthaftern zu beschriftenden Objekten im zweidimensionalen mit je vier möglichen Beschriftungspositionen.

Gibt es eine Beschriftung punkthafter Objekte, die keine Überlappung oder Verdeckung hat?

Das beschriebene Problem der überlappungsfreien Beschriftung ist NP-vollständig (vgl. Abschnitt 2.2.1). Technisch erfolgt der Nachweis, ob ein Problem NP-vollständig ist, durch eine Transformation des Problems auf ein bekanntes und bewiesenes NP-vollständiges Problem ([CLS01], [MAN89]).

Eines der bekanntesten nachgewiesenen NP-vollständigen Probleme, ist das *Erfüllbarkeitsproblem* (*SAT – satisfiability-problem*). Das Erfüllbarkeitsproblem ist die Frage, ob es zu einem gegebenen booleschen Ausdruck eine Belegung der Variablen des Ausdrucks mit den Werten *true* oder *false* (wahr oder falsch) gibt, sodass der ganze Ausdruck den Wert *true* annimmt ([CLS01], [MAN89]).

Z.B. ist der boolesche Ausdruck $B = (x \vee y \vee z) \wedge (\bar{x} \vee z) \wedge y$, wenn y *true* ist und x und z *false* sind, wahr ($\vee \equiv$ oder; $\wedge \equiv$ und; $\bar{w} \equiv$ nicht w). Also ist er erfüllbar, obwohl er bei der Belegung x und z *true* und y *false*, falsch liefert.

Ein Algorithmus, der alle Kombinationen von Belegungen überprüft, um festzustellen, ob der Ausdruck erfüllbar ist, kann mit $O(2^n)$ abgeschätzt werden [CLS01] ($n \equiv$ Anzahl der Variablen des Ausdrucks; für den Ausdruck B im Beispiel ist $n = 3$).

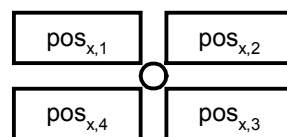


Abbildung 8 Bezeichnung der Beschriftungspositionen eines punkthaftern Objekts x bei vier zulässigen diskreten Positionen.

Anstelle einer formalen Transformation des Problems der Schriftplatzierung auf das Erfüllbarkeitsproblem zum Nachweis der NP-Vollständigkeit [MAS93], wird die Transformation an einem Beispiel (Abbildung 9) illustriert.

Ziel ist die Entwicklung eines booleschen Ausdrucks für eine Beschriftungssituation (Abbildung 9), der, wenn er ausgewertet wird, eine Aussage darüber macht, ob es eine konfliktfreie Beschriftung aller Objekte gibt oder nicht. Es ist dabei irrelevant, wie viele und welche konfliktfreien Beschriftungsmöglichkeiten es gibt.

² nach heutigem Kenntnisstand, vgl. Abschnitt 2.2.1.

Im Folgenden sei, ohne Beschränkung der Allgemeinheit, die Schriftplatzierung auf die Beschriftung punkthafter Objekte beschränkt, die jeweils vier (feste) Beschriftungspositionen mit den Bezeichnungen in Abbildung 8 haben. Für das Beispiel in Abbildung 9 wird x ersetzt durch 1 für Düsseldorf, 2 für Essen und 3 für Dortmund. Für jedes zu beschriftende Objekt wird eine Disjunktion $(pos_{x,1} \vee pos_{x,2} \vee pos_{x,3} \vee pos_{x,4})$ aufgestellt, da mindestens eine der Positionen ausgewählt sein muss, damit das Objekt beschriftet wird.

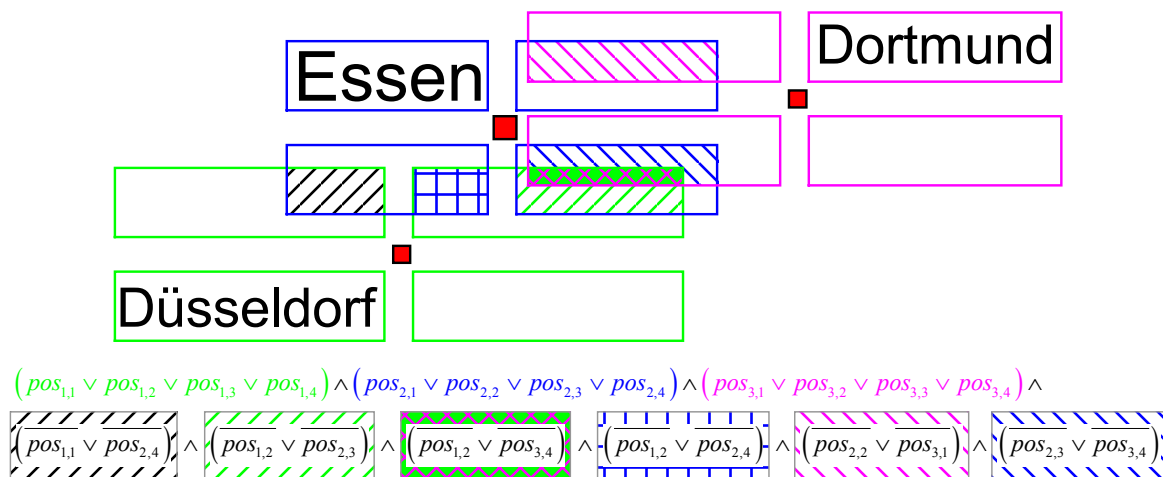


Abbildung 9 Das Beschriftungsproblem als boolescher Ausdruck am Beispiel der punkthaften Beschriftung der Städte Düsseldorf, Essen und Dortmund.

Die Disjunktionsterme werden durch Konjunktionen miteinander verknüpft; in Abbildung 9 ist dies die erste Zeile des booleschen Ausdrucks (die Farben der Disjunktionsterme entsprechen der Farbe der Beschriftungspositionen in der Grafik, die sie repräsentieren). Dadurch wird sichergestellt, dass jedes zu beschriftende Objekt eine Beschriftung erhält.

Als nächstes werden die Konflikte ermittelt und durch boolesche Terme repräsentiert. Jeder Konflikt zwischen zwei Beschriftungspositionen wird durch eine Disjunktion der negierten Positionsausdrücke repräsentiert und mit einer Konjunktion an den entwickelten Term angehängt (Abbildung 9 zweite Zeile des booleschen Ausdrucks, die unterlegte Farbe der Terme entspricht der Farbe der Konflikte in der Grafik). Wird keine oder eine der beteiligten Konfliktpositionen ausgewählt, liefert der Term *true* zurück, d.h. es entsteht kein Konflikt. Durch die Negierung der beiden Positionsausdrücke wird sichergestellt, dass, wenn beide Positionen zur Beschriftung ausgewählt werden, der Term *false* liefert und somit der gesamte Ausdruck bzw. die konfliktfreie Beschriftung für die gesamte Beschriftungssituation scheitert.

Für die Ermittlung, ob eine konfliktfreie Beschriftung existiert, müssen im Worst-Case alle Kombinationen von Belegungen getestet werden, was zu der oben beschriebenen exponentiellen Laufzeit führt. Bei vier möglichen Beschriftungspositionen pro zu beschriftenden Objekt kann die Laufzeit mit $O(4^n)$ nach oben abgeschätzt werden.

Eine mögliche konfliktfreie Beschriftungslösung für das Beispiel in Abbildung 9 ist

$$pos_{1,2} = pos_{2,2} = pos_{3,2} = true$$

$$pos_{1,1} = pos_{1,3} = pos_{1,4} = pos_{2,1} = pos_{2,3} = pos_{2,4} = pos_{3,1} = pos_{3,3} = pos_{3,4} = false$$

Der boolesche Ausdruck liefert für die Belegung

$$pos_{1,2} = pos_{2,3} = pos_{3,1} = true$$

$$pos_{1,1} = pos_{1,3} = pos_{1,4} = pos_{2,1} = pos_{2,2} = pos_{2,4} = pos_{3,2} = pos_{3,3} = pos_{3,4} = false$$

false, da es zu einem Konflikt zwischen Beschriftungspositionen von Essen und Düsseldorf kommt.

Wird die Anzahl der Beschriftungspositionen pro zu beschriftenden Objekt auf zwei beschränkt, kann gezeigt werden, dass das Problem nicht mehr in der Klasse NP liegt [FOW91]. Jedoch ist dann die Beschriftungsqualität aus kartographischer Sicht unzureichend.

Da das Schriftplatzierungsproblem NP-vollständig ist, wie skizziert und eine optimale Lösung zu einer sehr hohen Laufzeit führt (vgl. Abschnitt 2.2.1), werden zur Bestimmung von Lösungen in der Regel heuristische Verfahren eingesetzt (vgl. Abschnitt 2.2.1), wie das in dieser Arbeit entwickelte und die im nächsten Kapitel dargestellten.

Kapitel 3

Computerbasierte Verfahren der Kartenbeschriftung

Dieses Kapitel gibt einen kurzen Überblick über existierende Textplatzierungsverfahren. Einleitend erfolgt eine Darstellung der allgemeinen Vorgehensweise bei der automatischen Textplatzierung, bevor auf Auswahlalgorithmen eingegangen wird, die den Kern der Verfahren bilden. Zum Schluss erfolgt eine Vorstellung von ausgewählten Textplatzierungssystemen bzw. Ansätzen, deren Ziele nicht auf die Darstellung und Entwicklung von Auswahlalgorithmen für die Beschriftung punkthafter Objekte beschränkt sind.

3.1 Vorgehensweise bei der automatischen Textplatzierung

Im Folgenden wird ein allgemeiner Ablauf der automatischen Textplatzierung skizziert, wie er bei der Mehrzahl der (traditionellen) Verfahren ([CHR95], [ECM96], [PET96], [PEP97], [COO88]) anzufinden ist (Abbildung 10).

In der Regel werden zunächst diskrete Beschriftungspositionen in Abhängigkeit der Beschriftungsklasse, wie punkt-, linien- oder flächenhaft aber auch spezifischere Beschriftungsklassen, wie Gipfel oder Siedlungsflächen, des zu beschriftenden Objekts erzeugt. Im Anschluss oder parallel zur Erzeugung der Beschriftungspositionen werden diese lokal zum zu beschriftenden Objekt bewertet. Die Ermittlung des Informationsverlusts, der durch Verdeckung von Kartenobjekten entsteht, die keine Schriftobjekte sind, erfolgt anschließend.

Ein Auswahlalgorithmus wählt zu jedem zu beschriftenden Objekt eine Beschriftungsposition aus. Dabei handelt es sich um ein Optimierungsproblem, in das folgende Faktoren einfließen:

- Vermeidung der Überlappung von Schrift (d.h. keine Überlappung von ausgewählten Beschriftungspositionen)
- Beschriftung möglichst vieler Objekte
- Wahl einer lokal optimalen Beschriftungsposition

Abgesehen vom ersten Punkt, der kategorisch erfüllt sein muss, können die einzelnen Faktoren frei gewichtet werden, wobei indirekt die zuvor besprochenen Bewertungen einfließen. Ziel ist eine lokal und global optimale Beschriftung. Dieses Problem ist, wie in Abschnitt 2.2.3 dargestellt, sehr komplex und rechenaufwendig. Die erzielten Ergebnisse sind stark von den Verfahren zur Erzeugung der Beschriftungspositionen und der Qualität der Bewertungskriterien abhängig. Hier entstehende Unzulänglichkeiten kann der Auswahlalgorithmus nicht ausgleichen. Daher werden oft heuristische Algorithmen, die ein suboptimales Ergebnis in der

Auswahlphase erzielen aber dabei eine wesentlich kürzere Laufzeit benötigen, verwendet (vgl. nachfolgenden Abschnitt sowie Abschnitt 2.2.1 und 2.2.3).

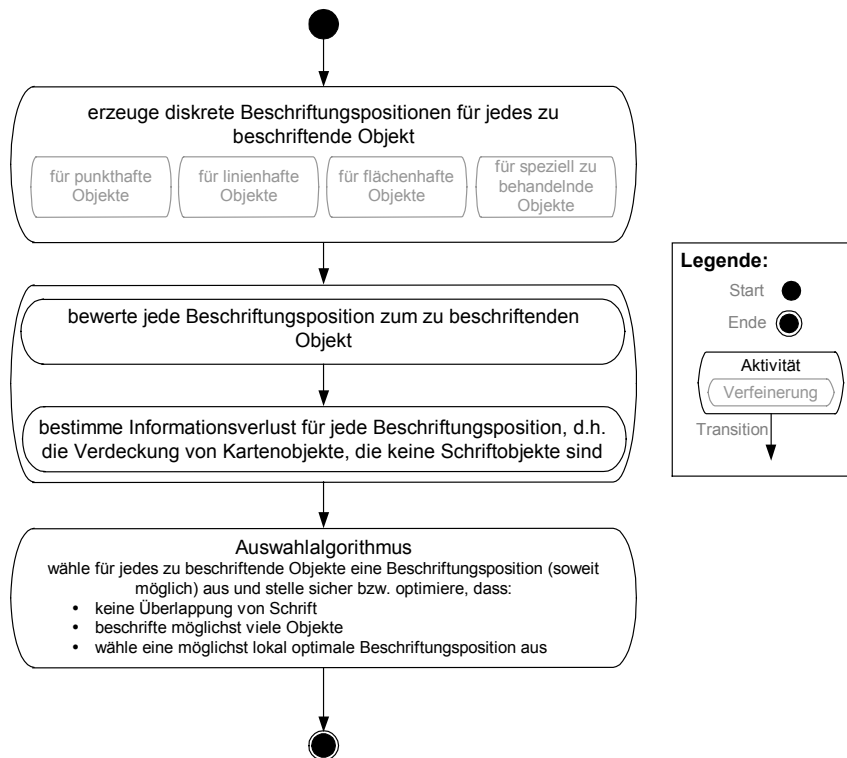


Abbildung 10 Allgemeiner Ablauf der automatischen Textplatzierung in Form eines Activity-Diagramms [RJB99].

3.2 Auswahlalgorithmen

Das Problem der automatischen Textplatzierung wurde erstmals 1972 von Yoeli [YOE72] untersucht und die Beschriftung von punkt- und flächenhaften Objekten behandelt. Erst seit Anfang der 80er Jahre, in denen unter anderem Hirsch ([HIR80], [HIR82]) einen ersten wichtigen Versuch zur automatischen Beschriftung von Punktobjekten veröffentlicht hat, wird diesem Forschungsgebiet mehr Bedeutung beigemessen.

Aus Sicht der Informatik ist der Auswahlalgorithmus von Interesse, der den Kern der automatischen Textplatzierung bildet. Für dieses kombinatorische Problem wurde eine Vielzahl an Verfahren entwickelt, deren wichtigste Vertreter in diesem Abschnitt skizziert werden.

Da die Beschriftung punkthafter Objekte einfacher als die der anderen Beschriftungsklassen, wie linien- oder flächenhafter Objekte, ist und die Komplexitätsklasse des Beschriftungsproblems unverändert bleibt, wurden viele Algorithmen zunächst für die Beschriftung punkthafter Objekte entwickelt, wie in diesem Abschnitt dargestellt. Später erfolgte eine Erweiterung zur Beschriftung linien- und flächenhafter Objekte. Dabei wurden gleichzeitig Informationsverluste, die durch Verdeckung entstehen, berücksichtigt. Dieses führt zu den stärker auf kartographische Aspekte eingehenden Textplatzierungssystemen, die im nächsten Abschnitt vorgestellt werden.

Im letzten Abschnitt wurde die Aufgabenstellung des Auswahlalgorithmus umrissen. Wie schon in den vorhergehenden Abschnitten (vgl. Abschnitt 2.2.1 und 2.2.3) diskutiert, gibt es zwei Klassen von Auswahlalgorithmen. Die erste Klasse sind die, die alle Beschriftungsmöglichkeiten testen und damit eine global optimale Beschriftung erzielen. Wegen der NP-Vollständigkeit dieses Problems ist ihre Laufzeit exponentiell. Die zweite Klasse von Algo-

rithmen treffen stets lokal optimale Entscheidungen und finden somit ein lokales Optimum, das aber nicht unbedingt mit dem globalen Optimum übereinstimmen muss. Um dennoch ein globales Optimum zu erreichen werden Heuristik verwendet. Dies wird im zweiten Teil dieses Abschnitts detailliert behandelt.

Exemplarisch werden einige Auswahlalgorithmen im Folgenden kurz dargestellt. Details zu diesen Verfahren, eine umfassende Übersicht und Vergleich von Auswahlalgorithmen können in [CAR87], [CHR95], [CMS95], [PEP97], [WOL99] gefunden werden.

Stellvertretend für die erste Klasse, die eine globale Suche durchführen, steht *Backtracking* aus dem Bereich der logischen Programmierung und KI [NIL80]. Hier kann man zwei Varianten unterscheiden: bei der erschöpfenden Suche werden alle möglichen Kombinationen von Labeln getestet, bewertet und die Kombination ausgewählt, die das beste Ergebnis liefert. Bei der zweiten Variante wird nach der ersten überlappungsfreien Platzierung gestoppt. Wenn eine überlappungsfreie Platzierung existiert, wird diese gefunden.

Die möglichen Labelpositionen werden nach einer vorher definierten Reihenfolge bzw. Priorität ausgewählt. Wenn ein Objekt nicht beschriftet werden kann oder alle Objekte beschriftet wurden und alternative Kombinationen getestet werden müssen, werden eine oder mehrere zuletzt ermittelten Beschriftungen rückgängig gemacht und es wird versucht, auf einem anderen Weg neu zu beschriften. Es wird also eine andere mögliche Labelposition ausgewählt, wodurch ein Suchbaum entsteht.

Der *Greedy Algorithmus* [CLR90] ist einer der Vertreter für heuristische Algorithmen [CHR95]. Bei diesem Verfahren werden die zu beschriftenden Objekte sequentiell (d.h. nacheinander) nach einem vorgegebenen Muster bearbeitet, z.B. in der Reihenfolge der Priorität der Objekte. Dabei wird jedes Objekt an der Position beschriftet, die im Vergleich zu den anderen möglichen Positionen das bessere Gesamtbewertungsergebnis der bisher bearbeiteten Objekte liefert. Eine Nachbehandlung, d.h. eine Repositionierung von zuvor beschrifteten Objekten, findet nicht statt.

Der (*Discrete*) *Gradient Descent Algorithmus* basiert auf dem Greedy Algorithmus, lässt aber Nachbehandlungen zu [CHR95]. Dazu werden die Beschriftungspositionen der Konfliktpartner des aktuell bearbeiteten Objekts darauf überprüft, ob deren Repositionierung zu einer besseren Gesamtbewertung führt. Wenn nötig, wird diese durchgeführt.

Methoden der Linearen Programmierung [SCH86], [CLR90] als Auswahlalgorithmus werden in [ZOR86] und [KRE94] verwendet. Dazu wird das Schriftplatzierungsproblem mit den in diesem Gebiet entwickelten Methoden, die auch Heuristiken umfassen, gelöst. Das Verhältnis zwischen Laufzeit und Gesamtbeschriftungsqualität ist, wie in [CHR95] ermittelt, ungünstiger im Vergleich zu den anderen hier beschriebenen Verfahren.

Ausgehend von der logischen Programmierung wird in [COO88], [JON89], [COJ90], [JON90] das Auswahlverfahren mit Constraints beschrieben. Ähnlich zu der Beweisführung der NP-Vollständigkeit mit Hilfe des Erfüllbarkeitsproblems und der booleschen Ausdrücken in Abschnitt 2.2.3 wird das Optimierungsproblem des Auswahlalgorithmus als ein Prolog-Ziel formuliert.

Das Verfahren von [HIR80], [HIR82] ist auf die Beschriftung von punkthaften Objekten beschränkt (Abbildung 11). Der Label kann frei um das punkthafte Objekte herumgleiten (entlang eines Abstandskreises, Abbildung 11 a) und b)). Der (kürzeste) Abstand zwischen Objekt und Label ist dabei konstant und das Schriftobjekt ist immer waagrecht ausgerichtet. Wenn es zu Überlappungen kommt, werden Überlappungs- bzw. Resultierendenvektoren gebildet (Abbildung 11 c)). Anschaulich schieben sich die Label von der Überlappungsstelle weg unter Berücksichtigung ihres konstanten Abstands zum zu beschriftenden Objekt.

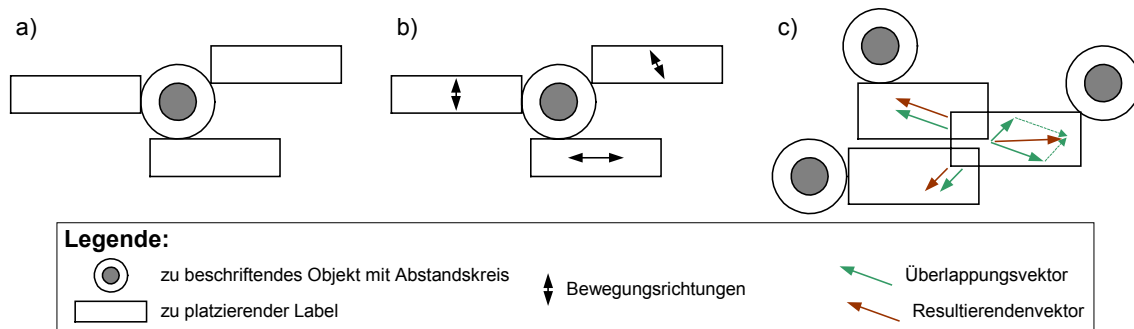


Abbildung 11 Verfahren von Hirsch [HIR80]. [HIR82]: a) und b) zeigen mögliche Beschriftungspositionen und Bewegungsrichtungen; c) die Auflösung von Überlappungen.

Heuristischen Algorithmen, wie der Gradient Descent Algorithmus oder Greedy, haben in der Regel den Nachteil, dass sie in *lokale Minima* (lokal optimale Beschriftung) verfallen. Um zu einem globalen Minimum, d.h. einer insgesamt optimalen Beschriftung, zu gelangen, müssen Verschlechterungen der Gesamtsituation in Kauf genommen werden (Abbildung 12).

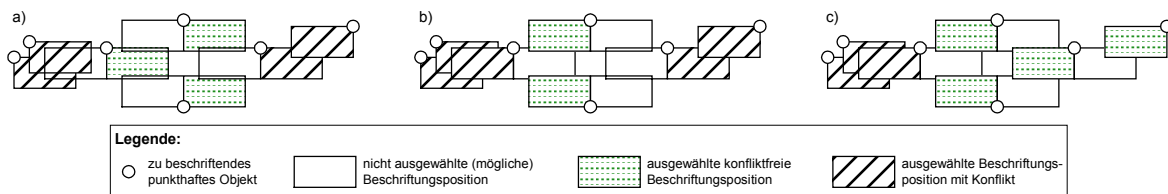


Abbildung 12 Der Weg von lokalen Minima (4 Überlappungen in a)) zu einem globalen Minimum (3 Überlappungen in c)) über Inkaufnahme von Verschlechterungen (5 Überlappungen in b)).

Simulated Annealing [KGV83] ist ein stochastischer Optimierungsalgorithmus, dessen Grundidee aus der statistischen Physik kommt. Dieser Algorithmus lässt Verschlechterungen der Gesamtbewertung bis zu einem gewissen Grad zu, um so aus lokalen Minima zu einem globalen Minimum zu gelangen. Mit zunehmender Zeit wird die Wahrscheinlichkeit eine Verschlechterung zu akzeptieren gesenkt und somit der Grad der akzeptierten Verschlechterung. [CHR95], [CMS92], [CMS95] und [PEP97] setzten dieses Verfahren um, wobei gute Ergebnisse in Bezug auf Laufzeit und Beschriftungsqualität erzielt wurden.

Einen anderen Ansatz wird in [HEB98] und [PPH99] entwickelt, auf dem auch das in dieser Arbeit präsentierte Verfahren basiert. Eine wesentliche Eigenschaft dieses Ansatzes ist die Zerteilung des Beschriftungsvorgangs in eine Präprocessing- und eine Interaktionsphase. Dabei werden soweit wie möglich die laufzeitintensiven Berechnungen in die Präprocessingphase verlagert, in welcher der so genannte Konfliktgraph als Speicher- und Zugriffsstruktur aufgebaut wird, bevor es zu einer Interaktion des Nutzers mit dem System kommt. Typische Methoden für die Interaktion mit dem Bildschirm, wie Scrollen und Zoomen, werden unterstützt. In der Interaktionsphase kann einer der zuvor beschriebenen Auswahlalgorithmen wie Greedy oder Simulated Annealing verwendet werden. Diese werden für die Nutzung des in der Präprocessingphase aufgebauten Konfliktgraphen adaptiert. Dadurch kann eine Verkürzung der Laufzeit der Interaktionsphase erzielt werden. Im folgenden Kapitel und insbesondere in Abschnitt 4.1.2 wird auf diesen Ansatz detailliert eingegangen.

Den Fokus auf technische und theoretische Aspekte aus Sicht der Informatik legen Alexander Wolff, Marc van Kreveld und ihre Gruppen ([WAG94], [WAW95], [AKS97], [WAW97], [KSW98], [KSW99], [DKS99], [STK99a], [STK99b], [WOL99], [QWX00], [TWX00],

[STR01], [STW01], [WTX02], [SCH95]). Ihre Arbeiten befassen sich mit der Reduktion der Laufzeit durch Einschränkungen, wie die Begrenzung der Anzahl der möglichen Beschriftungspositionen auf zwei oder durch die Begrenzung der Anzahl der möglichen Konflikte, um polynomiale Laufzeit zu erzielen. Ein anderer Ansatz ist, dass alle punkthaften Objekte mit der gleichen Labelgröße, die maximal sein soll, beschriftet werden sollen (*Label-Size Maximization Problem*). Dies wird durch ein stetiges und gleiches Wachstum aller Label erreicht. Die Beschriftung punkthafter Objekte mit zwei Labeln, mit Kreisen, oder mit zwei Kreisen werden ebenfalls behandelt, ebenso, wie die Erweiterung auf kontinuierliche Beschriftungspositionen.

3.3 Textplatzierungssysteme

In diesem Abschnitt werden kurz einige Textplatzierungssysteme skizziert, die neben den im letzten Abschnitt behandelten punkthaften, auch linien- und flächenhafte Objekte beschriften, sowie Informationsverluste, die durch Verdeckung entstehen, berücksichtigen. Teilweise bauen sie auf den im letzten Abschnitt beschriebenen Auswahlalgorithmen auf. Der Einblick in die interne Struktur kommerzieller Textplatzierungssysteme ist nur sehr beschränkt möglich, da sie nach außen als „Blackbox“ auftreten. Der Überblick beschränkt sich auf die wichtigsten Systeme.

Freeman und Ahn gelang mit *AUTONAP* [AHF83], [AHF84a], [AHF84b], [FRE91], einem regelbasierten Expertensystem zur automatischen Beschriftung von Kartenobjekten, das in Fortran geschrieben ist, ein Durchbruch in der Beschriftungsqualität. *AUTONAP* kann punkt-, linien- und flächenhafte Objekte beschriften. Die Beschriftung flächenhafter Objekte ist nicht auf die achsenparallele Beschriftung beschränkt. Damit die Beschriftung die Geometrie der Fläche widerspiegelt, wird mit Hilfe von Wellenfronten das Skelett der Fläche, ein Linienzug, gebildet.

Mit diesem Hintergrund gründete Freeman 1997 die Firma *MapText*³, die das Produkt *Label-EZ* zur automatischen Schriftplatzierung anbietet. Der Einsatzbereich dieses regelbasierten Systems ist insbesondere die Beschriftung von vielen ähnlichen Karten, wie sie z.B. bei dem Statistischen Bundesamt der USA anfallen.

Zoraster setzt sein im letzten Abschnitt skizziertes Verfahren zur Beschriftung von Karten für die Erdölindustrie, die aus seismischen Messpunkten mit den zugehörigen Messergebnissen bestehen, ein.

NAMEX ist ein weiteres regelbasiertes Textplatzierungssystem, das auf Cook und Jones zurückgeht [COO88], [JON89], [COJ90], [JON90]. Es ist ein hybrides System, das auf der Programmiersprache Prolog im Bereich der Beschriftungsregeln und auf Fortran im Bereich der räumlichen Daten und deren Zugriffe basiert. Ziel ist es, Karten unterschiedlichen Maßstabs, u.a. 1 : 625 000, des britischen Ordnance Survey automatisch zu beschriften.

NAMEX setzt sich aus 5 Komponenten zusammen:

- Beschriftungsregeln
- Datenbank mit den zu beschriftenden Objekten
- Prolog-Mechanismen, um die Datenbank abzufragen

³ <http://www.maptext.com/>

und um die Schriftplatzierung zu bewerkstelligen:

- Benutzerschnittstelle
- Programm zur Visualisierung

Die Anzahl der möglichen Beschriftungspositionen pro punkt- und linienhaftes zu beschriftendes Objekt ist auf maximal zwanzig beschränkt. Die Beschriftung flächenhafter Objekte ist auf den horizontalen und diagonalen Verlauf der Beschriftung beschränkt. Durch die Verwendung von Prolog ist die explizite Darstellung von Regeln möglich.

Die Entwicklung von *Maplex*, einem regelbasiertem Softwareprodukt zur automatischen Kartenbeschriftung, geht auf Jones [JON89], [JON90] zurück. *Maplex* [ESR98] ist Teil der Produktpalette von *ESRI*⁴, das jedoch nicht homogen in die Produktkette *ArcGIS* eingebunden ist, sondern als stand-alone Produkt eingesetzt wird. In [KAH01] kann eine Analyse und ein Vergleich von *Maplex* mit weiteren kommerziellen Systemen – insbesondere im Hinblick auf die Beschriftung von linienhaften Objekte – gefunden werden.

Auf die Beschriftung von punkt-, linien- und flächenhaften Objekten sowie die Berücksichtigung von Verdeckungen von Kartenobjekten, die keine Schriftobjekte sind, ist der von Christensen verwendete Auswahlalgorithmus auf Basis von Simulated Annealing in [ECM96] erweitert. Die Beschriftung flächenhafter Objekte wurde dabei auf die Beschriftung von punkthaften zurückgeführt.

⁴ Environmental Systems Research Institute, Sitz in Redlands, USA.

<http://www.esri.com> bzw. <http://www.esri.com/maplex>

Kapitel 4

Vorausberechnung von Konflikten

Wie kann eine schnelle Beschriftung von Bildschirmkarten erreicht und das Scrollen und Zoomen unterstützt werden? Das ist die zentrale Frage, die in dieser Arbeit beantwortet wird. Dazu wird ein Konzept vorgestellt, das eine Teilung des Beschriftungsprozesses in zwei Phasen vorsieht. Ziel ist dabei die Verlagerung von möglichst vielen zeitintensiven Berechnungen in eine erste Phase, die vor einer Interaktion des Nutzers erfolgt. Das führt zu einer Entlastung der zweiten Phase und somit für den Nutzer zu einer schnellen Beschriftung. Für die Kopplung der beiden Phasen und die Unterstützung der Funktionen Scrollen und Zoomen wird dazu in der ersten Phase eine Datenstruktur generiert, auf die die zweite Phase aufbaut.

Dieses Kapitel behandelt die erste Phase, die so genannte Präprocessingphase. In Abschnitt 4.1 erfolgt ein Überblick über die erste Phase und die Datenstruktur, den Konfliktgraphen, bevor am Ende des Abschnitts die Struktur des Kapitels dargestellt wird. Die Kopplung von erster und zweiter Phase wird zunächst in Abschnitt 4.1.1 skizziert. Kapitel 5 behandelt schließlich ausführlich die zweite Phase, die Interaktionsphase und die Nutzung der gemeinsamen Datenstruktur. Zentrale Begriffe können im Glossar (Anhang S. 155) nachgeschlagen werden.

4.1 Konfliktgraph – Überblick und Idee

Der Konfliktgraph ist eine Datenstruktur, die in der Präprocessingphase aufgebaut und in der zeitkritischen Interaktionsphase zur schnellen Kartenbeschriftung genutzt wird. Ziel ist die Verlagerung von zeitintensiven Berechnungen aus der Interaktionsphase, in der ein Anwender mit der Bildschirmkarte kommuniziert bzw. diese zur Navigation nutzt, in eine Präprocessingphase. Der Konfliktgraph speichert dazu alle für die Beschriftung relevanten Informationen und stellt effiziente Methoden bereit, die eine kartographisch gerechte Kartenbeschriftung in Echtzeit ermöglichen. Im Gegensatz zu den meisten bisherigen Verfahren (vgl. Kapitel 3) ist das Konzept des reaktiven Konfliktgraphen nicht auf diskrete Beschriftungspositionen beschränkt, sondern lässt ein Kontinuum zu, das nur durch den Beschriftungsraum eingegrenzt ist. Insbesondere die für die Karteninteraktion wichtigen Funktionen wie Zoomen (freie Maßstabswahl), Scrollen (Änderung des Kartenausschnittes) sowie Zusammenführen und Beschriften heterogener Daten (freie Kombination von Beschriftungslayern/-themen) werden unterstützt. Um dieses Ziel zu erreichen, werden Datenstrukturen und Verfahren aus der Graphentheorie, Computergrafik sowie kartographisches Wissen genutzt und zusammengeführt.

Zur anschaulicheren Beschreibung der Grundidee und des Konzepts des Konfliktgraphen werden zunächst einige einschränkende Annahmen getroffen, bevor stufenweise das Konzept verallgemeinert wird. Erstes Ziel ist die Darstellung der Datenstruktur des Konfliktgraphen. Wie der Name sagt, handelt es sich dabei um einen Graphen (s. 2.2.1.1) der hier geometrisch eingebettet ist und der aus einer Menge von (Konfliktgraph-)Knoten und (Konfliktgraph-)Kanten besteht⁵. Die Knoten repräsentieren die zu beschriftenden Objekte und die Kanten die potentiellen Konflikte (Abbildung 13). Das bedeutet, dass der Konfliktgraph das „Negative“ speichert. Die Kanten sind ungerichtet und, wie die Knoten, attribuiert. In diesem Kapitel wird der Konfliktgraph mit Eigenschaften detailliert und formal beschrieben.

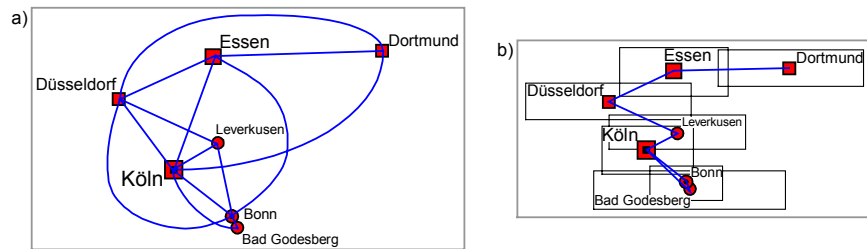


Abbildung 13 Der maßstabsunabhängige reaktive Konfliktgraph für zu beschriftende punkthafte Objekte in a) und zu einem spezifischen Maßstab mit Beschriftungsräumen in b).

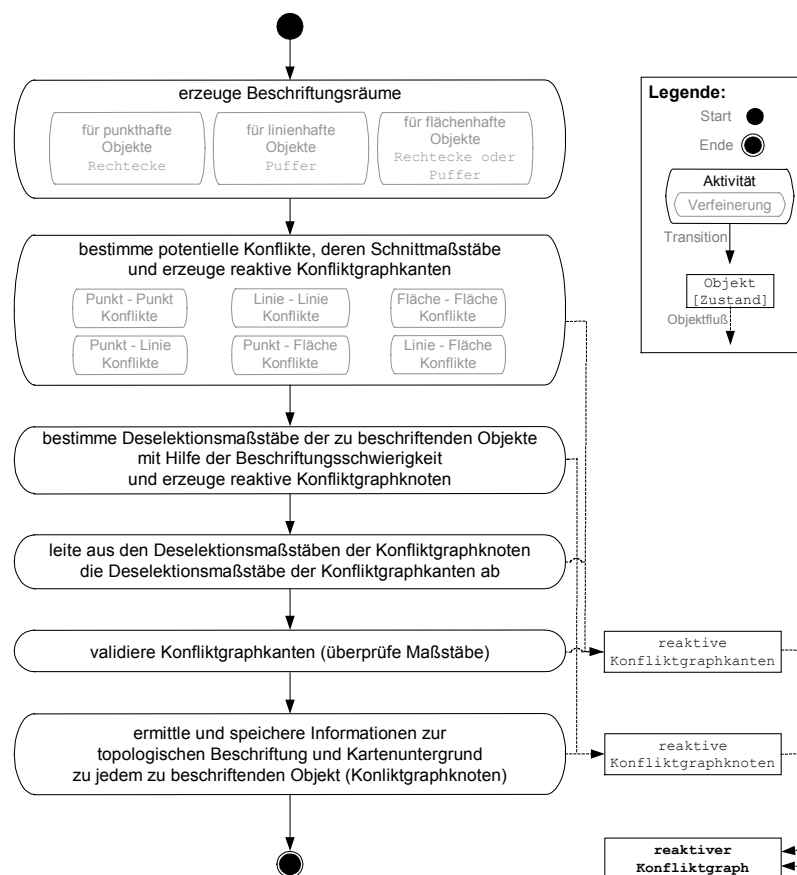


Abbildung 14 Überblick über die Präprocessingphase in Form eines Activity-Diagramms ([RJB99], [OES98]) erweitert um die Objektzustände des reaktiven Konfliktgraphen.

⁵ Im Folgenden werden zu beschriftendes Objekt, Konfliktgraphknoten, Konfliktknoten und Knoten synonym verwandt genauso wie potentieller Konflikt, mögliche (Schrift-)Überlappung, Beschriftungsraumkonflikt, Konfliktgraphkante, Konfliktkante und Kante.

Der Aufbau des Konfliktgraphen verläuft in zwei Schritten. Im ersten Schritt werden alle zu beschriftenden Objekte als Graphknoten aufgenommen und ihr zugehöriger Beschriftungsraum bestimmt (Abbildung 14). Er ist die Fläche in der alle möglichen Beschriftungspositionen für ein Objekt liegen. Im zweiten Schritt werden die Konfliktgraphkanten, d.h. die potentiellen Konflikte, ermittelt. Dazu werden die Beschriftungsräume der Konfliktgraphknoten gegenseitig auf Überlappung getestet. Die Beschriftungsräume sind parametrisiert mit dem Maßstab, da sie abhängig von der Schriftgröße und diese wiederum vom Maßstab ist. Ein Konfliktgraph zu einem spezifischen Maßstab wird mit *statischer Konfliktgraph* bezeichnet im Unterschied zum *reaktiven Konfliktgraph* der maßstabsunabhängig ist. Die Definition des Konfliktgraphen erfolgt in Abschnitt 4.2 bevor seine Generierung in 4.3 vorgestellt wird. Der Begriff *Maßstabsunabhängigkeit* im Kontext des reaktiven Konfliktgraphen und seine Grenzen aus kartographischer Sicht wird in Abschnitt 4.5 ausführlich diskutiert.

Die Betrachtung des Konfliktgraphen beschränkt sich zunächst auf punkthafte Objekte (Abschnitt 4.3.1) und deren potentielle Konflikte (4.3.2) bevor sie auf linien- und flächenhafte Objekte (4.3.4 bzw. 4.3.9) sowie auf Konflikte zwischen diesen Geometrietypen ausgeweitet wird. Die Generierung des gemeinsamen reaktiven Konfliktgraphen wird in Abschnitt 4.3.11 vorgestellt, bevor eine detaillierte Laufzeitanalyse der zuvor entwickelten Algorithmen erfolgt (4.3.11.1). Die Speicherung des Konfliktgraphen wird in Abschnitt 4.4 behandelt. Die Integration kartographischer Beschriftungsaspekte, wie die Berücksichtigung von Verdeckungen, die Informationsverluste zur Folge haben, und der topologischen Lage der Schrift in Bezug auf Trennlinien usw., in den reaktiven Konfliktgraphen erfolgt etappenweise in den Abschnitten 4.6 und 4.7.

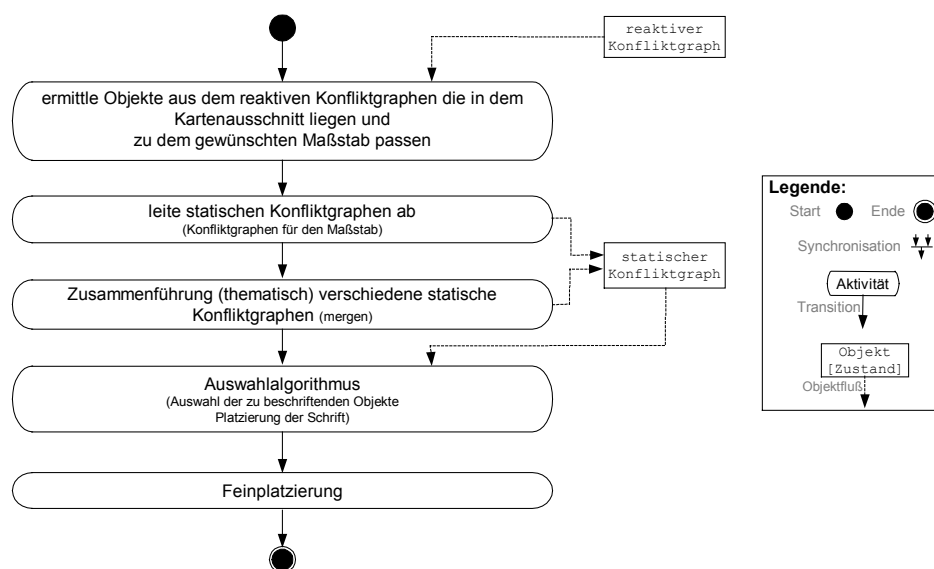


Abbildung 15 Der Ablauf der Interaktionsphase in Form eines Activity-Diagramms mit den Objektzuständen des Konfliktgraphen.

4.1.1 Verwendung des Konfliktgraphen

Die Ableitung einer Beschriftung aus dem Konfliktgraphen während der Interaktionszeit wird in Kapitel 5 diskutiert und beschrieben (Abbildung 15). Der Auswahlalgorithmus, der für die eigentliche Auswahl der Schriftposition verantwortlich ist (Abschnitt 5.2), nutzt die Datenstruktur und die zugehörigen Methoden des Konfliktgraphen. Für jedes ausgewählte Objekt, das erstmalig oder neu beschriftet werden soll, werden aus der Datenstruktur alle potentiellen Konfliktpartner entnommen. Diese sind die adjazenten Knoten. Der Konflikt-

graph erspart dabei die rechenaufwendigen Schnittoperationen. Während der Interaktionsphase muss so nur noch eine optimale, möglichst konfliktfreie Beschriftungsposition für das zu beschriftende Objekt hergeleitet werden (Abschnitt 5.2). Im Gegensatz zum globalen Problem der Bestimmung potentieller Konflikte ist dieses ein lokales Optimierungsproblem, das effizient gelöst wird.

Da der Auswahlalgorithmus auf Basis der Beschriftungsräume und der potentiellen Konflikte arbeitet und die Beschriftungsräume ein Vielfaches der Fläche ausmachen, die durch die Schrift letztendlich belegt werden, stellt sich die Frage, ob das vorgeschlagene Verfahren eine gute Beschriftungsqualität und eine angemessene Informationsdichte erreicht. Abbildung 16 zeigt das Ergebnis zweier Beschriftungsverfahren – b) und d) bzw. mit den Konfliktträumen a) und c). Das obere Ergebnis wird mit dem hier vorgestellten Verfahren erzielt und führt zu einer Beschriftung, deren Zuordnung eindeutig ist, im Gegensatz zu Standardverfahren, die lokale Optimierungen betreiben – c) und d). Dabei ist in Abbildung 16 a) und b) zu erkennen, dass die Schrift in dem hier vorgestellten Verfahren von den Konfliktzentren nach außen weg „geschoben“ wird, wie Imhof es fordert: „Die Karte soll nicht gleichmäßig mit Namen überdeckt werden. Andererseits aber sind dichte Namensknäule zu vermeiden. Dies ist sowohl bei der Namensauswahl, wie auch bei der Namensordnung zu beachten.“ [IMH62]

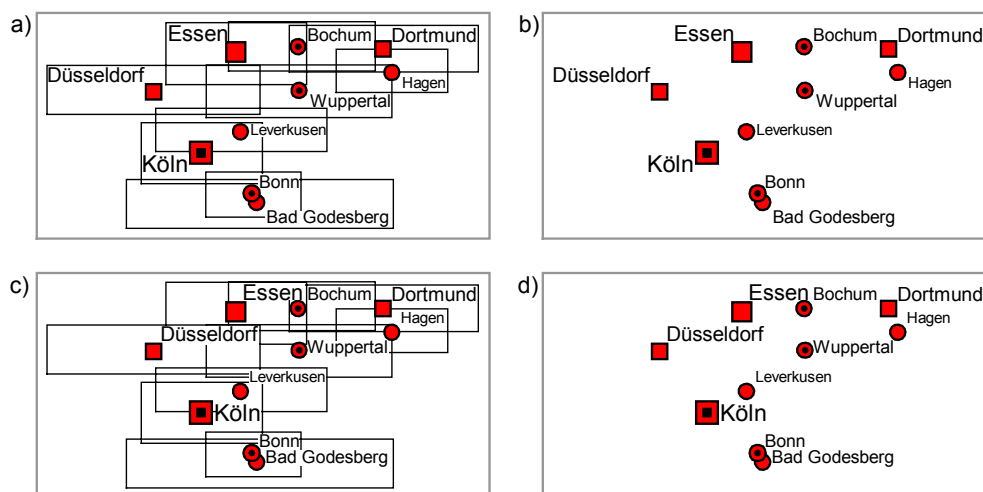


Abbildung 16 Die Beschriftung des gleichen Szenarios führt bei Verwendung unterschiedlicher Auswahl- und Platzierungsalgorithmen zu unterschiedlichen Beschriftungsergebnissen – a) und b) vs. c) und d).

4.1.2 Historie

Ein erster Ansatz der Zweiteilung des Beschriftungsvorgangs wurde erstmals in [PET96] entwickelt und vorgestellt. Ziel war die Trennung von rechenintensiven geometrischen Schnitt- bzw. Überlappungsoperationen und dem nachgeschalteten Auswahlalgorithmus, der dadurch auf das rein kombinatorische Problem beschränkt ist. Das führt zu einer Laufzeitreduktion im Auswahlalgorithmus, insbesondere bei dem Einsatz heuristischer Verfahren, wie Simulated Annealing. Diese Algorithmen wählen nach einer Heuristik zu beschriftende Objekte und deren Beschriftungspositionen aus. Eine wiederholte Neubeschriftung eines zu beschriftenden Objektes und Auswahl der gleichen Beschriftungsposition kann dabei eintreten. Durch das Preprocessing, dem Speichern der möglichen Überlappungen und einem effizienten Zugriff auf diese Informationen kann so eine Laufzeitreduktion erzielt werden.

Im Gegensatz zu dem hier vorgestellten Verfahren ist das in [PET96] auf einen festen Maßstab beschränkt und geht von diskreten Beschriftungspositionen (Labelboxen), nicht von Beschriftungsräumen, aus. Das Verfahren berücksichtigt die Beschriftung von punkt-, linien-

und flächenhaften Objekten. In Kapitel 3 und Abbildung 10 wird u.a. dieses Verfahren skizziert, welches stellvertretend für die traditionellen automatischen Beschriftungssysteme genommen werden kann.

Die Datenstruktur des statischen/reaktiven Konfliktgraphen wurde erstmals in der Diplomarbeit von Markus Heber [HEB98] verwendet und auf der ICC99 vorgestellt [PPH99]. Der Konfliktgraph wird in [HEB98] aus Sicht der Informatik dargestellt und ist auf die Beschriftung punkthafter Objekte an diskreten Beschriftungspositionen beschränkt. Abgesehen von der Vermeidung von Schriftüberlappungen und der Priorisierung von Beschriftungspositionen werden keine kartographischen Beschriftungsregeln berücksichtigt.

4.1.3 Bezugssystem

Spätestens bei der Berechnung der Schnittmaßstäbe in den Abschnitten 4.3.2.1, 4.3.5.1 und 4.3.7.1 stellt sich die Frage nach den Bezugssystemen, die den Geodaten und deren Visualisierung zu Grunde liegen. Realweltobjekte werden als Geo-Objekte gespeichert. Dabei bietet sich ihre Speicherung in *Realweltkoordinaten* an. Bei der Wahl eines Kartennetzentwurfs und Koordinatensystems muss das Problem der Abbildungsverzerrung, das bei Bildschirmkarten durch die Funktion Zoomen, d.h. die freie Maßstabswahl, entsteht, berücksichtigt werden. Dieses ist ein Problem aus dem allgemeineren Bereich der (Bildschirm-)Kartographie und nicht spezifisch für die Schriftplatzierung. Es liegt deshalb außerhalb des Fokus dieser Arbeit und wird in [HGM02] ausführlich diskutiert. Im Folgenden wird davon ausgegangen, dass Geo-Objekte in einem einheitlichen System in euklidischen Realweltkoordinaten, d.h. einem Maßstab von 1:1, vorliegen. Abstände können in diesem metrischen Raum durch die bekannten Standardverfahren berechnet werden.

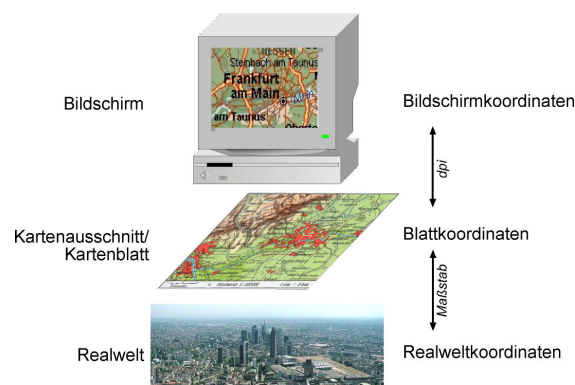


Abbildung 17 Realweltkoordinaten – Blattkoordinaten – Bildschirmkoordinaten. ([MIC01], [DIE76])

Für die Darstellung einer Karte zu einem spezifischen Maßstab werden die entsprechenden Objekte ausgewählt, und ihre Realweltkoordinaten in die zur Karte passenden *Blattkoordinaten* transformiert – durch Multiplikation mit dem spezifischen Maßstab (Abbildung 17). Bei der Berechnung der Schnittmaßstäbe wird mit Realwelt- und Blattkoordinaten gearbeitet.

Die Blattkoordinaten können jedoch nicht 1:1 auf einem Medium, wie dem Bildschirm oder dem Drucker, dargestellt werden, da diese bauartbedingte Auflösungen besitzen, die berücksichtigt werden müssen. Die Angabe der Auflösung erfolgt in *dpi* (Dots per Inch; Punkte pro 2,54cm). Das führt zu den *Bildschirmkoordinaten* (Abbildung 17). Betriebs- bzw. Grafik-Systeme stellen standardmäßig Funktionen zur Konvertierung zur Verfügung. Realweltkoordinaten in Meterauflösung können mit

$Koordinate_{Bildschirm} = Koordinate_{Realwelt} * Maßstab * \frac{Mediamauflösung}{0,0254}$ transformiert werden.

Die Auflösung eines Monitors ist abhängig von Bildschirmgröße, Abblendmaske und gewählter Auflösung und liegt deutlich unter der von Druckern (Anhaltswert für Röhrenmonitor sind 82dpi; Flachbildschirme haben einen etwas höheren dpi-Wert; Drucker haben eine Auflösung von 300dpi, 600dpi oder 1200dpi). Im folgenden wird auf die Betrachtung der Bildschirmkoordinaten verzichtet und ihre Konvertierung als integrierter Bestandteil der Plotfunktion des Grafiksystems angesehen.

Im Unterschied zu den Objekten, die in Realweltkoordinaten, d.h. im Maßstab 1:1 gespeichert sind, werden Signaturen und Beschriftungen zu einem Referenzmaßstab gespeichert, da sich ihre Größe nicht proportional zum Maßstab ändert. Das ist der Grund dafür, dass bei der Schnittmaßstabsberechnung Realwelt- und Blattschnittkoordinaten mit einfließen. Der Referenzmaßstab sollte in dem Maßstabsbereich liegen, in dem der reaktive Konfliktgraph verwendet wird. Die Signatur- und Beschriftungsgröße kann so für den Referenzmaßstab aus bestehenden Kartenwerken oder einem Musterblatt entnommen und als Referenzbeschriftung und -signatur mit dem Referenzmaßstab am Objekt abgelegt werden. Mit der in Abschnitt 2.1.2 vorgestellten Funktion kann die Referenzsignatur und -beschriftung mit dem Referenzmaßstab in den gewünschten Kartenmaßstab transformiert werden. Neben diesen drei Größen wird noch ein Referenzabstand zwischen Schrift und Signatur festgelegt.

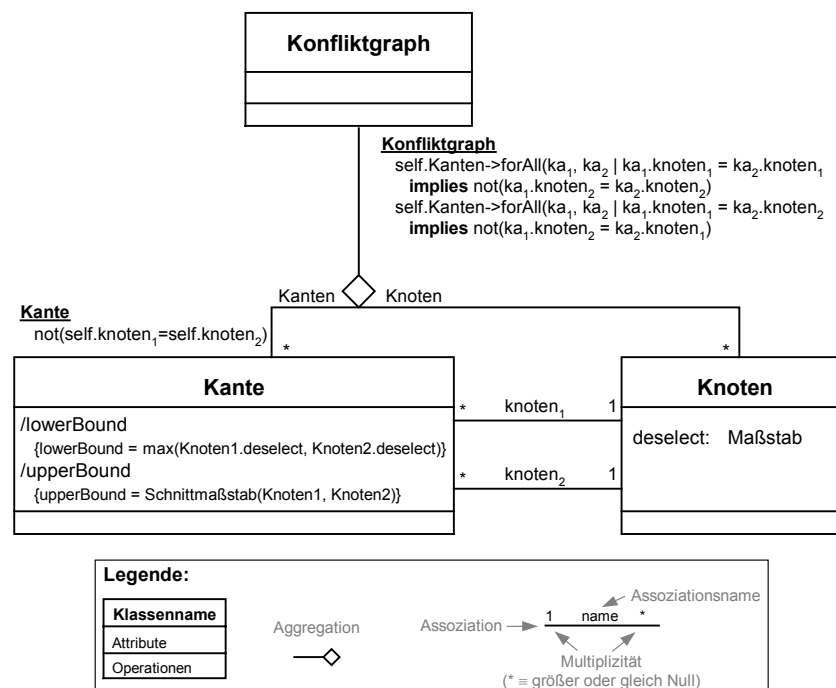


Abbildung 18 Der reaktive Konfliktgraph als Klassendiagramm in UML-Notation mit OCL-Ausdrücken (Object Constraint Language), um die Graph Eigenschaften sicherzustellen [OES98].

4.2 Definition des reaktiven Konfliktgraphen

Der *Konfliktgraph* ist ein Graph ($G \subseteq N \times E$), der aus Knoten und ungerichteten und schlingenfreien Kanten besteht (Abschnitt 2.2.2.1). Die (*Konfliktgraph*-)Knoten N repräsentieren die zu beschriftenden Objekte und die (*Konfliktgraph*-)Kanten $E \subseteq N \times N$ und

$(u, v) \in E$ mit $u, v \in N$ wobei: $u \neq v$ und $(u, v) = (v, u)$ die potentiellen Konflikte. Die Knoten sind mit dem *Deselektionsmaßstab*⁶ attribuiert und die Kanten mit den Intervallgrenzen *lowerBound* und *upperBound* (wird in Abschnitt 4.3.2.2 eingeführt). Die *lowerBound* ergibt sich aus dem Maximum der Deselektionsmaßstäbe der inzidenten Knoten und die *upperBound* aus dem Schnittmaßstab⁷ der Knoten (Abbildung 18).

Dieser maßstabsunabhängige Konfliktgraph wird *reaktiver Konfliktgraph* genannt im Unterschied zu dem *statischen Konfliktgraphen*, der zu einem spezifischen Maßstab und Kartenausschnitt gilt.

Aus Sicht der Datenstruktur und der Algorithmen ist der reaktive Konfliktgraph maßstabsunabhängig, d.h. der statische Konfliktgraph zu einem beliebigen Maßstab kann aus ihm durch einfach berechenbare Funktionen abgeleitet werden. Der reaktive Konfliktgraph gilt somit für jeden beliebigen Maßstab. Aus kartographischen Gesichtspunkten ist er jedoch auf Intervalle begrenzt, da es zu Geometriotypwechsel und damit zum Wechsel von Beschriftungsklassen sowie Generalisierungsprozessen kommt. Detailliert wird dieses Thema in Abschnitt 4.5 diskutiert.

4.3 Generierung des reaktiven Konfliktgraphen

Ausgehend von den drei Beschriftungsklassen, punkt-, linien- und flächenhaft, werden mögliche Beschriftungskonflikte bzw. die Schriftkonfliktklasse systematisch eingeführt. Die Beschriftungsräume und deren Maßstabsabhängigkeit sind dabei zentrale Punkte. Daraus entwickelt sich der Konfliktgraph, dessen Knoten die zu beschriftenden Objekte und dessen Kanten die Beschriftungskonflikte repräsentieren. Zur Ermittlung der potentiellen Beschriftungskonflikte wird für jede Kombination von Beschriftungsklassen ein Verfahren zur Bestimmung des Schnittmaßstabs entwickelt. Ein Deselektionskriterium, das auf der Bewertung der Beschriftungsschwierigkeit basiert, stellt sicher, dass in Abhängigkeit vom Maßstab und damit des zur Verfügung stehenden Platzes Objekte beschriftet werden oder nicht.

Für das Verständnis der Konfliktgraphgenerierung sind die Abschnitte punkthafte Objekte (4.3.1) und Punkt-Punkt-Konflikte (4.3.2) ausreichend.

4.3.1 Punkthafte Objekte

In Abschnitt 4.3.1.1 wird die Erzeugung des Beschriftungsraums für zu beschriftende punkthafte Objekte vorgestellt. Das punkthafte Objekt mit dem Beschriftungsraum wird durch den Konfliktgraphknoten repräsentiert. Die Ermittlung der potentiellen Schriftkonflikte zwischen Beschriftungen punkthafter Objekte wird in 4.3.2 beschrieben. Das führt zu einem ersten reaktiven, d.h. reaktiven Konfliktgraphen für die Beschriftung punkthafter Objekte.

4.3.1.1 Beschriftungsraum

In Kapitel 2.1.1.1 und 4.1.3 wurden Repräsentationen für Schriftobjekte vorgestellt. Um bei der automatischen Beschriftung eine effiziente Laufzeit zu erreichen, müssen weitere geometrische Abstraktionen getroffen werden, die die Datenstruktur des Konfliktgraphen unterstützen. Auf den ersten Blick erscheinen diese verlustbehaftet, ermöglichen jedoch die Beschriftung in Echtzeit. Durch die zum Schluss erfolgende Feinplatzierung (Abschnitt 5.3)

⁶ wird in den Abschnitten 4.3.3, 4.3.6 und 4.3.8 eingeführt

⁷ Im Folgenden wird Schnittmaßstab und Konfliktmaßstab synonym verwendet.

werden die in Kauf genommen geometrischen Vereinfachungen wieder revidiert. In der Einleitung zu diesem Kapitel sind die Vorteile dieser augenscheinlichen Vereinfachung, die zu einer eindeutigen Zuordbarkeit zwischen Schrift und zugehörigem Objekt führen, diskutiert. Insbesondere bei der Beschriftung linienhafter Objekte, die später behandelt werden, kommen die Stärken dieser Modellierung zum Vorschein.

Definition (Beschriftungsraum):

In Abhängigkeit der Beschriftungsklasse wird der Beschriftungsraum definiert. Dieser hat die folgenden Eigenschaften:

- er ist die konvexe Hülle aller möglichen Beschriftungspositionen, und
- er ist ein einfach berechenbares geometrisches Objekt



Abbildung 19 Beschriftungsraum für ein zu beschriftendes punkthafes Objekt (links). In der rechten Grafik sind in den Beschriftungsraum exemplarisch drei Labelboxen eingezeichnet.

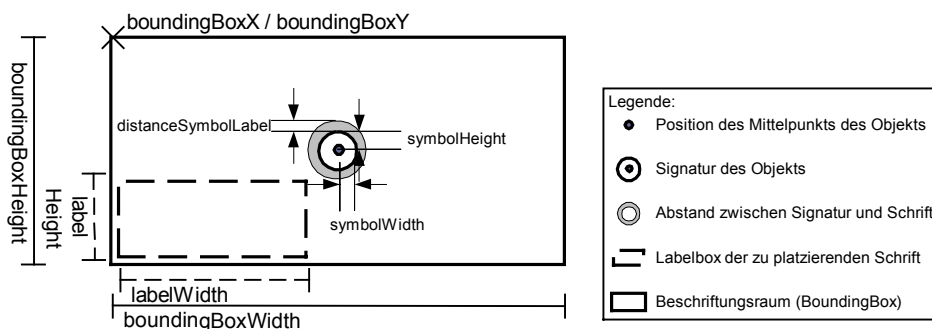


Abbildung 20 Beschriftungsraum (BoundingBox) für ein punkthafes Objekt, der sich aus den Größen der Labelbox und der Signatur ergibt.

Für punkthafte Objekte lässt sich der Beschriftungsraum (*BoundingBox*) als Rechteck beschreiben, wie in Abbildung 19 und Abbildung 20 dargestellt (vgl. Abschnitte 2.1.1.2 und 4.3.1.1.1). Wenn alle zu platzierenden Schriftobjekte bzw. die zugehörigen Labelboxen und Boundingboxen für den gleichen Maßstab definiert sind, kann so ein Konfliktgraph zu diesem festen Maßstab, sprich ein statischer Konfliktgraph, erzeugt werden. Durch Verwendung der in 2.1.2 entwickelte Skalierungsfunktion kann der Beschriftungsraum mit dem Maßstab parametrisiert werden. Im Folgenden muss zwischen der Transformation von Koordinaten eines Objekts von einem Maßstab in einen anderen und der Transformation der Größe, d.h. der Ausdehnung des Beschriftungsraums bzw. seiner Bestandteile unterschieden werden.

Ermittlung des Beschriftungsraums für punkthafte Objekte (parametrisiert mit dem Maßstab):

○ Gegeben (in Realwelt-Koordinaten):

- Position des Mittelpunkts des Objekts: $\text{objectX}, \text{objectY}$
- Referenzmaßstab für Labelbox, Signatur und Abstand: fixScale
- Größe der Referenz-SignaturBox (BoundingBox der Signatur): $\text{fixSymbolWidth}, \text{fixSymbolHeight}$
- Referenzabstand zwischen Signatur und Schrift: $\text{fixDistanceSymbolLabel}$
- Größe der Referenzlabelbox passend zum Referenzmaßstab der zu platzierenden Schrift: $\text{fixWidth}, \text{fixHeight}$

○ Skalierungsfunktionen (vgl. Abschnitt 2.1.2):

Der Skalierungsfaktor scaling überführt die Koordinaten eines Objektes aus dem Referenzmaßstab fixScale in den Zielmaßstab scale :

$$\text{scaling}(\text{scale}, \text{fixScale}) = \frac{\text{scale}}{\text{fixScale}} \quad (4.1)$$

Wert für die Änderung der Beschriftungsgröße (Abschnitt 2.1.2):

$$\text{value} \in [0..0,15] \quad (4.2)$$

Der Skalierungsfaktor referenceScaling überführt die Größe des Beschriftungsraums (BoundingBox) – Breite und Höhe nicht seine Position! – bzw. die der Bestandteile (Schrift (Labelbox), Signatur und des Abstandes) in Abhängigkeit des Referenzmaßstabs fixScale in den Zielmaßstab scale . Er unterscheidet sich von $\text{scaling}()$ in (4.1) um den Exponenten value , der die Schriftgröße (Beschriftungsraum) dem Zielmaßstab anpasst (Abschnitt 2.1.2). Er trägt dem Phänomen Rechnung, dass sich die Schriftgröße nicht proportional zum Maßstab ändert.

$$\text{referenceScaling}(\text{scale}, \text{fixScale}) = \text{scaling}(\text{scale}, \text{fixScale})^{\text{value}} \quad (4.3)$$

Die Breite der Schrift (Labelbox) im Zielmaßstab (scale) wird abgeleitet aus Referenzbreite (fixWidth) und Referenzmaßstab (fixScale):

$$\text{labelWidth}(\text{scale}, \text{fixScale}, \text{fixWidth}) = \text{fixWidth} * \text{referenceScaling}(\text{scale}, \text{fixScale}) \quad (4.4)$$

Die Höhe der Schrift (Labelbox) im Zielmaßstab wird abgeleitet aus Referenzhöhe (fixHeight) und Referenzmaßstab (fixScale):

$$\text{labelHeight}(\text{scale}, \text{fixScale}, \text{fixHeight}) = \text{fixHeight} * \text{referenceScaling}(\text{scale}, \text{fixScale}) \quad (4.5)$$

Analog zur Schrifthöhe und -breite lässt sich für die Breite der Signatur folgendes herleiten:

$$\begin{aligned} symbolWidth(scale, fixScale, fixSymbolWidth) = \\ fixSymbolWidth * referenceScaling(scale, fixScale) \end{aligned} \quad (4.6)$$

für die Höhe der Signatur im Zielmaßstab:

$$\begin{aligned} symbolHeight(scale, fixScale, fixSymbolHeight) = \\ fixSymbolHeight * referenceScaling(scale, fixScale) \end{aligned} \quad (4.7)$$

und für den Abstand zwischen Signatur und Schrift im Zielmaßstab:

$$\begin{aligned} distanceSymbolLabel(scale, fixScale, fixDistanceSymbolLabel) = \\ fixDistanceSymbolLabel * referenceScaling(scale, fixScale) \end{aligned} \quad (4.8)$$

○ Größe des Beschriftungsraums (Boundingbox):

- Breite:

- Die Breite der Boundingbox im Zielmaßstab ($scale$) in Abhängigkeit der Referenzlabelbox, -signatur und -abstand sowie dem zugehörigen Referenzmaßstab ergibt sich aus der doppelten Schriftlänge plus der doppelten Distanz zwischen Signatur und Schriftobjekt plus der Signaturbreite, jeweils abhängig vom Referenzwert und Referenzmaßstab.

- Formal ergibt sich:

$$\begin{aligned} boundingBoxWidth(scale, fixScale, fixWidth, fixDistanceSymbolLabel, fixSymbolWidth) = \\ 2 * labelWidth(scale, fixScale, fixWidth) + \\ 2 * distanceSymbolLabel(scale, fixScale, fixDistanceSymbolLabel) + \\ symbolWidth(scale, fixScale, fixSymbolWidth) \end{aligned} \quad (4.9)$$

durch einsetzen der Funktionen (4.4), (4.8), (4.6), (4.3) und (4.1):

$$\begin{aligned} boundingBoxWidth(scale, fixScale, fixWidth, fixDistanceSymbolLabel, fixSymbolWidth) = \\ (2 * (fixWidth + fixDistanceSymbolLabel) + fixSymbolWidth) * \left(\frac{scale}{fixScale} \right)^{value} \end{aligned} \quad (4.10)$$

- Höhe:

- Die Höhe der Boundingbox im Zielmaßstab in Abhängigkeit der Referenzlabelbox, -signatur und -abstand sowie dem zugehörigen Referenzmaßstab ergibt sich aus der doppelten Schrifthöhe plus der doppelten Distanz zwischen Signatur und Schriftobjekt plus der Signaturbreite, jeweils abhängig vom Referenzwert und Referenzmaßstab.

- Formal ergibt sich:

$$\begin{aligned} boundingBoxHeight(scale, fixScale, fixHeight, fixDistanceSymbolLabel, fixSymbolHeight) = \\ 2 * labelHeight(scale, fixScale, fixHeight) + \\ 2 * distanceSymbolLabel(scale, fixScale, fixDistanceSymbolLabel) + \\ symbolHeight(scale, fixScale, fixSymbolHeight) \end{aligned} \quad (4.11)$$

durch einsetzen der Funktionen (4.5), (4.6), (4.3) und (4.1):

$$\text{boundingBoxHeight}(\text{scale}, \text{fixScale}, \text{fixHeight}, \text{fixDistanceSymbolLabel}, \text{fixSymbolHeight}) = (2 * (\text{fixHeight} + \text{fixDistanceSymbolLabel}) + \text{fixSymbolHeight}) * \left(\frac{\text{scale}}{\text{fixScale}} \right)^{\text{value}} \quad (4.12)$$

4.3.1.1.1 Verbesserung der Lesbarkeit von Beschriftungen – Unterscheidbarkeit

Der angemessene Abstand zwischen Signatur und Beschriftung wird durch einen eigenen Parameter explizit sichergestellt, wie in Abbildung 20 dargestellt. Bei Beschriftungen mehrerer Objekte kann es jedoch vorkommen, dass die Beschriftungen sehr dicht beieinander liegen. Die fehlende grafische Trennbarkeit zwischen den Beschriftungen führt zu einer schlechten Lesbarkeit (Abbildung 21 b)). Um diesen Effekt zu vermeiden bzw. zu verringern kann eine „etwas zu große“ Labelbox, z.B. eine halbe Schrifthöhe breiter und höher, gewählt werden, wie in Abbildung 21 c) dargestellt. Innerhalb dieser kann die Schrift frei platziert werden, d.h. möglichst weit von anderen Beschriftungen entfernt. Dadurch kann eine bessere Lesbarkeit erzielt werden (Abbildung 21 e)). Die „Übergröße“ der Labelbox wird aus den kartographischen Beschriftungsregeln geschlossen, die in [IMH62] und [ELL01] dargestellt sind.

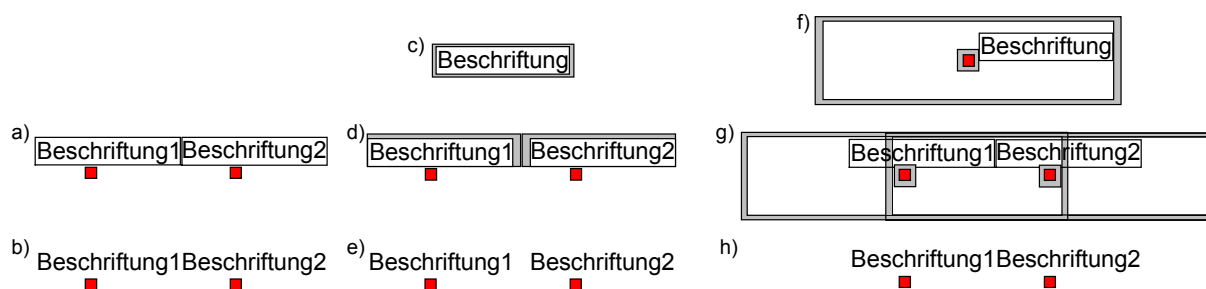


Abbildung 21 Verbesserung der Lesbarkeit von Beschriftungen, d.h. eine Verhinderung der Verschmelzung zweier Beschriftungen zu augenscheinlich einer.

Die offensichtliche Alternative, anstelle der Vergrößerung der Labelbox den Beschriftungsraum zu vergrößern, wie in Abbildung 21 f) dargestellt, führt nicht zum Erfolg. Diese Modellierung kann nicht einen gewünschten Mindestabstand zwischen zwei Beschriftungen garantieren, da Überlappungen zwischen Beschriftungsräumen und zwischen Beschriftungsraum und Labelbox während des Auswahlprozesses zulässig sind. In Abbildung 21 h) ist eine mögliche, zulässige Beschriftungssituation nach dem Auswahlalgorithmus dargestellt. Die grafische Trennbarkeit der Beschriftungen kann nicht sichergestellt werden und das Ergebnis ist analog zu dem in Abbildung 21 b) dargestellt.

Im Folgenden wird daher von einer „etwas zu großen“ Labelbox, wie in Abbildung 21 c) dargestellt, ausgegangen, deren „Übergröße“ bei der Erzeugung der Labelbox zum Referenzmaßstab automatisch aus kartographischen Vorgaben einfließt.

4.3.1.1.2 Sonderfälle

Durch die Abstraktion von möglichen Beschriftungspositionen auf den Beschriftungsraum für die Datenstruktur des Konfliktgraphen können auch Beschriftungen für *Sonderfälle*, wie z.B. Berggipfel oder Küstenorte (Abbildung 22), mit dem Konzept für die Beschriftung punkthafter Objekte berücksichtigt werden. Für diese Sonderfälle müssen Verfahren zur

Bestimmung der Labelbox und der Boundingbox angegeben werden, genauso wie für die Beschriftungsschwierigkeit und den Auswahlalgorithmus (Abschnitt 5.2).

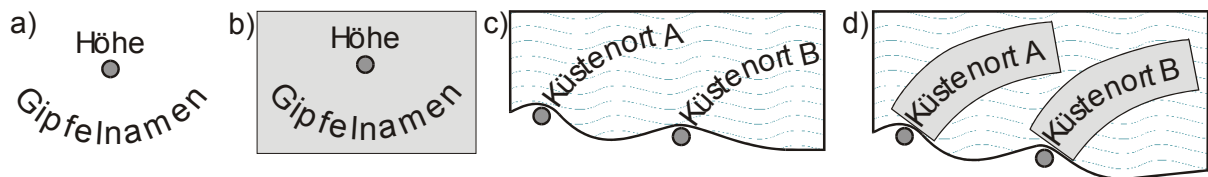


Abbildung 22 Beschriftung für Gipfel a) und Küstenorte c) mit zugehörigem Beschriftungsraum (Boundingbox) b) bzw. d).

4.3.2 Beschriftungskonflikt: punkthafes – punkthafes Objekt

Im vorherigen Abschnitt wurden die Konfliktgraphknoten, die zu beschriftenden Objekte und ihre Beschriftungsräume behandelt. Hier werden die zugehörigen Konflikte zwischen möglichen Beschriftungsräumen, wenn vorhanden, ermittelt, was zu einer ersten Version des Konfliktgraphen für punkthafte Objekte führt.

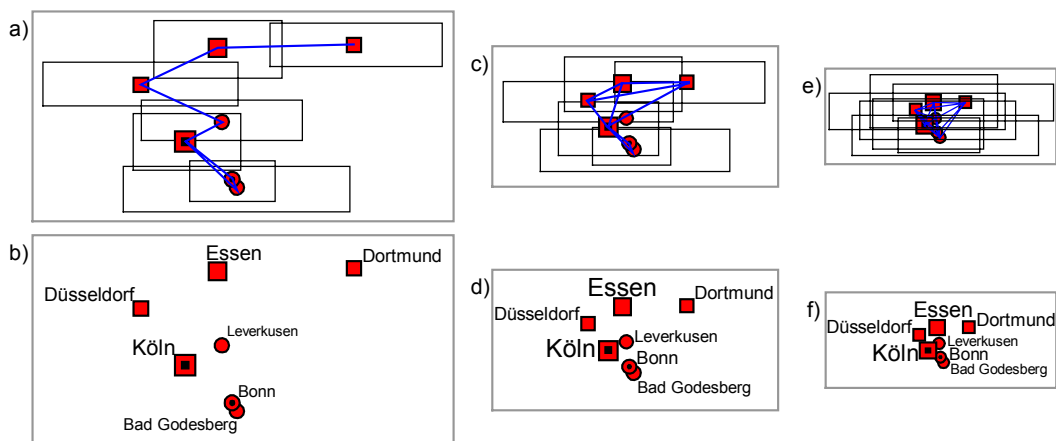


Abbildung 23 Statische Konfliktgraphen, dargestellt in drei Maßstäben a), c) und e) sowie mögliche Beschriftungen in b), d) und f). Es gibt einen (kleinen) Maßstab, bei dem es zwischen jedem zu beschriftenden Objekt einen Konflikt gibt e). Dies führt zu schlecht lesbaren Karten f).

Ausgehend von einem festen Maßstab, welcher einen statischen Konfliktgraphen für diesen Maßstab zur Folge hat, werden die Beschriftungsräume aller Konfliktgraphknoten gegenseitig auf Schnitt untersucht. Gibt es einen Schnitt, dann wird dieser durch eine Konfliktgraphkante zwischen den zugehörigen Konfliktgraphknoten repräsentiert (Abbildung 23 a)). Für die Ermittlung der Schnitte können die aus der Computergeometrie bekannten Standardverfahren, wie z.B. Scan-Line, benutzt werden [PRS85].

Dieser Ansatz lässt sich nicht direkt für die Herleitung des reaktiven Konfliktgraphen erweitern, da es eine Erweiterung eines 2-dimensionalen Problems (Fläche) auf ein 3-dimensionales (Fläche + Maßstab) ist.

Es stellt sich die Frage, wie der Maßstab bestimmt werden kann, bei und unter dem es zu einem Schnitt zwischen Beschriftungsräumen kommt. Das Durchlaufen des gesamten Maßstabsbereiches (vom größten zum kleinsten) [1:1 bis 1:∞], in z.B. 1er-Schritten (...1:48.345, 1:48.346, 1:48.347,...) aber auch die Verwendung einer Intervallschachtelung ist zu ineffizient. Es ist schwer zu ermitteln, welches der größte Maßstab ist, der noch einen

Konflikt verursacht. Bei jedem zu testenden Maßstab müssen alle Beschriftungsräume gegenseitig auf Schnitt getestet werden.

Daran schließt sich das folgende Problem an: Wenn ein potentieller Konflikt zwischen zwei zu beschriftenden Objekten in einem Maßstab erkannt wurde, bedeutet das, dass dieser Konflikt auch bei jedem kleineren Maßstab auftritt (Abbildung 23 e)). Dieses führt dazu, dass es einen Maßstab gibt, bei dem jedes zu beschriftende Objekt mit jedem anderen einen potentiellen Konflikt hat.

4.3.2.1 Schnittmaßstab

Bevor das Problem der Ermittlung des Schnittmaßstabs behandelt wird, erfolgt die Definition des Begriffs.

Definition (Schnittmaßstab):

Der Schnittmaßstab ist der Maßstab, ab und unterhalb dem es zu einem Konflikt zwischen den Beschriftungsräumen zweier zu beschriftender Objekte kommt.

Das erste Problem, die Ermittlung des Maßstabs, ab dem erstmals ein Konflikt auftritt, lässt sich durch eine Änderung der Strategie lösen. Bei der im letzten Abschnitt beschriebenen Vorgehensweise werden Maßstäbe durchlaufen und es wird getestet, ob es zu Konflikten zwischen beliebigen Objekten kommt. In 4.3.1.1 wurde vorgestellt, wie mit der Referenzlabelbox und dem Referenzmaßstab eine dem Maßstab entsprechende Labelbox bzw. Boundingbox erzeugt wird. Damit und mit der Diskussion über Koordinaten in 4.1.3 kann der *Schnittmaßstab* analytisch bestimmt werden, unterhalb dem es einen Konflikt zwischen zwei zu beschriftenden Objekten gibt. Dies führt zu einer neuen Vorgehensweise. Anstelle die Maßstäbe zu durchlaufen und dann auf Schnitte zu testen, kann so zwischen allen möglichen Paaren an zu beschriftenden Objekten der Schnittmaßstab bestimmt werden. Diese mathematisch, analytische Vorgehensweise führt zu einer offensichtlichen Effizienzsteigerung.

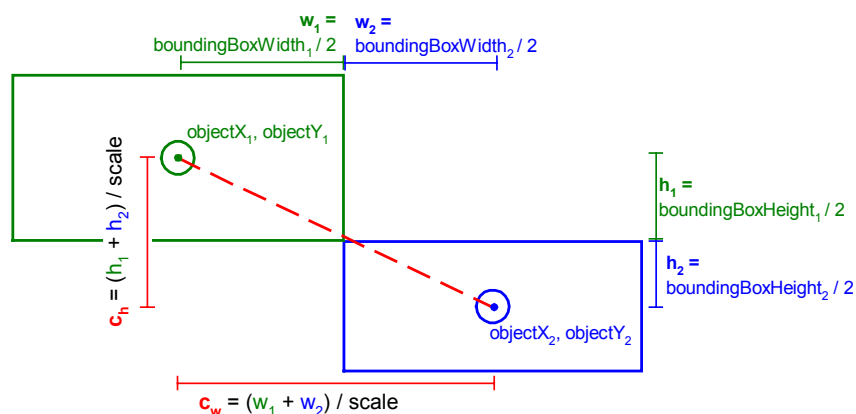


Abbildung 24 Berechnung des Schnittmaßstabs zwischen zwei punkthaften zu beschriftenden Objekten (der Abbildungsmaßstab ist gleich dem Schnittmaßstab).

In Abbildung 24 ist die Situation zur Bestimmung des Schnittmaßstabs anhand von zwei zu beschriftende Objekte mit ihren Beschriftungsräumen dargestellt. Die Beschriftungsräume sind parametrisiert mit den Größen aus der Referenzbeschriftung (*fixHeight*, *fixSymbolHeight* bzw. *fixWidth*, *fixSymbolWidth* und *fixDistanceSymbolLabel*) und dem zugehörigen Referenzmaßstab (*fixScale*) sowie dem Maßstab (*scale*).

Unter Ausnutzung der Achsenparallelität der Beschriftungsräume und des Abstands zwischen den zu beschriftenden Objekten (in Abbildung 24 c_h bzw. c_w) kann der Schnittmaßstab hergeleitet werden. Dazu werden getrennt für die y- und x-Richtung die Schnittgleichungen aufgestellt ($(h_1 + h_2)/scale = c_h$ und $(w_1 + w_2)/scale = c_w$). Beide Gleichungen sind zum einen mit Referenzbeschriftungsraum und -maßstab, sowie mit dem Maßstab parametrisiert. Da die Distanzen zwischen den Objekten (c_h und c_w) im Ursprungsmaßstab (Realwelt) sind, müssen diese mit dem Zielmaßstab multipliziert werden, um in den gleichen Maßstab, wie die Ausdehnungen der Boundingboxen, zu kommen. Aufgelöst nach dem Maßstab erhält man den Schnittmaßstab getrennt für y- und x-Richtung. Der kleinere dieser Schnittmaßstäbe ist der gesuchte Schnittmaßstab, ab und unter dem sich die beiden Beschriftungsräume schneiden.

Im Folgenden wird der Schnittmaßstab für die y-Richtung bestimmt.

Für c_h aus Abbildung 24 ergibt sich unter Verwendung von Gleichung (4.12) folgendes (aus Gründen der Übersichtlichkeit wurde auf die Darstellung der Parameter für die Funktionen h_1 und h_2 verzichtet):

$$c_h = \frac{h_1 + h_2}{scale_h} = \frac{\left(fixHeight_1 + fixDistanceSymbolLabel_1 + \frac{fixSymbolHeight_1}{2} \right) * \left(\frac{scale_h}{fixScale_1} \right)^{value}}{scale_h} + \frac{\left(fixHeight_2 + fixDistanceSymbolLabel_2 + \frac{fixSymbolHeight_2}{2} \right) * \left(\frac{scale_h}{fixScale_2} \right)^{value}}{scale_h} \quad (4.13)$$

durch ausklammern des Maßstabs und Auflösung nach ihm ergibt sich:

$$scale_h = \left(\frac{\frac{c_h}{\frac{fixHeight_1 + fixDistanceSymbolLabel_1 + \frac{fixSymbolHeight_1}{2}}{fixScale_1^{value}} + \frac{fixHeight_2 + fixDistanceSymbolLabel_2 + \frac{fixSymbolHeight_2}{2}}{fixScale_2^{value}}}}{\frac{1}{value-1}} \right) \quad (4.14)$$

Analog für die x-Richtung ergibt sich:

$$scale_w = \left(\frac{\frac{c_w}{\frac{fixWidth_1 + fixDistanceSymbolLabel_1 + \frac{fixSymbolWidth_1}{2}}{fixScale_1^{value}} + \frac{fixWidth_2 + fixDistanceSymbolLabel_2 + \frac{fixSymbolWidth_2}{2}}{fixScale_2^{value}}}}{\frac{1}{value-1}} \right) \quad (4.15)$$

Ein Wert größer gleich Eins für $scale_h$ oder $scale_w$ bedeutet, dass eine Überlappung im Horizontalen bzw. Vertikalen im Maßstab 1:1 vorliegt. Da ein Maßstab im Intervall $[1..0[$ liegt, wird der zugehörige Wert auf Eins gesetzt.

Der kleinere für die getrennt in y- und x-Richtung ($scale_h$ und $scale_w$) ermittelten Schnittmaßstäben ist derjenige, ab und unterhalb kleinerer Maßstab dem es zu einer Überlappung zwischen den Beschriftungsräumen der beiden zu beschriftenden Objekte kommt: $Schnittmaßstab = \min(scale_h, scale_w)$

4.3.2.2 Deselektionsmaßstab

Nachdem der Schnittmaßstab bestimmt ist, wird in diesem Abschnitt das zweite Problem angegangen: Unterhalb eines Maßstabs, dem kleinsten Schnittmaßstab, kommt es zwischen jedem zu beschriftenden Objekt zu einem Konflikt der Beschriftungsräume, was an der Maß-

stabsunabhängigkeit des Konfliktgraphen liegt (Abbildung 26, der rotunterlegte Maßstabsstrahl). Während es z.B. in Abbildung 25 bei einem Maßstab 1:100.000 keine Beschriftungskonflikte zwischen den Orten Bonn, Bornheim und Köln gibt, treten diese ab und unterhalb eines Maßstabs von 1:3.000.000 auf. Es ist offensichtlich, dass in dem kleinen Maßstab versucht werden sollte, die Orte Bonn und Köln zu beschriften, während Bornheim auf Grund des Platzmangels und der Priorität bei diesem Maßstab nicht berücksichtigt wird. D.h. Bonn und Köln sind Konfliktgraphknoten im Maßstab 1:100.000 und 1:3.000.000 und folglich sollte versucht werden beide Orte bei beiden Maßstäben zu beschriften. Bornheim hingegen ist nur im Konfliktgraph bei dem Maßstab 1:100.000 vertreten und ist kein Knoten im Maßstab 1:3.000.000, da seine Priorität zu niedrig ist, d.h. er ist bei diesem Maßstab deselektiert.

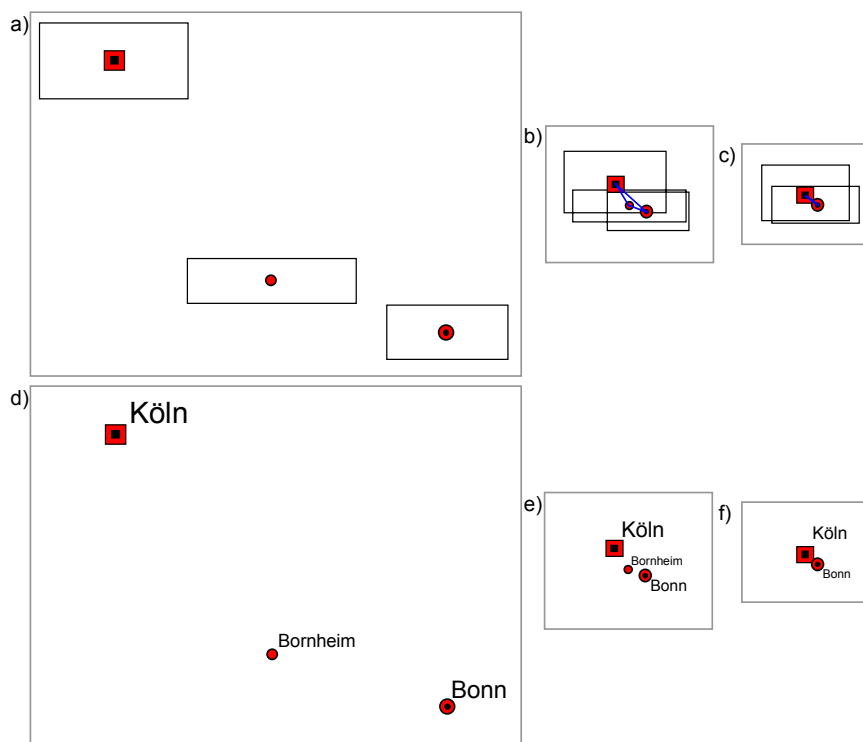


Abbildung 25 Deselektion von zu beschriftenden Objekten: Statische Konfliktgraphen in den Maßstäben 1:100.000 a) und d), 1:1.000.000 b) und e), 1:3.000.000 c) und f). Bei dem kleinsten Maßstab c) und f) ist Bornheim deselektiert.

Die Diskussion macht die Abhängigkeit zwischen dem Deselektionsmaßstab der Kante und der Deselektionsmaßstäbe der inzidenten Knoten deutlich. Nachdem die Schnittmaßstäbe aller Konflikte bestimmt sind (vgl. Abschnitt 4.3.11 Generierung des reaktiven Konfliktgraphen), werden die Deselektionsmaßstäbe für die Knoten (zu beschriftende Objekte) abgeleitet, die wiederum den Deselektionsmaßstab für den Konflikt ergeben (Abbildung 26 aus *deselect* der Knoten 1 und Knoten 2 ergibt sich die *lowerBound* des Konflikts – linker gestrichelter Pfeil).

Definition (Deselektionsmaßstab einer Konfliktgraphkante – `lowerBound`):

Der Deselektionsmaßstab der Konfliktgraphkante ist der Maßstab, ab und unter dem der Konflikt nicht mehr auftritt, da einer der beiden Konfliktpartner (Konfliktgraphknoten) nicht mehr beschriftet wird. Er ergibt sich aus dem größeren der beiden Deselektionsmaßstäbe der inzidenten Konfliktgraphknoten.

Der Deselektionsmaßstab des Knotens ist abhängig von der Beschriftungsschwierigkeit des betrachteten Knotens, d.h. vom zu beschriftenden Objekt und dem Maßstab (siehe die

folgenden Abschnitte). Die inzidenten Knoten vererben ihre Deselektionsmaßstäbe ihrer (gemeinsamen) Kante, d.h. dem gemeinsamen Konflikt. Der kleinere der beiden Deselektionsmaßstäbe wird zum Deselektionsmaßstab der Kante bzw. des Konflikts.

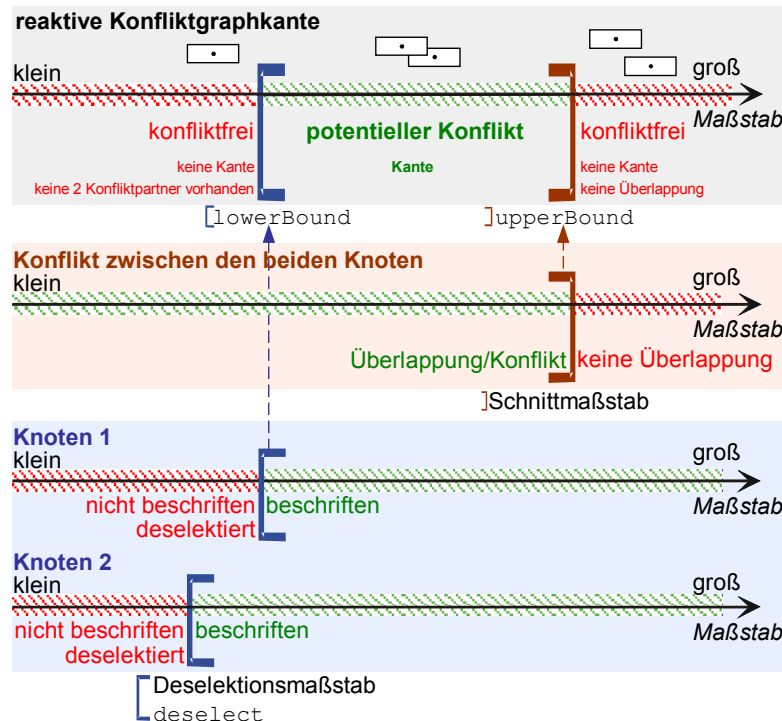


Abbildung 26 Das Maßstabsintervall für eine Konfliktkante (grau unterlegt) ergibt sich aus den Deselektionsmaßstäben der inzidenten Knoten (blau unterlegt) und dem Schnittmaßstab (rotbraun unterlegt).

4.3.3 Beschriftungsschwierigkeit und Deselektion: punkthafes Objekt

Die Entwicklung von Kriterien, die maßstabsabhängig entscheiden, ob versucht werden soll, ein Objekt in einem Maßstab zu beschriften oder vom Beschriftungsprozess auszuschließen, d.h. zu deselektieren, ist das Ziel des Verfahrens, das in den nachfolgenden Abschnitten verfolgt wird. Dabei wird ein *Deselektionsmaßstab* für jedes zu beschriftende Objekt bestimmt (Abbildung 26, blauunterlegte untere Maßstabsstrahlen). Für Maßstäbe, die größer als der Deselektionsmaßstab sind, wird versucht das zugehörige Objekt zu beschriften, unterhalb des Deselektionsmaßstabs wird das Objekt nicht beschriftet.

Definition (Deselektionsmaßstab eines Konfliktknoten):

Der Deselektionsmaßstab eines Konfliktknoten ist der Maßstab, ab und unterhalb dessen ein zu beschriftendes Objekt nicht mehr beschriftet wird.

Er wird mit Hilfe der Beschriftungsschwierigkeit während der Generierung des Konfliktgraphen festgelegt.

Definition (Beschriftungsschwierigkeit):

Die Beschriftungsschwierigkeit beschreibt, wie schwer es ist, ein Objekt zu einem vorgegebenen Maßstab und unter Berücksichtigung einer Menge von Konfliktpartnern zu beschriften. Dies kann entweder über eine Bewertung der Konflikte oder über die verbleibenden Beschriftungsmöglichkeiten erfolgen.

Der Deselektionsmaßstab ist abhängig von der Priorität des Objekts in einem bestimmten Maßstab bezüglich anderen Objekte zu denen potentielle Konflikte bestehen, sowie kartographischen Gestaltungsregeln. Imhof beschreibt diese wie folgt: „Die Karte soll nicht gleichmäßig mit Namen überdeckt werden. Andererseits aber sind dichte Namensknäuel zu vermeiden. Dies ist sowohl bei der Namensauswahl wie auch bei der Namenordnung zu beachten.“ [IMH62]

Die Priorität eines Objektes wird aus den zugehörigen Objektdaten entnommen bzw. abgeleitet. Als Prioritätskriterium für Städte kann z.B. die Einwohnerzahl genommen werden. Abhängig von dem gewünschten Kartentyp ist auch eine Kombination von verschiedenen Attributen denkbar, z.B. aus administrativer Bedeutung und Einwohnerzahl.

Im Gegensatz zu Signaturen, die real existierende Objekte mit Raumbezug repräsentieren und dementsprechend fixe Positionen in Karten haben, hat die Beschriftung große Freiheitsgrade. Das Schriftobjekt für punkthafte Objekte kann in der Regel ringsum die zugehörige Signatur platziert werden. Dieser Platz wird durch die Boundingbox repräsentiert. Diese nimmt einen vielfachen Raum der letztendlich genutzten Fläche ein. Hat eine Boundingbox keine Konflikte mit anderen, dann gibt es keinen Grund das zugehörige Objekt aus Platzgründen nicht zu beschriften.

Das Deselektionskriterium ist zentraler Bestandteil des Konfliktgraphen. Zu beschriftende Objekte werden als Knoten im reaktiven Konfliktgraphen repräsentiert. Der Deselektionsmaßstab ist eines der Knotenattribute. Für die Beschriftung eines Kartenausschnitts wird ein für diesen spezifischen Maßstab abgeleiteter statischer Konfliktgraph genutzt. Knoten aus dem reaktiven Konfliktgraphen, deren Deselektionsmaßstab kleiner als der Maßstab des statischen Konfliktgraphen sind, werden in diesen übernommen. Alle Knoten des statischen Konfliktgraphen werden vom Auswahlalgorithmus versucht zu beschriften.

Wie verhält es sich, wenn es zu mehreren Konflikten kommt? In den folgenden Abschnitten werden unterschiedliche Bewertungskriterien und ihre Verwendung als Deselektionskriterium vorgestellt. Das Kriterium in 4.3.3.1 ist auf die Anzahl der potentiellen Konflikte (den Konfliktgrad) beschränkt. Im Gegensatz dazu werden in den Verfahren, die in 4.3.3.2 und 4.3.3.3 entwickelt werden, unter Berücksichtigung der räumlichen Verteilung eine reziproke Aussage gemacht, in dem der verbleibende Beschriftungsraum ermittelt und beurteilt wird. Zum Schluss erfolgt eine kurze Analyse der vorgestellten Verfahren.

4.3.3.1 Knotengrad – Anzahl der Konflikte

Als erster Anhaltspunkt kann eine quantitative Bewertung dienen, die Anzahl der Konflikte in Kombination mit der Priorität zu einem vorgegebenen Maßstab. Die ersten beiden Informationen stehen im zugehörigen Konfliktgraphknoten und können effizient aus ihm gewonnen werden. Das führt zu dem Deselektionskriterium, eine maximale Anzahl an Konflikten mit wichtigeren Objekten (*maxHigher*) zuzulassen. Algorithmus 1 skizziert dieses Verfahren.

Angewandt auf das Beispiel in Abbildung 27 wird im Folgenden Algorithmus 1 zeilenweise vorgestellt. Daraus entstehen die in Tabelle 1 und Tabelle 2 dargestellten Maßstäbe, die schließlich zu dem in Abbildung 28 dargestellten Konfliktgraphen führen bei einem maximalen Knotengrad von drei zu höher prioritären Objekten.

Abbildung 27 zeigt die zu beschriftenden punkthafte Objekte, nach ihrer Priorität, den Einwohnerzahl aufsteigend sortiert, ergibt sich: Bad Godesberg, Leverkusen, Bonn, Düsseldorf, Dortmund, Essen und Köln (Tabelle 1 erste Spalte bzw. erste Zeile).

```

1  Input: zu beschriftende punkthafte Objekte mit Lage, Referenzbeschriftung und
        Priorität sowie maxHigher Wert für maximale Anzahl an Konflikten mit
        wichtigeren Objekten
Output: reaktiver Konfliktgraph (mit Knoten und Kanten) Deselektionskriterium:
5  maximale Anzahl an Konflikten mit wichtigeren Objekten
    // erzeuge Konfliktgraphkanten zu Objekten mit höherer Priorität und bestimme
    // den Deselektionsmaßstab für den betrachteten Konfliktgraphknoten
    für jedes zu beschriftende Objekt (Knoten) aufsteigend nach Priorität:
    erzeuge Konfliktgraphknoten und Beschriftungsraum zu aktuellem Objekt
10  berechne Schnittmaßstab zu jedem wichtigeren Objekt (höhere Priorität)
    sortiere diese Objekte absteigend nach Schnittmaßstab
    nimm die ersten maxHigher Objekte (die mit dem größten Schnittmaßstab)
    erzeuge Konfliktgraphkanten:
        zwischen aktuellem Objekt (Knoten) und den maxHigher ermittelten
15  setze upperBound auf zugehörigen ermittelten Schnittmaßstab
    setze Knoten-Attribut (Deselektionsmaßstab):
        setze deselect auf Schnittmaßstab mit dem maxHigher+1 Objekt (+ε)
        wenn das maxHigher+1 Objekt existiert andernfalls auf 0
    // behandle Sonderfall: Konfliktgraphknoten mit gleicher Priorität
20 für jedes Paar an zu beschriftenden Objekten mit gleicher Priorität:
    berechne zu jedem Paar den Schnittmaßstab
    erzeuge Konfliktgraphkanten zu jedem Paar, wenn der Schnittmaßstab größer
        ist als die beiden deselect-Maßstäbe der Objekte
        und setze upperBound auf den ermittelten Schnittmaßstab
25 // bestimme die fehlenden Attribute der Konfliktgraphkante und
    // teste die Konfliktgraphkante bzw. Maßstabsintervall auf (Ir-)Relevanz
    für jede Konfliktgraphkante:
        setze Attribut lowerBound der Kante auf den größeren deselect-Maßstab
        der beiden inzidenten Knoten (Objekte)
30 eliminiere Kante deren Attribut upperBound kleiner ist als das Attribut
    lowerBound

```

Algorithmus 1 Ermittlung der Konflikte und Generierung der Konfliktgraphknoten sowie der Konfliktgraphkanten mit Knotengrad und Priorität als Deselektionskriterium.

▪ Algorithmus 1 – Zeile 6 bis 18

Jeder Schleifendurchlauf von Algorithmus 1 Zeile 8 bis 18 erzeugt für das Beispiel aus Abbildung 27 je einen Konfliktgraphknoten für das betrachtete Objekt und je eine Spalte in Tabelle 1 und Tabelle 2 (von links nach rechts), wie im Folgenden beschrieben.

Beginnend bei dem Ort mit der niedrigsten Priorität, Bad Godesberg, werden die Konfliktmaßstäbe zu allen Orten mit einer höheren Priorität bestimmt (Algorithmus 1 – Zeile 8 und 10; Tabelle 1 zweite Spalte). Diese Objekte werden absteigend nach ihrem Schnittmaßstab sortiert (Algorithmus 1 – Zeile 11). In diesem Beispiel sollen maximal drei Konflikte mit Orten höherer Priorität zugelassen werden (*maxHigher* = 3). Zu den Orten bzw. Konfliktgraphknoten mit den drei (Wert von *maxHigher*) höchsten Schnittmaßstäben werden Konfliktgraphkanten erzeugt (Algorithmus 1 – Zeile 12 und 13), d.h. zu Bonn, Köln und Leverkusen. Das Attribut *upperBound* der Konfliktgraphkante, das angibt, unterhalb welchen Maßstabs ein potentieller Konflikt eintritt, wird auf den zugehörigen ermittelten Schnittmaßstab gesetzt (Algorithmus 1 – Zeile 15) $Bad\ Godesberg \leftrightarrow Bonn.upperBound = 1:61.000$, $Bad\ Godesberg \leftrightarrow Köln.upperBound = 1:439.000$ und $Bad\ Godesberg \leftrightarrow Leverkusen.upperBound = 1:846.000$.

Durch die Einschränkung auf eine maximale Anzahl an Konflikten entsteht daraus ein *Deselektionsmaßstab* (*deselect*) (Algorithmus 1 – Zeile 16 bis 18). Das ist der Maßstab, unterhalb dessen das zugehörige Objekt nicht mehr beschriftet und gegebenenfalls nicht mehr dar-

gestellt wird. Dieser Maßstab ist der Schnittmaßstab mit dem $\text{maxHigher}+1$ -ten Objekt mit höherer Priorität aus der nach dem Schnittmaßstab sortierten Liste. Damit es nur zu Konflikten mit den ersten maxHigher Objekten kommt und nicht mit dem $\text{maxHigher}+1$ -ten, dessen Schnittmaßstab zur Bestimmung des Deselektionsmaßstabs des Objektes verwendet wird, muss dieser um einen ε -Wert erhöht werden, der Deselektionsmaßstab für Bad Godesberg ist $1:2.247.000$ bzw. $1:2.247.001$ damit der Konflikt mit Düsseldorf nicht mehr eintritt (Tabelle 1 zweite Spalte und Tabelle 2 zweite Spalte).

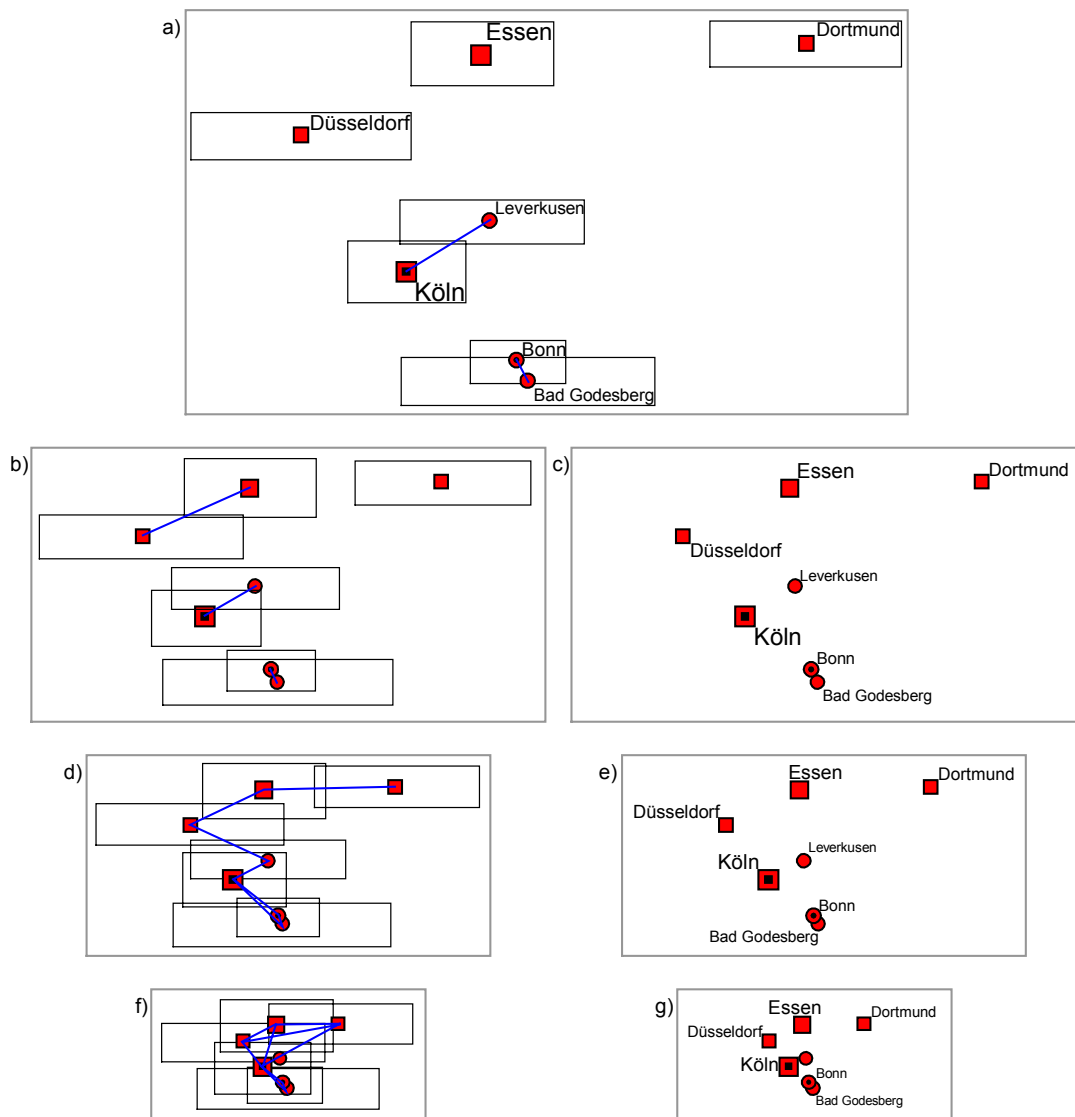


Abbildung 27 Statische Konfliktgraphen zu vier Maßstäben, wobei maximal drei Konflikte zu Objekten höherer Priorität zulässig sind – Deselektion von Leverkusen in f) bzw. g) (vgl. Tabelle 2).

Die verbleibenden Orte werden in derselben Weise nach ihrer Priorität abgearbeitet (Algorithmus 1 – Schleife in den Zeilen 8 bis 18; verbleibende Spalten in Tabelle 1 und Tabelle 2). Die Deselektionsmaßstäbe von Düsseldorf, Dortmund, Essen und Köln werden auf 0 gesetzt, da sie mit drei oder weniger wichtigeren Objekten in Konflikt stehen und daher immer versucht werden zu beschriften.

- Algorithmus 1 – Zeile 19 bis 24

Da in diesem Beispiel die beiden Objekte Düsseldorf und Dortmund die gleiche Priorität besitzen, wird ihr potentieller Konflikt, durch die zweite Schleife (Algorithmus 1 – Zeile 20 bis 24) erkannt und die zugehörigen Konfliktgraphkante erzeugt. Die Deselektionsmaßstäbe für das Beispiel sind in Tabelle 2 aufgelistet.

Im Gegensatz zu den potentiellen Konflikten gibt es für die zu beschriftenden Objekte, den Konfliktgraphknoten, keine obere Grenze, denn das würde bedeuten, dass das Objekt erst unterhalb eines bestimmten Maßstabs dargestellt bzw. beschriftet wird.

Das führt zu einem Konfliktgraphen, dessen Kanten mit der `upperBound` attribuiert sind (Abbildung 28 b)) und dessen Knoten mit dem Deselektionsmaßstab (Abbildung 28 c)).

- Algorithmus 1 – Zeile 25 bis 31

Nachdem alle zu beschriftenden Objekte (Konfliktgraphknoten) abgearbeitet sind, müssen die ermittelten potentiellen Konflikte (Konfliktgraphkanten) durchlaufen und geprüft werden (Algorithmus 1 - Zeile 27 bis 31). Dazu wird zu jedem potentiellen Konflikt (Konfliktgraphkante) der Maßstab hergeleitet, ab dem der Konflikt nicht mehr auftritt. Das ist genau dann der Fall, wenn einer der beiden Konfliktpartner nicht mehr beschriftet wird. Für jede Konfliktgraphkante wird die untere Maßstabsgrenze (`lowerBound`) auf den höheren Deselektionsmaßstab der beiden inzidenten Knoten gesetzt (Algorithmus 1 - Zeile 28 und 29). Eine Konfliktgraphkante bzw. ein potentieller Konflikt kann nur dann eintreten, wenn die `lowerBound` kleiner als die `upperBound` ist (Algorithmus 1 – Zeile 30 und 31).

Das Ergebnis ist der in Abbildung 28 d) dargestellte reaktive, attributierte Konfliktgraph.

erster Konflikt	Bad Godesberg	Leverkusen	Bonn	Düsseldorf	Dortmund	Essen	Köln
Bad Godesberg							
Leverkusen	1: 1.404.000						
Bonn	1: 100.000	1:1.262.000					
Düsseldorf	1: 2.247.000	1: 659.000	1:2.140.000				
Dortmund	1: 3.367.000	1:1.653.000	1:3.306.000	1:930.000			
Essen	1: 2.805.000	1:1.307.000	1:2.731.000	1:494.000	1: 681.000		
Köln	1: 681.000	1: 274.000	1: 555.000	1:904.000	1: 1.740.000	1:1.440.000	

Tabelle 1 Schnittmaßstab zu Objekten höherer Priorität. Kursiv sind die Schnittmaßstäbe dargestellt, die wegen der Überschreitung der maximal zulässigen Anzahl an Konflikten mit Objekten höherer Priorität, in diesem Beispiel drei, nicht mehr erreicht werden.

Konfliktgraphknoten	Bad Godesberg	Leverkusen	Bonn	Düsseldorf	Dortmund	Essen	Köln
Deselektionsmaßstab	1:2.247.001	1:1.307.001	1:3.306.001	0	0	0	0

Tabelle 2 Deselektionsmaßstab unterhalb dem das zugehörige Objekt nicht mehr beschriftet wird – 0 wenn das Objekt bei jedem Maßstab beschriftet wird.

Das Kriterium bzw. der Algorithmus bestimmt zu Objekten ihren Deselektionsmaßstab, der Maßstab unterhalb dessen sie aus Konkurrenzgründen nicht mehr dargestellt werden, in Abhängigkeit von der Beschriftungsdichte und Priorität. Dabei ist im Einzelfall zu entscheiden ob nur die Schrift oder auch die zu beschriftende Signatur nicht dargestellt werden soll. Diese Entscheidung ist vom Informationsgehalt der Signatur ohne Schrift abhängig. Das führt dazu, dass z.B. bei einer Deutschlandkarte im Ruhrgebiet nur die größeren Städte dargestellt und beschriftet werden und gleichzeitig in schwächer besiedelten Gebieten, wegen der geringen Konkurrenz und dem größeren Raum, auch Objekte mit niedrigeren Prioritäten dargestellt werden. Bochum mit einer höheren Priorität (ca. 400.000 Einwohner) wird in einer Deutschlandkarte nicht mehr dargestellt während Vechta (ca. 25.000 Einwohner) noch dargestellt wird. Beschriftungs- und Auswahlregeln werden in [IMH62] und [ELL01] diskutiert.

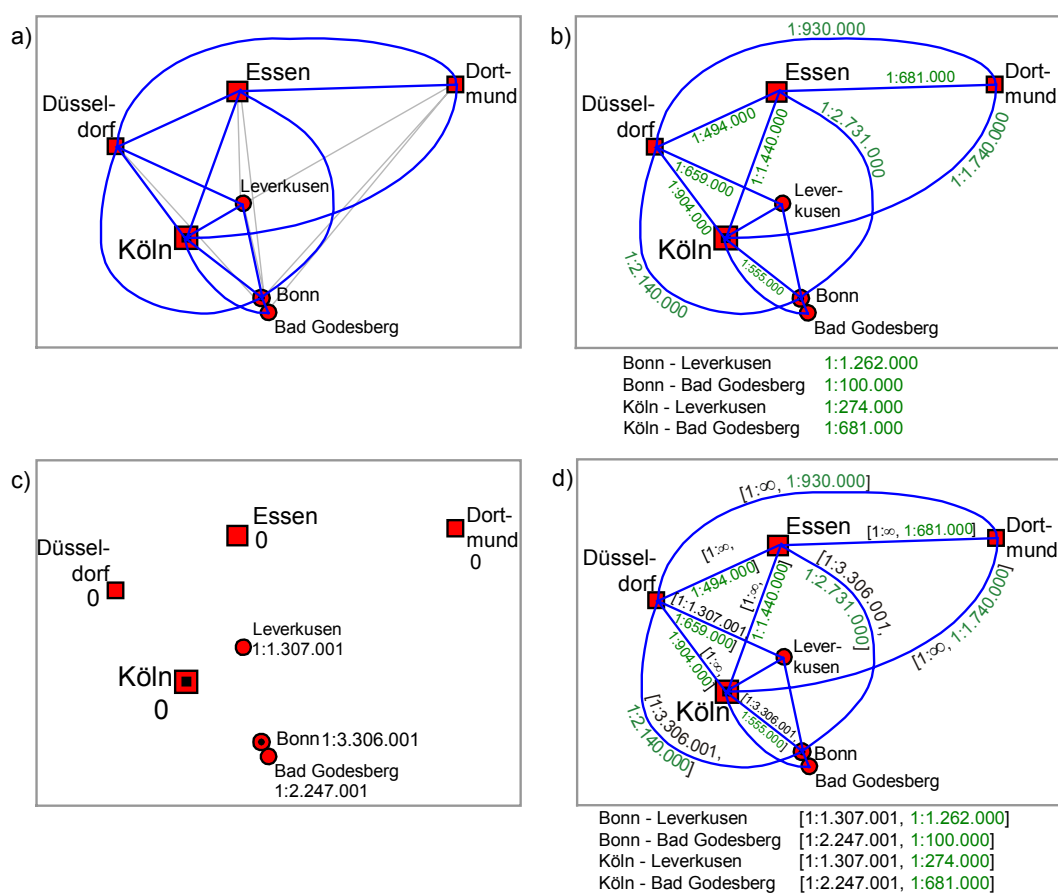


Abbildung 28 Reaktiver Konfliktgraph (vgl. Abbildung 27): a) blau relevante und grau irrelevante Kanten; b) mit $upperBound$ attribuierte Kanten; c) Deselektionsmaßstäbe der Knoten; d) Kanten mit Maßstabsintervall $[lowerBound, upperBound]$.

Durch die Betrachtung der möglichen Konflikte zu höher prioritären Objekten und den Konflikten zwischen Paaren von Objekten mit gleicher Priorität wird sichergestellt, dass alle möglichen Konflikte genau einmal und nicht doppelt erkannt werden. Gleichzeitig stellt das Kriterium eine Begrenzung der Konflikte zu Objekten mit *höherer Priorität* sicher, was zu der gewünschten Ausdünnung der zu beschriftenden Objekte abhängig vom Maßstab führt. Die maximale Anzahl an Konflikten, die ein Objekt haben kann, ist dadurch nicht begrenzt. Das führt dazu, dass Objekte mit höherer Priorität in der Regel eine größere Anzahl an Konflikten haben als Objekte mit niedriger. Dieses wird in Abbildung 28 a) bzw. d) deutlich. In dem

reaktiven Konfliktgraphen hat Leverkusen drei Konflikte, d.h. den Knotengrad drei, während das höher prioritäre Köln sechs Konflikte hat.

Der wesentliche Bestandteil des Deselektionskriteriums ist eine maßstabsabhängige Bewertungsfunktion für die Beschriftungsschwierigkeit. Das hier vorgestellte Kriterium verwendet dazu die Anzahl der potentiellen Konflikte mit Beschriftungsräumen wichtigerer Objekte, die effizient aus dem Konfliktgraphen abgeleitet werden können, der Knotengrad. Im Folgenden soll zur übersichtlicheren Darstellung von dem Kriterium *höhere Priorität* abgesehen werden, das mit den entwickelten Kriterien kombinierbar ist.

Die Konfliktsituation in Abbildung 27 d) bzw. e) zwischen den Objekten Dortmund, Essen, Düsseldorf und Köln soll nun näher betrachtet werden. Dabei hat das Objekt Essen bzw. der zugehörige Beschriftungsraum sowie der von Düsseldorf jeweils zwei Konflikte. Dabei ist offensichtlich, dass das Objekt Düsseldorf einfacherer zu beschriften ist als Essen. Die Ursache liegt in der räumlichen Verteilung der Konflikte aus Sicht der zu beschriftenden Objekte. Da die geometrische Lage bzw. Verteilung der Konflikte in dem vorgestellten Kriterium nicht berücksichtigt wird, sind für das hier beschriebene Kriterium beide Objekte gleich schwer zu beschriften. Im Folgenden wird ein Verfahren beschrieben, das dieses berücksichtigt.

4.3.3.2 Größter freier Winkel

Am Ende des letzten Abschnittes wurde an einem realen Beispiel beschrieben, zu welchen Fehlbewertungen es kommen kann, wenn die geometrische Lage der Konflikte zu den zu beschriftenden Objekten nicht berücksichtigt wird. In Abbildung 29 sind zwei Situationen dargestellt, die bei der Bestimmung der Beschriftungsschwierigkeit allein durch den Knotengrad nicht unterschieden werden können. In Abbildung 29 a) und b) haben die beiden roten zu beschriftenden Objekte jeweils vier Konflikte. Es ist offensichtlich, dass für die linke Situation problemlos eine konfliktfreie Beschriftungsposition gefunden werden kann, im Gegensatz zur rechten.

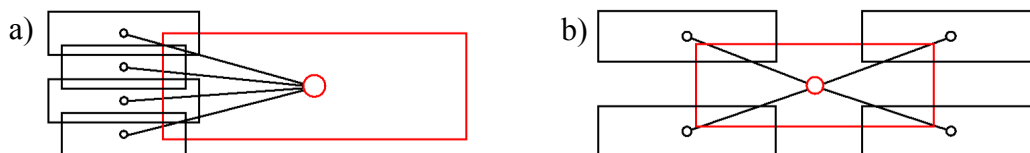


Abbildung 29 Der Knotengrad, d.h. der Konfliktgrad eines Knotens, beschreibt nicht ausreichend die Schwierigkeit, den zugehörigen Knoten zu beschriften. Das rote zu beschriftende Objekt hat in a) und b) den gleichen Konfliktgrad, offensichtlich ist jedoch das rote Objekt in a) leichter zu beschriften.

Für eine präzisere Bestimmung der Beschriftungsschwierigkeit bzw. der Beurteilung der Konflikte müssen geometrische Informationen einbezogen werden. Anschaulich ist das der erste Schritt zum Übergang vom Konfliktgraphen zum *Konfliktgebirge*.

Definition (Konfliktgebirge):

Das Konfliktgebirge ist der in den drei dimensional Raum eingebettete statische Konfliktgraph. Die Lage der Konfliktgraphknoten ergibt sich aus den Koordinaten der Geobjekte und die Höhe aus seinen Attributen Priorität und Beschriftungsschwierigkeit. Die Einbettung einer Konfliktgraphkante ergibt sich durch Interpolation aus Lage, Höhe und Beschriftungsraum der Konfliktgraphknoten.

Das *Konfliktgebirge* repräsentiert zum einen die zu beschriftenden Objekte und ihre potentiellen Konflikte und unterstützt zum anderen mit der z-Koordinate (der Höhe) den Auswahlalgorithmus bei der Auswahl der zu beschriftenden Objekte, bei der Reihenfolge der Abarbeitung dieser und bei der Platzierung der Beschriftung jedes einzelnen zu beschriftenden Objekts in seinem Beschriftungsraum. Diese Betrachtungsweise ist für die Beschriftung punkthafter Objekte zu einem festen Maßstab anschaulich. Die Integration zu beschriftender linienhafter Objekte, die – wie später beschrieben – durch Konfliktgraphknoten im Konfliktgraphen repräsentiert werden, führen zu einem Verlust der Anschaulichkeit.

Wie sollen die Konfliktgraphkanten in das Konfliktgebirge integriert bzw. zwischen den Konfliktgraphknoten im Konfliktgebirge interpoliert werden? Konfliktgraphkanten können nicht eins-zu-eins in das Konfliktgebirge übernommen werden. Ihr Anfangs- und Endpunkt, die inzidenten Knoten, besitzen eine geometrische Einbettung. Die Kante an sich, anschaulich die kürzeste Verbindung zwischen Anfangs- und Endpunkt, gibt nicht die genaue Lage und Ausdehnung des Konflikts wieder. Sie repräsentiert einen potentiellen Konflikt, der eine Überlappung von Beschriftungsräumen ist. Die Überlappung, d.h. die Schnittfläche der rechteckigen Beschriftungsräume ist wieder eine Fläche und folglich ist der Konflikt flächenhaft.

Da die Beschriftung der Objekte das primäre Ziel ist und die Konflikte dafür „das Mittel zum Zweck“ sind, werden diese im Folgenden aus Sicht der Objekte betrachtet und modelliert. Im Konfliktgebirge wird an jedem Knoten zu jeder inzidenten Kante ein Konfliktsektor angegeben (Abbildung 30). Aus Sicht der Konfliktknoten ist das der Sektor, in dem die Konfliktfläche liegt. Zu jeder Konfliktgraphkante gehören somit zwei Sektoren, deren Ausgangspunkte an beiden Enden liegen und deren Öffnungen entgegengesetzt sind. Durch Schnittbildung erhält man somit eine Approximation der Konfliktfläche.

Definition (Sektor):

Der Sektor eines Knotens ist die Fläche zwischen zwei Halbgraden, die in dem Knoten beginnen.

Definition (Konfliktsektor):

Der Konfliktsektor eines Konfliktknotens bezüglich einer Konfliktkante ist der Schnitt des kleinsten Sektors des Knotens, der die Überlappungsfläche der Beschriftungsräume der beiden zur Kante inzidenten Knoten enthält, mit dem Beschriftungsraum des Knotens.

Definition (größter freier Winkel eines Konfliktgraphknotens):

Der größte freie Winkel eines Konfliktgraphknotens ist der Winkel des größten Sektors, der disjunkt zu allen Konfliktsektoren ist.

Durch den größten freien Winkel kann die lokale Beschriftungsschwierigkeit genauer bestimmt werden. Je größer dieser größte freie Winkel ist, desto größer ist der konfliktfreie Beschriftungsraum des zugehörigen Objekts. Dadurch steigt die Wahrscheinlichkeit, eine konfliktfreie bzw. konfliktarme Beschriftungsposition zu finden. In Abbildung 30 b) sind die zugehörigen größten freien Winkel eingetragen.

Aus der Kombination des größten freien Winkels und der Priorität des zu beschriftenden Objekts leitet sich die Höhe des eingebetteten Knoten im Konfliktgebirge ab.

Bevor Details, sowie die Vor- und Nachteile der Modellierung diskutiert werden, beschreiben dieser und der folgende Absatz deren Nutzung. Die modellierten und daraus ableitbaren Informationen fließen in den Auswahlalgorithmus ein, der die zu beschriftenden Objekte auswählt und für jedes eine Beschriftungspositionen ermittelt bzw. es im Nachhinein

ausschließt. Wichtige zu beschriftende und schwer beschriftbare Objekte, die den geringsten Ausweichplatz haben, können so, wenn man die Beschriftungsreihenfolge abhängig von der Höhe der Objekte wählt, in der Beschriftungsreihenfolge bevorzugt bearbeitet werden. Mit dieser Modellierung wird das kartographische Vorgehen der Beschriftung vom Dichtezentrum der Karte zum Kartenrand umgesetzt.

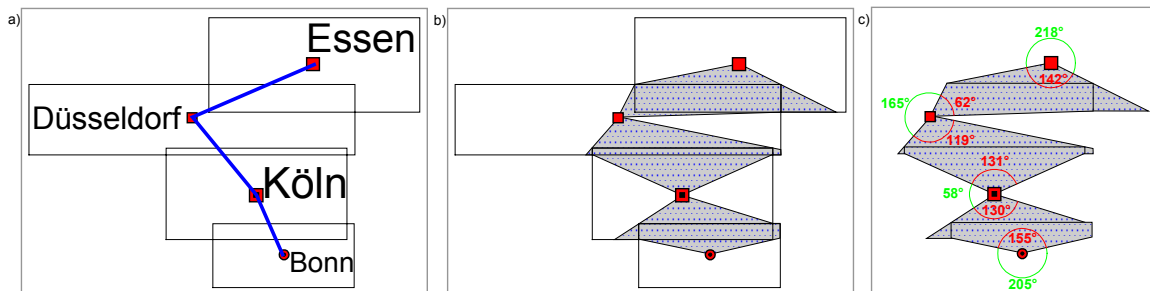


Abbildung 30 Größter freier Winkel als Beschriftungsschwierigkeitskriterium: a) zeigt das Konfliktszenario; b) und c) die Konfliktspektoren; in c) sind die größten freien Winkel grün dargestellt. Das Objekt Düsseldorf ist wegen des größeren freien Winkels (165°) leichter zu beschriften als Köln (58°), obwohl der Konfliktgrad gleich ist.

Wenn die Konfliktsflächen, die aus den zwei Konfliktspektoren zusammengesetzt sind, als Grate modelliert werden, können – anschaulich beschrieben – die Beschriftungsboxen an diesen seitlich herunterrutschen, von den Konflikten weg in die lokal konfliktärmsten Gebiete. Dieses beschriebene Verfahren unterstützt zum einen das Finden einer konfliktfreien bzw. konfliktarmen Beschriftungsposition. Gleichzeitig wird eine eindeutige Beschriftung gewährleistet, da Konfliktgebiete eine mehrdeutige Beschriftungszuordnung bedeuten.

Die beschriebene Modellierung und Vorgehensweise ist anschaulich, lässt sich jedoch, wie die folgende Diskussion zeigen wird, nicht umsetzen, da es einige Schwachpunkte gibt.

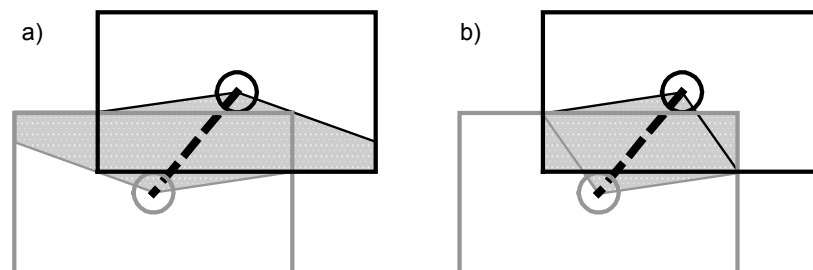


Abbildung 31 Konstruktion der Konfliktspektoren: In a) decken Konfliktspektoren mehr als die Konfliktsfläche ab; in b) weniger.

Abbildung 31 zeigt exemplarisch zwei Varianten für die Definition des Konfliktspektors. Da die Beschriftungsräume achsenparallel ausgerichtet sind, ist deren Schnitt ein ebenfalls achsenparalleles Rechteck. Dieses wird durch die Konfliktspektoren angenähert. Es liegt hier eine Approximation vor, da bis auf den Sonderfall, dass eine Ecke des Schnittrechtecks auf dem Knoten liegt (bzw. zwei zueinander diagonale Ecken des Schnittrechtecks auf den zugehörigen Konfliktsknoten liegen), es immer „Verschnitt“ gibt. Dieser Verschnitt kann, wie in Abbildung 31 a) dargestellt, zuungunsten der konfliktfreien Fläche sein oder, wie b) zeigt, die Konfliktsfläche nicht vollständig umschließen. In beiden Fällen tritt ein Verlust an Genauigkeit ein.

Neben diesem Genauigkeitsverlust existiert ein systematischer Fehler, der bei der Platzierung des Schriftobjekts innerhalb des Beschriftungsraums unter Nutzung der Konfliktsektoren und des größten freien Winkels auftritt. Die zu platzierenden Schriftobjekte können verlustfrei durch achsenparallele Rechtecke approximiert werden. Da die Begrenzungen der Konfliktsektoren bis auf den oben besprochenen Sonderfall nicht achsenparallel sind, tritt hier ein weiterer Verlust auf. Das führt dazu, dass konfliktfreie Beschriftungspositionen existieren können, die aber auf Grund der Modellierung als Konflikt markiert sind und dadurch nicht erkannt werden. Dieser Fehler kann bei beiden oben angesprochenen Konstruktionen der Konfliktsektoren auftreten (Abbildung 32 a) und b)). Wird der Konfliktsektor, wie in Abbildung 31 b) dargestellt, „optimistisch“ definiert, können aus dem größten freien Winkel vermeintliche konfliktfreie Beschriftungspositionen abgeleitet werden, die jedoch einen Konflikt besitzen (Abbildung 32 c)).

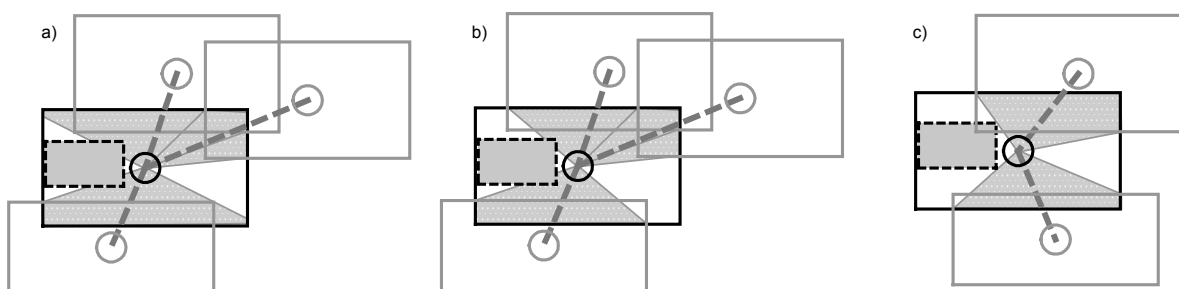


Abbildung 32 Die Ableitung von konfliktfreien Beschriftungspositionen mit dem Kriterium des größten freien Winkels ist fehlerbehaftet.

Aus Gründen der Vollständigkeit soll auf ein weiteres Problem bei der Modellierung der Konflikte bzw. dem Auffinden einer konfliktfreien Beschriftungsposition mit dem größten freien Winkel hingewiesen werden. In Abbildung 33 sind zwei Beschriftungs- bzw. Konfliktsituationen dargestellt. Obwohl der größte freie Winkel für die beiden schwarzen punkthaften Objekte in beiden Darstellungen gleich groß ist, folgt daraus nicht, wie man vermuten würde, dass die beiden Objekte gleich schwer zu beschriften sind bzw. dass beide eine konfliktfreie Beschriftungsposition besitzen oder beide keine. Neben dem größten freien Winkel muss seine Lage berücksichtigt werden. Die Geometrie der letztendlich zu platzierenden Labelbox, insbesondere die orthogonal ausgerichteten rechten Winkel, geht hier nicht explizit in das Verfahren ein.

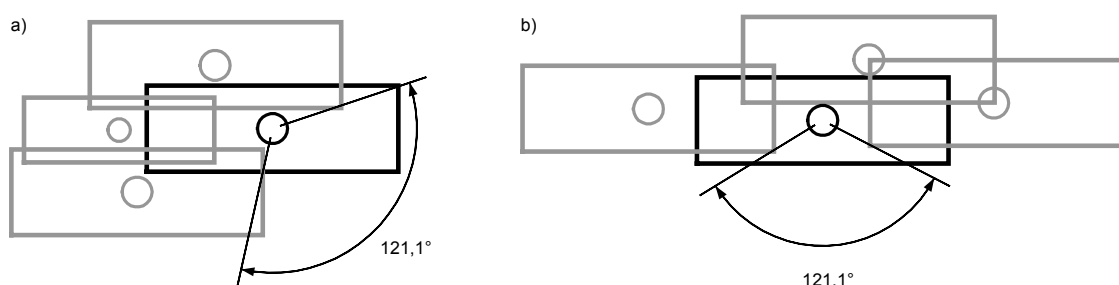


Abbildung 33 Abhängigkeit der Beschriftungsschwierigkeit des Kriteriums größter freier Winkel von der Lage der Winkel: a) ist einfacher zu beschriften als b).

Die Integration der Geometrie in die Bestimmung der Beschriftungsschwierigkeit und damit in das Deselektionskriterium ist, wie einleitend dargestellt, sinnvoll. Jedoch hat die Modellierung der Konflikte durch Konfliktsektoren und den größten freien Winkel, die zu

dem anschaulichen Konfliktgebirge für zu beschriftende punkthafte Objekte führt, Schwachstellen. Im nächsten Abschnitt wird ein Modell, das diese Schwachstellen nicht besitzt, vorgestellt, ebenso wie die Integration des geometrischen Kriteriums in die Generierung des reaktiven Konfliktgraphen.

4.3.3.3 Freie Beschriftungsfläche

Wie im vorherigen Abschnitt diskutiert, ist die Modellierung von Konflikten durch Konfliktsektoren bzw. die Unterstützung der Beschriftung durch den größten freien Winkel ungünstig, da die zu platzierende Labelbox nicht berücksichtigt wird. In diesem Abschnitt wird ein neues geometrisches Kriterium entwickelt, das diese Nachteile nicht aufweist und darüber hinaus die Ableitung geeigneter konfliktfreier oder -armer Beschriftungspositionen direkt unterstützt sowie eine Bewertung der Konfliktsituation.

Dazu werden lokal zu jedem zu beschriftenden Objekt die zugehörigen Konflikte bestimmt. Die Reihenfolge, in der die Konfliktpartner betrachtet werden, ist beliebig, d.h. die im folgenden entwickelte Funktion Beschriftungsschwierigkeit ist eindeutig, jedoch nicht additiv⁸. Die achsenparallele und damit untereinander parallele Ausrichtung der Beschriftungsräume wird im Folgenden ausgenutzt. Der nicht leere Schnitt zwischen den achsenparallelen Beschriftungsräumen ist wieder ein achsenparalleles Rechteck, das in den beteiligten Beschriftungsräumen liegt und bis auf einen Sonderfall (im nächsten Abschnitt mit Hilfe von Abbildung 39 wird gezeigt, dass das vorgestellte Verfahren diesen berücksichtigt) diese an den Rändern „berührt“ (für topologische Begriffe s. [EGH90]). Diese Konfliktrechtecke bzw. der verbleibende freie Platz werden zu jedem zu beschriftenden Objekt gespeichert. Außerdem wird für die Ermittlung des freien Beschriftungsraums ausgenutzt, dass die Labelbox immer den Rand des Beschriftungsraums berührt, sowie zunächst vereinfachend angenommen den Mittelpunkt des zu beschriftenden punkthaften Objekts. Am Ende des Abschnitts wird eine Verallgemeinerung beschrieben, die die Ausdehnung der Signatur und den Abstand zwischen Signatur und Schriftobjekt (Labelbox) berücksichtigt.

4.3.3.3.1 Einfacher Fall

Das hier neuentwickelte Verfahren für die Bestimmung der Beschriftungsschwierigkeit zu einem Maßstab wird, wie in Abschnitt 4.3.11 beschrieben, in die Generierung des reaktiven Konfliktgraphen, auch in Kombination mit dem Kriterium maximale Anzahl der Konflikte (4.3.3.1), integriert. Für die explizite Parametrisierung mit dem Maßstab müssen die verwendeten Größenangaben der Labelbox mit *labelWidth* und *labelHeight*, des Beschriftungsraums mit *boundingBoxWidth* und *boundingBoxHeight*, der Signatur mit *symbolHeight* und *SymbolWidth*, etc. durch die mit Maßstab und Referenzen parametrisierten Ausdrücke, die in 4.3.1.1 entwickelt wurden, ersetzt werden, worauf aus Gründen der übersichtlicheren Darstellung verzichtet wurde.

Konkret sieht das abstrakt beschriebene Verfahren wie folgt aus: Der Beschriftungsraum des zu beschriftenden Objekts wird in vier Quadranten aufgeteilt, die sich in dem Mittelpunkt des Objekts treffen, wie in Abbildung 34 a) dargestellt.

⁸ Die Funktion Beschriftungsschwierigkeit hat zwei Parameter ($A, \{\text{Konfliktpartner}\}$), das Objekt A , von dem die Beschriftungsschwierigkeit bestimmt werden soll und die Menge der Konfliktpartner. Beschriftungsschwierigkeit ist nicht additiv, da $\text{Beschriftungsschwierigkeit}(A, \{B\}) + \text{Beschriftungsschwierigkeit}(A, \{C\})$ nicht notwendig gleich $\text{Beschriftungsschwierigkeit}(A, \{B, C\})$ ist.

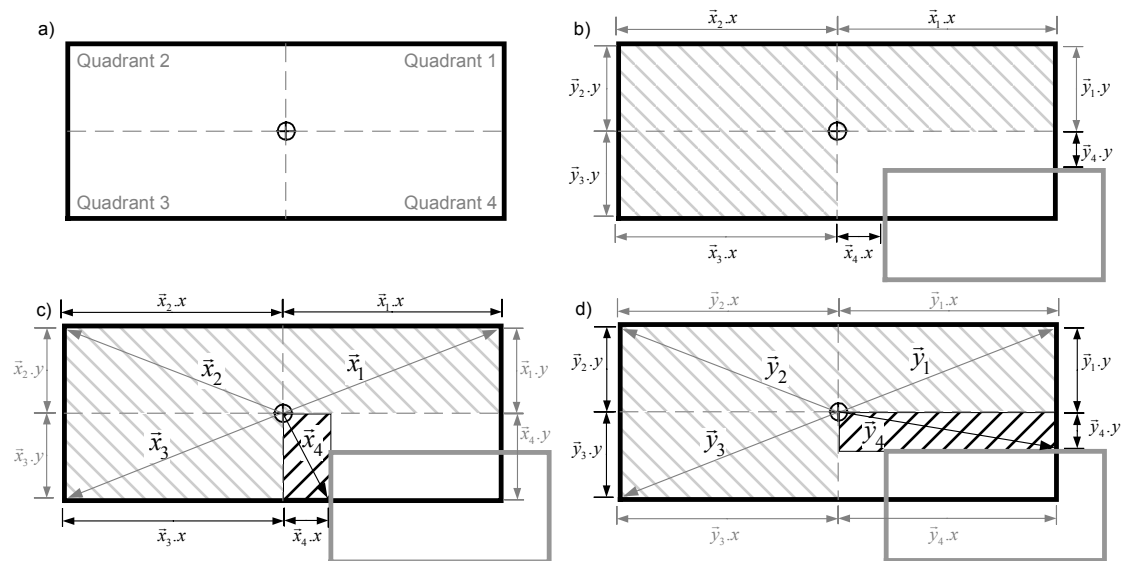


Abbildung 34 Bestimmung der größten freien Beschriftungsfläche. Die Aufteilung des Beschriftungsraums in vier Quadranten ist in a) dargestellt. Für jeden Quadranten⁹ wird ein \bar{x} und ein \bar{y} -Vektor bestimmt, deren Ursprung im Mittelpunkt des zu beschriftenden Objekts liegt c) und d) bzw. die ausschlaggebenden Koordinaten der Vektoren b).

Definition (Quadrant – punkthafte Objekte):

Der Beschriftungsraum eines punkthaften Objekts ist, unter der Annahme, dass die Signatur keine Ausdehnung hat und es keinen Abstand zwischen Signatur und Beschriftungsposition gibt, in vier Quadranten zerlegbar. Eine Ecke jedes Quadranten liegt im Mittelpunkt des Beschriftungsraum und die diagonale Ecke auf einer Ecke des Beschriftungsraums. Die Breite jedes Quadranten entspricht der Breite der zu platzierenden Labelbox nach Definition des Beschriftungsraums und die Höhe des Quadranten der Höhe der Labelbox.

Für jeden Quadranten⁹ wird ein \bar{x} und ein \bar{y} -Vektor¹⁰ angegeben, deren Ursprünge im Mittelpunkt des zu beschriftenden Objekts liegen. Die Namensgebung des Vektors \bar{x} und \bar{y} ist auf den ersten Eindruck verwirrend, aber sinnvoll, da die zugehörige x - bzw. y -Koordinate die wichtigen Informationen trägt, die komplementäre Koordinate ist konstant für das zu beschriftende Objekt.

Existiert kein Konflikt, so beschreiben die durch die Vektoren aufgespannten Rechtecke ihren Quadranten (für die konfliktfreien Quadranten eins bis drei sind die \bar{x} -Vektoren in Abbildung 34 c) dargestellt – in Abbildung 34 d) die zugehörigen konfliktfreien \bar{y} -Vektoren).

Im Fall eines Konflikts werden die betroffenen Quadranten ermittelt (in Abbildung 34 b) der vierte Quadrant). Die x -Koordinate des \bar{x} -Vektors wird dabei auf die kleinste horizontale Distanz zwischen dem Mittelpunkt des zu beschriftenden Objekts und dem Konfliktrechteck gesetzt (Abbildung 34 b)). Anlog wird in der Vertikalen für den \bar{y} -Vektor verfahren. Die y -Koordinate des \bar{x} -Vektors ist konstant für das zu beschriftende Objekt die halbe Höhe des

⁹ Zur Unterscheidung der \bar{x} und \bar{y} -Vektoren und deren x - und y -Koordinate zwischen verschiedenen Quadranten erhalten diese als Index ihre Quadrantennummer (z.B. für den \bar{x} -Vektor des ersten Quadranten $\bar{x}_1.x$, $\bar{x}_1.y$)

¹⁰ Ist einer der Koordinaten des Vektors Null ($\bar{a}.x=0$ oder $\bar{a}.y=0$), dann ist die Fläche, die der Vektor aufspannt, leer und wird mit $\bar{a} = \emptyset$ bezeichnet. Dies darf nicht mit dem Nullvektor verwechselt werden.

Beschriftungsraums, bzw. die Höhe der Labelbox und die x -Koordinate des \bar{y} -Vektors konstant die halbe Breite des Beschriftungsraums bzw. die Breite der Labelbox (Abbildung 34 c) für die Fläche, die der \bar{x} -Vektor beschreibt und d) für die Fläche des \bar{y} -Vektor).

Die anschauliche Beschreibung lässt sich wie folgt formalisieren unter Berücksichtigung der Annahme, dass die Signatur keine Ausdehnung hat ($boundingBoxWidth = 2 * labelWidth$ und $boundingBoxHeight = 2 * labelHeight$). Der Mittelpunkt des zu beschriftenden Objekts sei $(objectX, objectY)$ (vgl. 4.3.1.1). Für jeden Quadranten $Quadrant_i$ wird zu jedem Konfliktpartner $Konfliktpartner_j$, wenn vorhanden, das Konfliktrechteck $Konfliktrechteck_{ij}$ berechnet (da ein Konfliktpartner mit mehreren Quadranten einen Konflikt verursachen kann, können mehrere Konfliktrechtecke – maximal vier, für jeden Quadranten einen – pro zu beschriftendes Objekt entstehen).

Definition (Konfliktrechteck):

Das Konfliktrechteck ist die Schnittfläche der achsenparallelen Rechtecke des Quadranten und der des Beschriftungsraums des Konfliktpartners:

$$Konfliktrechteck_{ij} = Quadrant_i \cap Konfliktpartner_j \quad (4.16)$$

$0 \leq j \leq \text{Anzahl der Konflikte}, i \square \text{ betrachteter Quadrant mit } 1 \leq i \leq 4$

Jedes Konfliktrechteck ist achsenparallel und kann durch seine vier Eckpunkte, die Koordinaten $kp_{ij1}, kp_{ij2}, kp_{ij3}, kp_{ij4}$ (mit $i = \text{betrachteter Quadrant}$ und $j = \text{betrachteter Konflikt}$), beschrieben werden.

Definition (\bar{x} - und \bar{y} -Vektor):

Der Ursprung des \bar{x} - und des \bar{y} -Vektors eines Quadranten $Quadrant_i$ ($1 \leq i \leq 4$) ist in der Ecke des Quadranten, die dem zu beschriftenden Objekt am nächsten ist. Die Koordinaten sind wie folgt definiert:

$$\bar{x} = \begin{cases} \bar{x}_i.x = \text{vorzeichenH} * \text{labelWidth} \\ \bar{x}_i.y = \text{vorzeichenV} * \text{labelHeight} \end{cases} \left. \vphantom{\bar{x}} \right\} \text{Anzahl der Konflikte in Quadrant}_i = 0$$

$$\bar{x} = \begin{cases} \bar{x}_i.x = \text{vorzeichenH} * \min(|\text{objectX} - kp_{ijk}.x|) \\ \bar{x}_i.y = \text{vorzeichenV} * \text{labelHeight} \end{cases} \left. \vphantom{\bar{x}} \right\} \text{sonst, } \forall j, k : j \leq \text{Anzahl der Konflikte in Quadrant}_i \wedge 1 \leq k \leq 4$$

(4.17)

$\text{vorzeichenH} = 1, \text{ wenn } i \in \{1, 4\}$ und $\text{vorzeichenH} = -1, \text{ wenn } i \in \{2, 3\}$
 $\text{vorzeichenV} = 1, \text{ wenn } i \in \{1, 2\}$ und $\text{vorzeichenV} = -1, \text{ wenn } i \in \{3, 4\}$

für den \bar{y} -Vektor folgt analog:

$$\bar{y} = \begin{cases} \bar{y}_i.x = \text{vorzeichenH} * \text{labelWidth} \\ \bar{y}_i.y = \text{vorzeichenV} * \text{labelHeight} \end{cases} \left. \vphantom{\bar{y}} \right\} \text{Anzahl der Konflikte in Quadrant}_i = 0$$

$$\bar{y} = \begin{cases} \bar{y}_i.x = \text{vorzeichenH} * \text{labelWidth} \\ \bar{y}_i.y = \text{vorzeichenV} * \min(|\text{objectY} - kp_{ijk}.y|) \end{cases} \left. \vphantom{\bar{y}} \right\} \text{sonst, } \forall j, k : j \leq \text{Anzahl der Konflikte in Quadrant}_i \wedge 1 \leq k \leq 4$$

(4.18)

$\text{vorzeichenH} = 1, \text{ wenn } i \in \{1, 4\}$ und $\text{vorzeichenH} = -1, \text{ wenn } i \in \{2, 3\}$
 $\text{vorzeichenV} = 1, \text{ wenn } i \in \{1, 2\}$ und $\text{vorzeichenV} = -1, \text{ wenn } i \in \{3, 4\}$

Vor der Demonstration des Verfahrens anhand eines Beispiels mit mehreren Konflikten wird die Ermittlung von konfliktfreien Beschriftungspositionen behandelt.

Die zu platzierende Schrift (Labelbox) kann sich bei genauerer Untersuchung nicht frei in x - und y -Richtung bewegen. Zum einen muss die Labelbox vollständig innerhalb des Beschriftungsraums¹¹ liegen und darf das zu beschriftende Objekt nicht verdecken. Anschaulich bedeutet das, dass die Labelbox an der Innenseite der Grenze des Beschriftungsraums „entlanggleitet“.

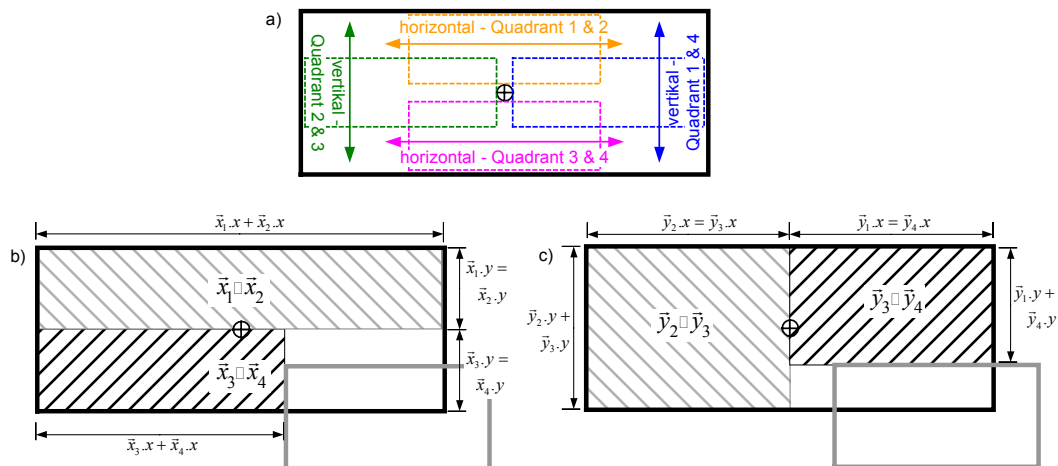


Abbildung 35 Ermittlung der Existenz konfliktfreier Beschriftungspositionen (Fortführung des Beispiels aus Abbildung 34). Vier Bewegungsräume sind für die Labelbox möglich a). Diese vier (überlappenden) Räume entsprechen den zusammen betrachteten Vektoren $\bar{x}_1 \square \bar{x}_2$ und $\bar{x}_3 \square \bar{x}_4$ b) und $\bar{y}_1 \square \bar{y}_4$ und $\bar{y}_2 \square \bar{y}_3$ c).

Definition (Bewegungsraum – punkthafte Objekte):

Der Bewegungsraum eines punkthafte zu beschriftenden Objekts ist der Bereich der zulässigen Beschriftungspositionen ohne Berücksichtigung von Konflikten, in dem die Labelbox innerhalb des Beschriftungsraums an seiner Innenseite entlanggleiten kann. Es gibt zwei horizontale unter der oberen Grenze bzw. über der unteren Grenze des Beschriftungsraums und zwei vertikale links der rechten bzw. rechts der linken Grenze des Beschriftungsraums. Für den einfachen Fall s. Abbildung 35 a) gilt:

- horizontal:
 - Quadrant 1 und Quadrant 2
 - Quadrant 3 und Quadrant 4
- vertikal:
 - Quadrant 1 und Quadrant 4
 - Quadrant 2 und Quadrant 3

Zur Feststellung, ob eine konfliktfreie Beschriftungsposition existiert, werden die \bar{x} und \bar{y} -Vektoren der Quadranten genutzt. Die Vektoren spannen, ausgehend vom Mittelpunkt des zu beschriftenden Objekts, Flächen auf. Über eine vertikale Kante benachbarte und durch \bar{x} -Vektoren aufgespannte Flächen werden zusammen betrachtet (\bar{x}_1 und \bar{x}_2 sowie \bar{x}_3 und \bar{x}_4), genauso wie die durch eine horizontale Kante benachbarten \bar{y} -Vektoren aufgespannten Flächen (\bar{y}_1 und \bar{y}_4 sowie \bar{y}_2 und \bar{y}_3 – Abbildung 35).

¹¹ Definition Beschriftungsraum s. 4.3.1.1

Definition ($\bar{x}_1 \square \bar{x}_2$ und $\bar{y}_1 \square \bar{y}_2$):

Vereinigung der Flächen von entweder zwei \bar{x} - oder zwei \bar{y} - Vektoren mit gleichem Ursprung. Die Höhe der $\bar{x}_1 \square \bar{x}_2$ -Fläche ist gleich der Höhe der Labelbox (*labelHeight*), nach Definition des \bar{x} -Vektors, und die Breite gleich der Summe der x-Koordinaten der beiden Vektoren. Die Breite der $\bar{y}_1 \square \bar{y}_4$ -Fläche ist gleich der Breite der Labelbox (*labelWidth*), nach Definition des \bar{y} -Vektors, und die Höhe ist gleich der Summe der y-Koordinaten der beiden Vektoren. Ist die resultierende Fläche leer, dann wird das durch das Symbol „ \emptyset “ repräsentiert.

Aus Gründen der Übersichtlichkeit wird im Folgenden von den Flächen $\bar{x}_1 \square \bar{x}_2$, $\bar{x}_3 \square \bar{x}_4$, $\bar{y}_1 \square \bar{y}_4$ und $\bar{y}_2 \square \bar{y}_3$ gesprochen, da die Flächen ihre Lage im Raum und ihren Bezugspunkt, den Mittelpunkt des zu beschriftenden Objekts, beibehalten.

Hat mindestens eine dieser vier Flächen mindestens die gleiche Größe wie die zu platzierende Labelbox, existiert eine konfliktfreie Beschriftungsposition in eben dieser Fläche.

Formal lässt sich der Test auf konfliktfreie Beschriftungsposition(en) wie folgt als boolescher Ausdruck formulieren:

$$\begin{aligned}
 & (\text{labelWidth} \leq (\bar{x}_1.x + \bar{x}_2.x)) \vee \\
 & (\text{labelWidth} \leq (\bar{x}_3.x + \bar{x}_4.x)) \vee \\
 & (\text{labelHeight} \leq (\bar{y}_1.y + \bar{y}_4.y)) \vee \\
 & (\text{labelHeight} \leq (\bar{y}_2.y + \bar{y}_3.y))
 \end{aligned} \tag{4.19}$$

Jede Zeile steht für einen der zusammen betrachteten Vektorräume. In dem Ausdruck wird ausgenutzt, dass die Höhe der Labelbox immer in die Höhe der \bar{x} -Vektorräume passt bzw. die Breite der Labelbox immer in die Breite der \bar{y} -Vektorräume bedingt durch deren Definition. Dadurch wird das zweidimensionale Problem auf eine Dimension mit Disjunktion reduziert. D.h. die \bar{x} -Vektorräume müssen auf die Länge der Labelbox getestet werden und die \bar{y} -Vektorräume auf die Höhe der Labelbox. Daher wurde die Bezeichnung \bar{x} -Vektor bzw. \bar{y} -Vektor gewählt.

Abbildung 36 zeigt den Fall, dass mehr als ein Konflikt eintritt. In Abbildung 36 a) sind die für das Verfahren wichtigen Koordinaten der Vektoren $\bar{x}_1.x$, $\bar{x}_2.x$, $\bar{x}_3.x$, $\bar{x}_4.x$, $\bar{y}_1.y$, $\bar{y}_2.y$, $\bar{y}_3.y$ und $\bar{y}_4.y$ (Gleichung (4.17) und (4.18)) eingetragen und in Abbildung 36 b) die in Gleichung (4.19) summierten Koordinaten. Die zusammen betrachteten von den Vektoren aufgespannten Flächen $\bar{x}_1 \square \bar{x}_2$, $\bar{x}_3 \square \bar{x}_4$, $\bar{y}_1 \square \bar{y}_4$ und $\bar{y}_2 \square \bar{y}_3$ sind in Abbildung 36 c) bzw. d) dargestellt. Da $\bar{x}_1.x = \bar{x}_2.x = 0$ gilt, ist die zugehörige zusammen betrachtete Fläche $\bar{x}_1 \square \bar{x}_2$ leer¹⁰. In e) und f) von Abbildung 36 sind die verbleibenden zusammen betrachteten Vektorräume dargestellt, die eine Zeile des booleschen Ausdrucks von Gleichung (4.19) erfüllen, d.h. mindestens die Größe der zu platzierenden Labelbox besitzen, also eine Lösung und damit eine konfliktfreie Beschriftungsposition bieten. Die verbleibende horizontale Bewegungsfreiheit für die Labelbox in der Fläche $\bar{x}_3 \square \bar{x}_4$ und die Vertikale in $\bar{y}_1 \square \bar{y}_4$ ist in Abbildung 36 e) bzw. f) erkennbar. Die Fläche $\bar{y}_2 \square \bar{y}_3$ fällt weg, da $\text{labelHeight} \leq (\bar{y}_2.y + \bar{y}_3.y)$ nicht erfüllt ist.

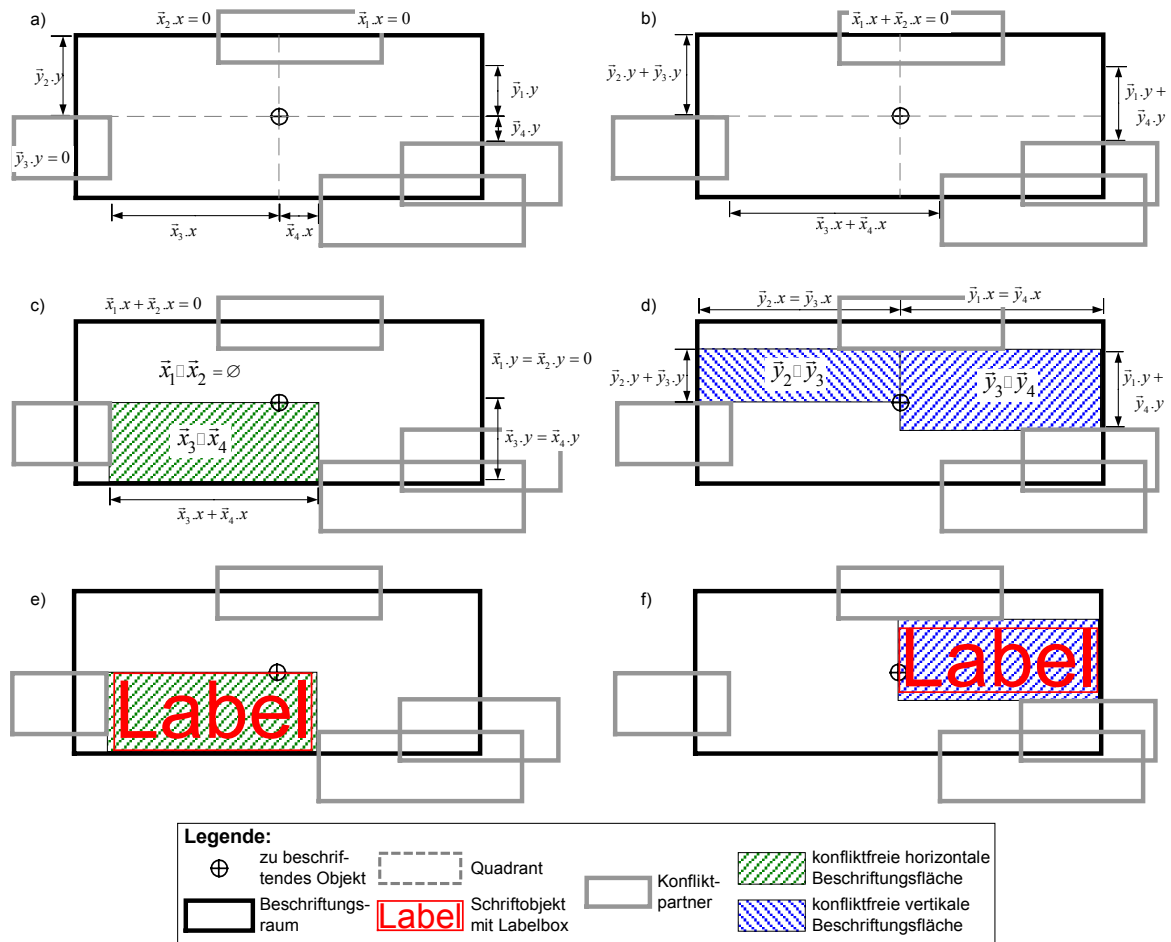


Abbildung 36 Ermittlung konfliktfreier Beschriftungspositionen für ein zu beschriftendes punkthafte Objekt mit mehreren Konflikten. Die linken Abbildungen a), c) und e) beschreiben die horizontalen und die rechten Abbildungen b), d) und f) die vertikalen Bewegungsräume.

Im Folgenden werden zwei besondere Konfliktsituationen besprochen, die zeigen, dass das vorgestellte Verfahren auch diese berücksichtigt. Sie sind in Abbildung 37 dargestellt. In Abbildung 37 a) bis c) liegt die Konfliktfläche und damit der Beschriftungsraum des Konfliktpartners vollständig innerhalb eines Quadranten. Die Überlappung des Mittelpunktes des zu beschriftenden Objekts durch eine Konfliktfläche ist in Abbildung 37 d) bis f) dargestellt. In diesem Fall existiert keine konfliktfreie Beschriftungsposition.

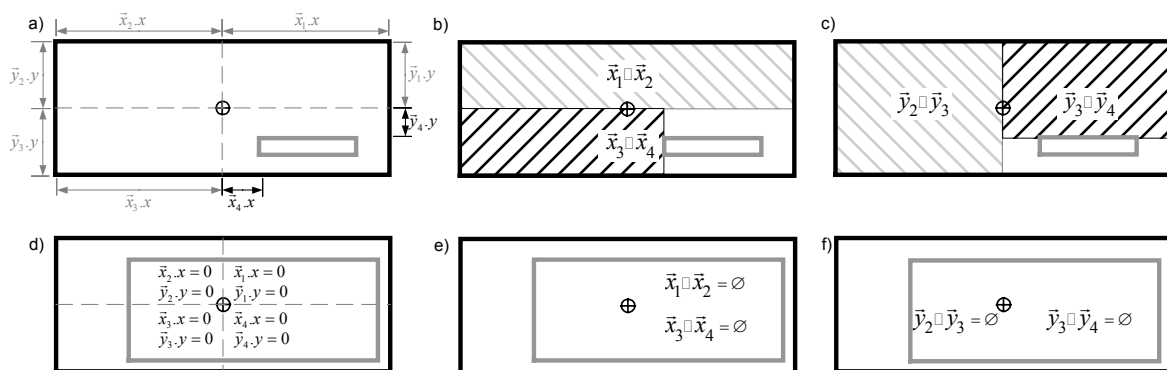


Abbildung 37 Der Konflikt liegt innerhalb des Beschriftungsraums. Abbildungen a) bis c) zeigen den Fall, dass ein Konflikt innerhalb eines Quadranten liegt, während in d) bis f) der Konflikt den Mittelpunkt des zu beschriftenden Objekts überlappt.

Um das dargestellte Verfahren als Deselektionskriterium verwenden zu können, wird eine Bewertungsfunktion benötigt. Ziel ist die Bewertung einer Konfliktsituation für ein zu beschriftendes Objekt zu einem Maßstab. Als Ergebnis ist nicht nur eine boolesche Aussage gesucht: es gibt eine konfliktfreie oder es gibt keine konfliktfreie Beschriftungsposition. Vielmehr ist eine differenzierte (qualitative) Bewertung erforderlich, die eine Aussage darüber macht, wie schwierig es ist, eine konfliktfreie Beschriftung für das Objekt zu finden bzw. wie hart der Konflikt ist. Zusätzlich muss diese Bewertung einen „objektiven“ Vergleich von Konfliktsituation verschiedener zu beschriftender Objekte ermöglichen.

Für die Bewertung wird eine Kombination aus konfliktfreien Beschriftungspositionen und konfliktfreien Flächen im Beschriftungsraum genutzt, die auf dem booleschen Ausdruck zum Auffinden konfliktfreier Beschriftungspositionen (4.19) beruht. Dazu werden die Terme der Gleichung (4.19) einzeln ausgewertet und normiert, d.h. der Ausdruck für die Summe der x-Werte wird durch die Länge der Labelbox (*labelWidth*) geteilt, die der y-Werte durch die Höhe der Labelbox (*labelHeight*).

Definition (Bewertung der Bewegungsräume – *value* – einfacher Fall):

Für jeden Bewegungsraum wird in Abhängigkeit der Bewegungsrichtung, seiner Lage und seinen Konflikten der Wert *value* wie folgt für die horizontalen Bewegungsräume definiert:

$$value_1 = \frac{x_1 \cdot x + x_2 \cdot x}{labelWidth} , \quad 0 \leq value_1 \leq 2 \quad (4.20)$$

$$value_2 = \frac{x_3 \cdot x + x_4 \cdot x}{labelWidth} , \quad 0 \leq value_2 \leq 2 \quad (4.21)$$

und für die vertikalen:

$$value_3 = \frac{y_1 \cdot y + y_4 \cdot y}{labelHeight} , \quad 0 \leq value_3 \leq 2 \quad (4.22)$$

$$value_4 = \frac{y_2 \cdot y + y_3 \cdot y}{labelHeight} , \quad 0 \leq value_4 \leq 2 \quad (4.23)$$

Value ist ein normierter Wert für die Bewertung und Vergleich der verbliebenen Bewegungsräume nach Berücksichtigung von Konflikten unabhängig von der Bewegungsrichtung und Lage der Bewegungsräume.

Der in Gleichung (4.19) aufgestellte Test auf konfliktfreie Beschriftungsposition lässt sich somit wie folgt ausdrücken:

$$\max(value_1, value_2, value_3, value_4) \geq 1 \quad \Leftrightarrow \quad Gleichung(4.22) = wahr \quad (4.24)$$

Dies führt zu zwei Bewertungsmöglichkeiten:

$$labelScore_{max} = \max(value_1, value_2, value_3, value_4) , \quad 0 \leq labelScore_{max} \leq 2 \quad (4.25)$$

$$labelScore_{sum} = \sum_{i=1}^4 value_i , \quad 0 \leq labelScore_{sum} \leq 8 \quad (4.26)$$

In $labelScore_{max}$ wird nur die konfliktärmste der vier möglichen vereinigten Vektorflächen berücksichtigt, während in $labelScore_{sum}$ alle berücksichtigt werden, unter Inkaufnahme, dass Beschriftungspositionen doppelt gezählt werden (horizontal und vertikal). Abgesehen von

dieser Besonderheit unterscheiden sich die beiden Bewertungen auf den ersten Blick nicht. Jedoch ist die Folgerung, ob eine konfliktfreie Beschriftungsposition existiert, bei beiden Bewertungen unterschiedlich. Für die erste Bewertung (Gleichung (4.25)) existiert genau dann mindestens eine konfliktfreie Beschriftungsposition, wenn $labelScore_{max} \geq 1$ (wegen der Normalisierung) ist. Ist $labelScore_{max} < 1$ existiert keine konfliktfreie Beschriftungsposition (Abbildung 38).

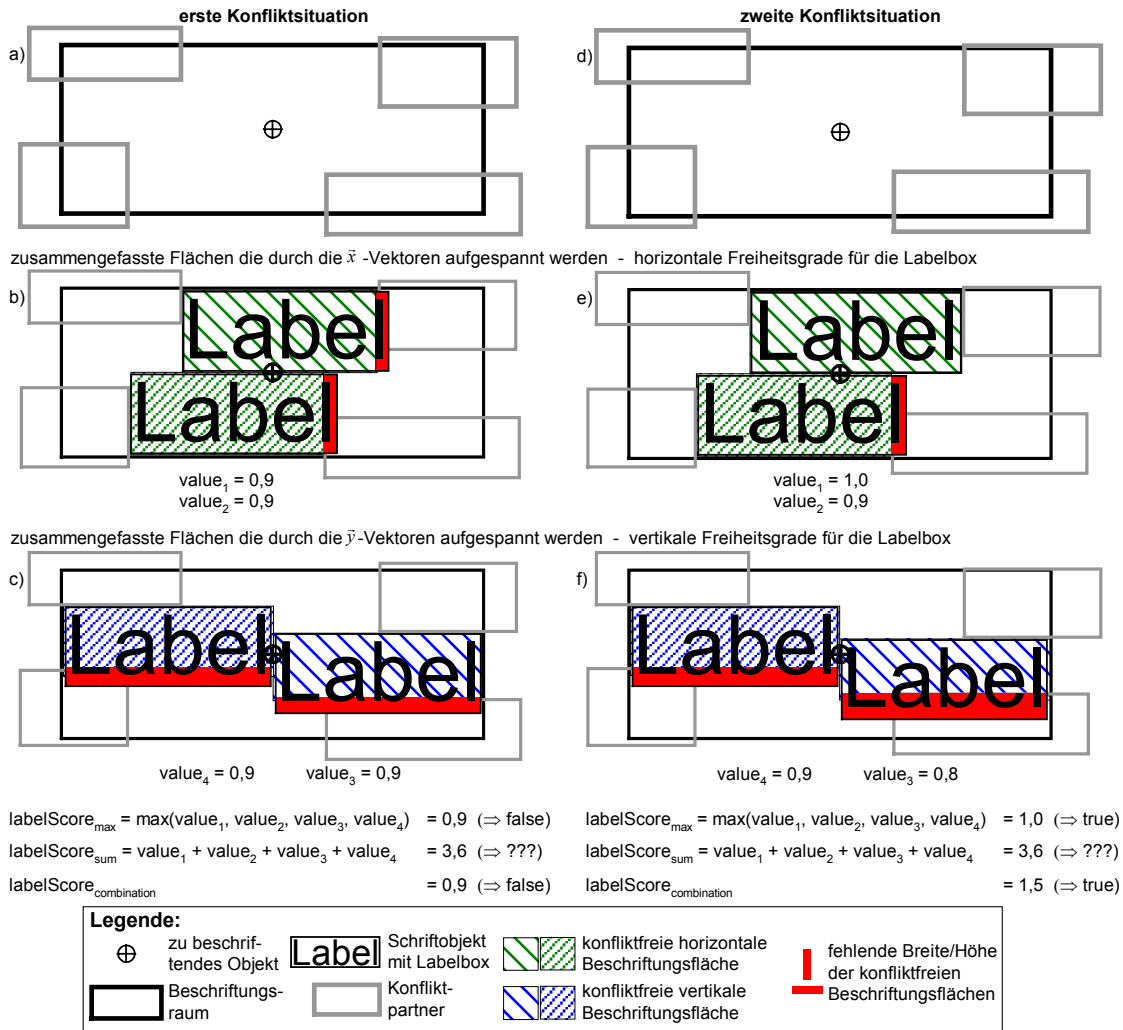


Abbildung 38 Bewertungen von Konfliktsituationen mit den entwickelten Beschriftungskriterien am Beispiel der in a) und b) dargestellten Konfliktsituationen. Die Bewertungen mit den drei entwickelten Kriterien sind unten dargestellt. Hinter den Zahlenwerten ist angegeben, ob eine konfliktfreie Beschriftungsposition existiert („true“), nicht existiert („false“) bzw. ob das Kriterium (für den Zahlenwert) keinen Schluss darauf zulässt („???“).

Für die sichere Existenz mindestens einer konfliktfreien Beschriftungsposition mit der Bewertung $labelScore_{sum}$ muss $labelScore_{sum} \geq 4$ sein. Drei der $value$ -Werte können unter Eins liegen, es existiert keine konfliktfreie Beschriftungsposition und der vierte $value$ -Werte den Wert Eins haben – es existiert eine konfliktfreie Beschriftungsposition. Es kann jedoch bei einem Wert von $1 \leq labelScore_{sum} < 4$ eine konfliktfreie Beschriftungsposition vorliegen – es gibt aber keine sichere Aussage (einer der $value$ -Werte könnte den Wert Eins haben und damit eine konfliktfreie Beschriftungsposition ausdrücken und die restlichen $value$ -Werte den Wert Null besitzen) (Abbildung 38).

Daher ist für die Bestimmung der Beschriftungsschwierigkeit das Bewertungskriterium $labelScore_{max}$ vorzuziehen, dessen Wert $labelScore_{max} < 1$ angibt, dass (sicher) keine konfliktfreie Beschriftungsposition existiert, und $labelScore_{max} \geq 1$, dass (sicher) eine konfliktfreie Beschriftungsposition existiert. Durch die Normierung ist ein objektiver Vergleich zwischen den Beschriftungsschwierigkeiten verschiedener zu beschriftender Objekte möglich. Das in Abschnitt 4.3.3.2 vorgestellte Kriterium, des größten freien Winkels, lässt neben den dargestellten Nachteilen diesen Vergleich nicht zu, da neben der Größe des größten freien Winkels eine Abhängigkeit von seiner Lage entscheidend ist.

Alternativ kann ein Bewertungskriterium aus einer Kombination von $labelScore_{sum}$ und $labelScore_{max}$ abgeleitet werden, so dass aus dem Kriterium $labelScore_{max}$ die Eigenschaft entnommen wird < 1 es existiert (sicher) keine konfliktfreie Beschriftungsposition und bei einem Wert ≥ 1 es existiert (sicher) mindestens eine konfliktfreie Beschriftungsposition. Eine solche Bewertung kann wie folgt aussehen (Abbildung 38):

$$labelScore_{combination} = \begin{cases} \frac{labelScore_{sum}}{4}, & \text{wenn } labelScore_{max} < 1 \\ 1 + \frac{labelScore_{sum}}{8}, & \text{wenn } labelScore_{max} \geq 1 \end{cases}, \quad 0 \leq labelScore_{combination} \leq 2 \quad (4.27)$$

```

1  Input: Ein zu beschriftendes punkthafes Objekt mit Beschriftungsraum und Label-
      box sowie Konfliktpartner und deren Beschriftungsräume zum vorgegebenen
      Maßstab, zu dem die Beschriftungsschwierigkeit bestimmt werden soll.
      Output: Bewertung der Beschriftungsschwierigkeit des betrachteten punkthafes
5     Objekts zu vorgegebenem Maßstab mit dem Kriterium freie
      Beschriftungsfläche und den drei entwickelten Bewertungsfunktionen:
       $labelScore_{max}$ ,  $labelScore_{sum}$  und  $labelScore_{combination}$  (vom Typ Float)
      // mittelpunkt = Mittelpunkt des zu beschriftenden Objekts
      // konfliktfläche = Schnitt der Beschriftungsflächen der Konfliktpartner
10    // Initial für das betrachtete zu beschriftende Objekt zu einem Maßstab:
       $x_{1..4} = boundingBoxWidth / 2 = labelWidth$ 
       $y_{1..4} = boundingBoxHeight / 2 = labelHeight$ 
      für jeden Konfliktpartner
          bestimme konfliktfläche
15     bestimme Konfliktquadrant bzw. Konfliktquadranten
          für jeden beteiligten Konfliktquadranten  $i$ 
               $x_i = \text{minimum}(x_i, \text{horizontaleDistanz}(\text{mittelpunkt}, \text{konfliktfläche}))$ 
               $y_i = \text{minimum}(y_i, \text{vertikaleDistanz}(\text{mittelpunkt}, \text{konfliktfläche}))$ 
               $value_1 = (x_1 + x_2) / labelWidth$ 
20      $value_2 = (x_3 + x_4) / labelWidth$ 
               $value_3 = (y_1 + y_4) / labelHeight$ 
               $value_4 = (y_2 + y_3) / labelHeight$ 
               $labelScore_{max} = \text{maximum}(value_1, value_2, value_3, value_4)$ 
               $labelScore_{sum} = value_1 + value_2 + value_3 + value_4$ 
25     wenn  $labelScore_{max} < 1$ 
          dann
               $labelScore_{combination} = labelScore_{sum}/4$ 
          sonst
               $labelScore_{combination} = 1 + labelScore_{sum}/8$ 

```

Algorithmus 2 Bestimmung der Beschriftungsschwierigkeit für ein zu beschriftendes punkthafes Objekt zu einem Maßstab. Es wird angenommen, dass die Signatur keine Ausdehnung hat und es keinen Abstand zwischen Signatur und zu platzierendem Schriftobjekt gibt.

Aus der skizzierten Bewertung ergibt sich Algorithmus 2. In den Zeilen 11 bis 18 werden die Konfliktrechtecke sowie die \bar{x} und \bar{y} -Vektoren der Quadranten ermittelt. Die entwickelten

Bewertungen $labelScore_{max}$, $labelScore_{sum}$ und $labelScore_{combination}$ sind in den Zeilen 23 bis 29 umgesetzt, basierend auf den normierten $value$ -Werten der Quadranten (Zeilen 19 bis 22). Sie können als Deselektionskriterium genutzt werden und zur Bestimmung ob eine konfliktfreie Beschriftungsposition existiert.

4.3.3.3.2 Verallgemeinerung

Bisher wurde der Fall behandelt, dass die Signatur keine Ausdehnung besitzt und es keinen Abstand zwischen Signatur und Labelbox gibt, d.h. die Breite des Beschriftungsraums die doppelte Breite der Labelbox ist und die Höhe die doppelte Höhe der Labelbox. Die Signatur selbst hat aber eine Ausdehnung. Abbildung 39 zeigt den verallgemeinerten Fall an einem Beispiel. Die Größe der Signatur erscheint auf den ersten Blick unverhältnismäßig groß, dient aber der anschaulichen Erklärung des allgemeinen Falles – die Signatur ist breiter und höher als eine Labelbox. Der Beschriftungsraum ohne Signatur und Abstand zwischen Signatur und Labelbox – der Raum, in dem die Labelbox platziert werden soll – wird in Gebiete aufgeteilt, die maximal die Ausdehnung der Labelbox haben. Die Flächen, die in Abbildung 34 die Bezeichnungen Quadrant 1 bis 4 haben, sind in Gebiet 1 bis 4 umbenannt, da sie keine vollständigen Quadranten mehr beschreiben. Die freien Flächen zwischen diesen ehemaligen Quadranten werden mit Gebieten, die die Größe der Labelbox haben, „aufgefüllt“ (in Abbildung 39 Gebiete 5,1; 6,1; 7,1 und 8,1). Die verbleibenden Lücken werden, wenn vorhanden, mit Gebieten, die kleiner als die Labelbox sind, geschlossen (in Abbildung 39 Gebiete 5,2; 6,2; 7,2 und 8,2).

Definition (Gebiete – punkthafte Objekte):

Der Beschriftungsraum eines punkthafte Objekts unterteilt sich wie folgt in Gebiete, die maximal die Größe der Labelbox besitzen: In den Ecken des Beschriftungsraums liegen die Gebiete 1 bis 4, die die Größe der Labelbox haben. Die verbleibenden horizontalen Flächen zwischen den Gebieten 1 und 2 sowie den Gebieten 3 und 4 und die vertikalen Flächen zwischen den Gebieten 2 und 3 sowie den Gebieten 4 und 1 zwischen Signatur und Beschriftungsraumgrenze werden mit Gebieten, die die Größe der Labelbox haben, aufgefüllt. Verbleibende Lücken werden, wenn vorhanden mit Gebieten, die kleiner als die Labelbox sind, aufgefüllt. Die Gebiete oberhalb der Signatur erhalten gegen den Uhrzeigersinn die Bezeichnung $5,i$, mit i Zähler der Gebiete, links neben der Signatur $6,j$, mit j Zähler der Gebiete, unterhalb der Signatur $7,i$ und rechts neben der Signatur $8,j$ (Abbildung 39).

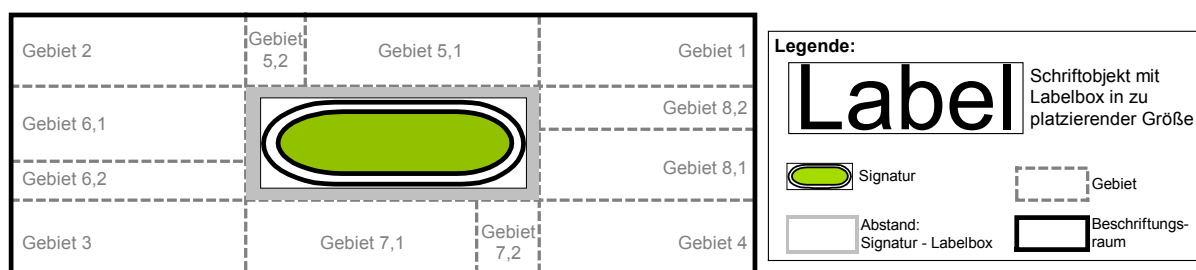


Abbildung 39 Der Beschriftungsraum wird bis auf die Signatur und den Abstand zwischen Signatur und Labelbox – also der Raum, in dem die Schrift platziert werden soll – in Gebiete aufgeteilt. Die Gebiete über und unter der Signatur (fünfer und sechser Gebiete¹²) haben alle die Höhe der Labelbox und die links und rechts der Signatur (sechser und achter Gebiete¹²) alle die Breite der Labelbox.

Im Folgenden wird die im Beispiel beschriebene Aufteilung in Gebiete formalisiert. Die Breite des Raums über bzw. unter der Signatur und dem Abstand links und rechts der

Signatur, also der Raum zwischen Gebiet 1 und 2 bzw. Gebiet 3 und 4, beträgt i -Labelboxen mit $i = \frac{\text{symbolWidth} + 2 * \text{distanceSymbolLabel}}{\text{labelWidth}}$. Die Größe der Gebiete 5,1 bis 5,*abgerundet*(i)

bzw. Gebiete 7,1 bis 7,*abgerundet*(i) entsprechen der Größe der Labelbox. Wenn i eine nicht ganze Zahl ist, dann hat das verbleibende Gebiet 5,*aufgerundet*(i) bzw. Gebiet 7,*aufgerundet*(i) die Breite $\text{labelWidth} * (i - \text{abgerundet}(i))$. Die Höhe der fünfer und siebener Gebiete¹² ist immer fest labelHeight nach Definition des Beschriftungsraums.

Analog wird für den vertikalen Raum zwischen Gebiet 2 und 3 bzw. Gebiet 4 und 1 verfahren. Die Gebiete 6,1 bis 6, j bzw. Gebiete 8,1 bis 8, j , die links und rechts neben der Signatur und dem Abstand liegen, verfahren. Die Breite dieser Gebiete beträgt immer labelWidth . Der vertikale zu füllende Abstand beträgt j -Labelboxen mit $j = \frac{\text{symbolHeight} + 2 * \text{distanceSymbolLabel}}{\text{labelHeight}}$. Die Größe der Gebiete 6,1 bis 6,*abgerundet*(j) bzw.

Gebiete 8,1 bis 8,*abgerundet*(j) entsprechen der Größe der Labelbox. Das verbleibende Gebiet 6, *aufgerundet*(j) bzw. Gebiet 8,*aufgerundet*(j), wenn j eine nicht ganze Zahl ist, hat die Höhe $\text{labelHeight} * (j - \text{abgerundet}(j))$.

Die Labelbox kann sich, anschaulich gesehen, wie zuvor in vier Bereichen entlang der Innenseite der Grenze des Beschriftungsraums „bewegen“: horizontal über der Signatur (Gebiet 1; 5,1; ... 5, i ; 2), unter der Signatur (Gebiet 3; 7,1; ... 7, i ; 4) sowie vertikal links (Gebiet 2; 6,1; ... 6, i ; 3) und rechts der Signatur (Gebiet 4; 8,1; ... 8, i ; 1).

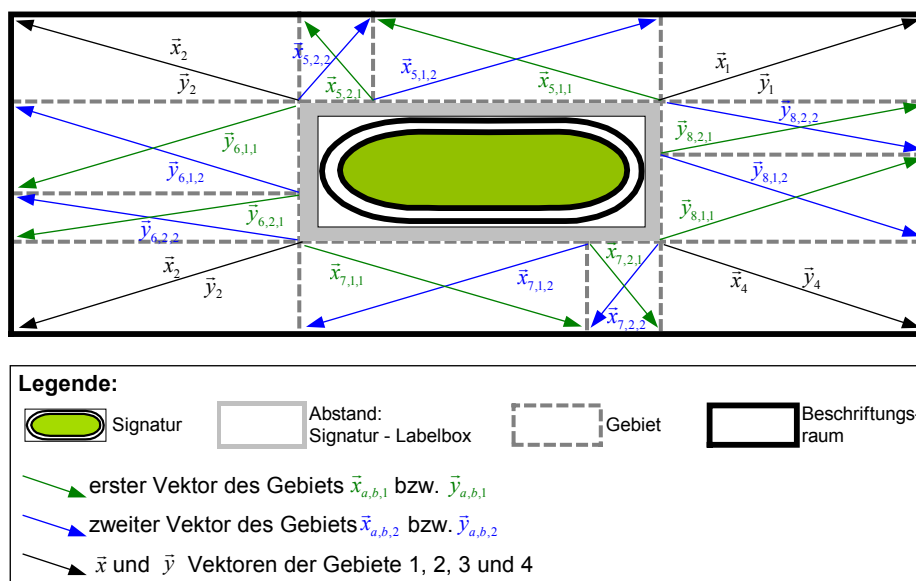


Abbildung 40 Zuordnung von Vektoren zu Gebieten.

Die Gebiete 1 bis 4 erhalten, wie die vier Quadranten zuvor, \bar{x} und \bar{y} -Vektoren, da sie sowohl in den horizontalen als auch vertikalen „Bewegungsrichtungen“ der Labelbox liegen. Der Ursprung beider Vektoren ist in der Gebietsecke, die am nächsten der Signatur ist (Gebiet 1 ist links unten, Gebiet 2 rechts unten, Gebiet 3 rechts oben und Gebiet 4 links oben – Abbildung 40).

¹² Fünfer, sechser, siebener und achter Gebiete bezeichnen die Gebiete 5, s ; 6, t ; 7, s bzw. 8, t mit $0 \leq s \leq \text{abgerundet}(i)$ bzw. $0 \leq t \leq \text{abgerundet}(j)$.

Die verbleibenden Gebiete erhalten jeweils zwei Vektoren, die horizontal angeordneten fünfer und siebener Gebiete jeweils zwei \bar{x} -Vektoren¹³ und die vertikal angeordneten sechser und achter Gebiete jeweils zwei \bar{y} -Vektoren¹³. In den Gebieten $5,i$ ist der Ursprung der Vektoren $\bar{x}_{5,i,1}$ in der unteren rechten Ecke bzw. der von $\bar{x}_{5,i,2}$ in der unteren linken Ecke des Gebiets. Die \bar{x} -Vektoren der Gebiete $7,i$ haben ihren Ursprung in der oberen linken Ecke $\bar{x}_{7,i,1}$ bzw. in der oberen rechten Ecke $\bar{x}_{7,i,2}$. Für die Gebiete $6,j$ ist der Ursprung der \bar{y} -Vektoren $\bar{y}_{6,j,1}$ in der rechten oberen Ecke bzw. der von $\bar{y}_{6,j,2}$ in der rechten unteren Ecken. Analog ist der Ursprung für die Gebiete $8,j$ von $\bar{y}_{8,j,1}$ in der linken unteren Ecke bzw. von $\bar{y}_{8,j,2}$ in der linken oberen Ecke.

An dem Beispiel, das in Abbildung 41 dargestellt ist, soll die Bestimmung der Beschriftungsschwierigkeit und das Auffinden einer konfliktfreien Beschriftungsposition gezeigt werden. Dazu werden die vom Konflikt betroffenen Gebiete und ihre Vektoren bestimmt. Die Vektoren, die keine Fläche aufspannen und daher wegfallen, sind rechts neben der Abbildung aufgelistet.

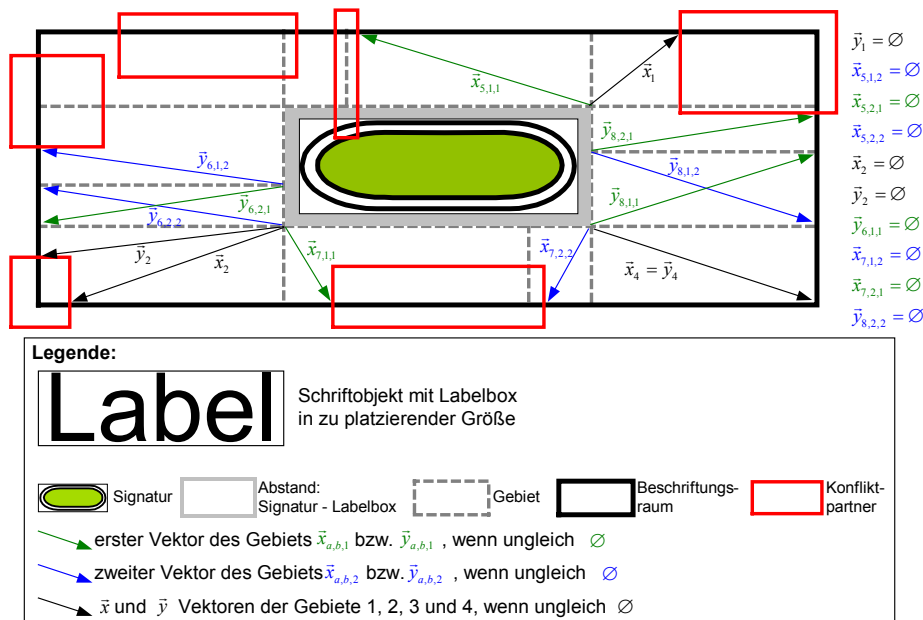


Abbildung 41 Ein Konfliktszenario für ein punkthaftes zu beschriftendes Objekt. Die entstehenden bzw. verbleibenden Vektoren sind in der Abbildung eingetragen, Vektoren, die wegfallen, sind rechts aufgelistet.

In Abbildung 42 sind die Beschriftungsräume für die horizontale Bewegungsrichtung gelb schraffiert dargestellt.

Definition (konfliktfreier Bewegungsraum):

Der konfliktfreie Bewegungsraum eines zu beschriftenden punkthaften Objekts ist der Bereich des Bewegungsraum, der keinen Konflikt besitzt¹⁴.

¹³ Gebiet_{g,i} (g = 5 oder 7) hat die Vektoren $\bar{x}_{g,i,1}$ und $\bar{x}_{g,i,2}$ und Gebiet_{g,i} (g = 6 oder 8) hat die Vektoren $\bar{y}_{g,i,1}$ und $\bar{y}_{g,i,2}$

Die Abbildung macht deutlich, dass mehr als ein konfliktfreier Bewegungsraum¹⁴ pro Bewegungsrichtung und Ebene entstehen kann (die beiden gelb schraffierten Räume unterhalb der Signatur). Es stellt sich die Frage, ob in so einer Fläche die Labelbox konfliktfrei platziert werden kann. Dies kann nicht eintreten, da Breite und Höhe der Gebiete kleiner gleich denen der Labelbox sind. Sobald ein Gebiet an einem Konflikt beteiligt ist, ist die verbleibende Fläche des Gebiets allein zu klein für eine konfliktfreie Platzierung. Hat ein Gebiet, wie in diesem Fall, auf beiden Seiten Konflikte, dann ist die verbleibende Fläche dazwischen folglich immer zu klein, um eine konfliktfreie Labelbox aufzunehmen. In Abbildung 43 sind die konfliktfreien Bewegungsräume für die vertikale Bewegungsrichtung dargestellt. Beide Räume umfassen mehr als zwei Gebiete. Diese beiden Gegebenheiten führen zu einem neuen Verfahren zur Bestimmung der Existenz einer konfliktfreien Beschriftungsposition und der Bewertung von Konfliktsituationen.

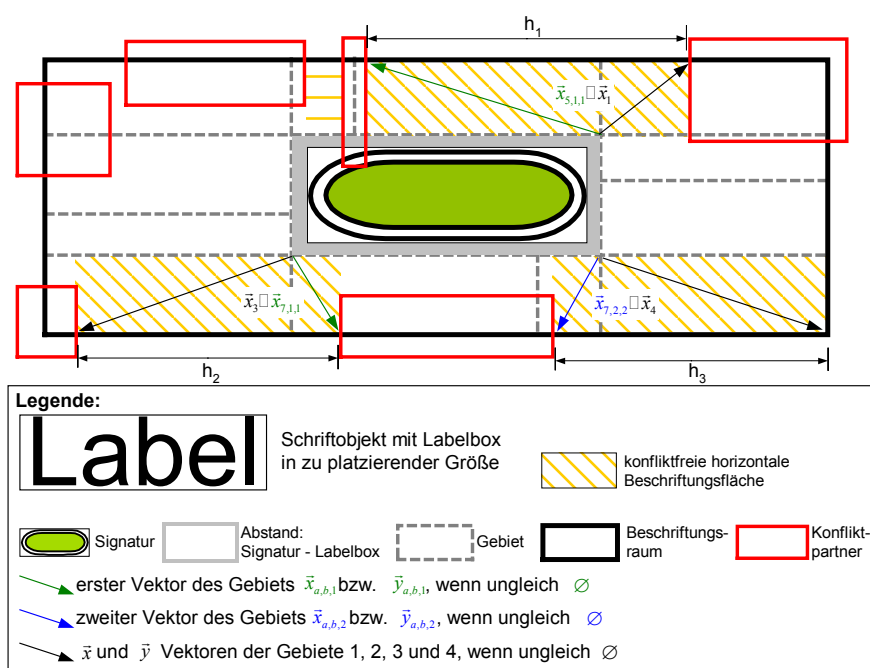


Abbildung 42 Die verbleibenden konfliktfreien horizontalen Bewegungsräume sind gelb schraffiert dargestellt. Die Labelbox kann in ihnen konfliktfrei platziert werden, wenn die Breite dieser Rechtecke (h_1 , h_2 und h_3) größer oder gleich der Breite der Labelbox ist.

Für die beiden horizontalen und die beiden vertikalen Bewegungsrichtungen (vgl. Abbildung 35 a)) müssen die konfliktfreien Bewegungsräume ermittelt werden. Im Gegensatz zu dem im letzten Abschnitt betrachteten Fall, die Signatur hat keine Ausdehnung, können mehrere dieser Räume pro Bewegungsrichtung auftreten (Abbildung 42). Um diese zum Beispiel für die obere horizontale Bewegungsrichtung zu ermitteln, müssen die Gebiete, die an der oberen horizontalen Beschriftungsraumgrenze liegen, von rechts nach links durchlaufen werden (Gebiet 1, Gebiete 5,1 – 5,i bis Gebiet 2) und die konfliktfreien Räume bestimmt werden (Algorithmus 3).

¹⁴ Im Unterschied zu einer konfliktfreien Beschriftungsposition garantiert ein konfliktfreier Bewegungsraum nicht die vollständige konfliktfreie Platzierung der Labelbox in diesem. Der horizontale konfliktfreie Bewegungsraum sichert zu, dass die zu platzierende Labelbox keinen Konflikt bezüglich der Höhe mit dem Raum hat. Die Breite des horizontalen Beschriftungsraums kann kleiner als die der Labelbox sein. Analog sichert der vertikale konfliktfreie Beschriftungsraum die konfliktfreie Platzierung der Labelbox bezüglich der Breite zu. Über die Höhe wird keine Zusicherung getroffen.

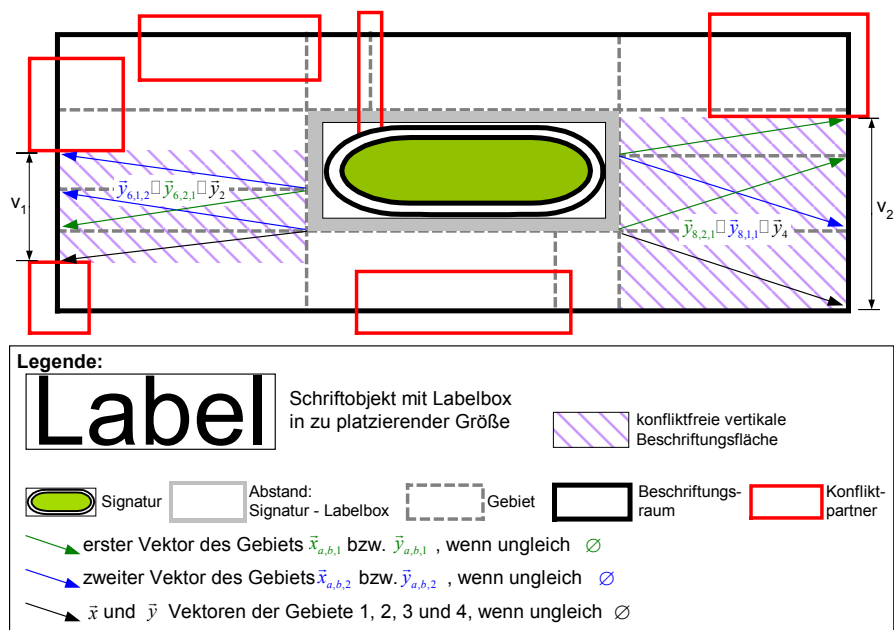


Abbildung 43 Die verbleibenden konfliktfreien vertikalen Bewegungsräume sind violett schraffiert dargestellt. Ist die Höhe dieser Flächen (v_1 und v_2) gleich oder größer der Höhe der Labelbox, kann die Labelbox konfliktfrei in ihnen platziert werden.

An jeder Grenze zwischen zwei dieser Gebiete existieren \bar{x} -Vektoren, die den konfliktfreien Raum nach rechts ($\bar{x}_{s,n-1,2}$) und nach links ($\bar{x}_{s,n,1}$) beschreiben (an der ersten bzw. letzten Grenze zwischen diesen Gebieten \bar{x}_1 und $\bar{x}_{s,1,1}$ bzw. $\bar{x}_{s,i,2}$ und \bar{x}_2). An jeder dieser gemeinsamen Grenzen wird angehalten und der \bar{x} -Vektor, der nach rechts zeigt, betrachtet. Ist die Breite der durch ihn aufgespannten Fläche kleiner als die Breite des zugehörigen (rechten) Gebiets, gibt es einen neuen konfliktfreien Bewegungsraum, der sich aus der zusammen betrachteten Fläche der nach rechts und nach links zeigenden Vektoren ergibt. Ist die Breite der Fläche des durch den nach rechts zeigenden Vektors gleich der Breite des zugehörigen (rechten) Gebiets, dann wird der aktuelle konfliktfreie Bewegungsraum durch die Fläche des nach links zeigenden Vektors erweitert. Die Fläche des nach rechts zeigenden Vektors ist in dem konfliktfreien Bewegungsraum durch den anderen Vektor des Gebiets berücksichtigt. Der konfliktfreie Bewegungsraum spannt sich über mehr als zwei Gebiete (Abbildung 43). Analog werden die konfliktfreien Bewegungsräume für die verbleibende horizontale und die vertikalen Bewegungsrichtungen bestimmt.

Um die horizontalen und vertikalen konfliktfreien Bewegungsräume sowie Konfliktsituationen verschiedener zu beschriftender punkthafter Objekte vergleichen zu können, werden die Breiten (bei den horizontal) bzw. die Höhen (bei den vertikal) ermittelten konfliktfreien Bewegungsräumen normiert, durch Division mit der Breite der Labelbox (*labelWidth*) bzw. der Höhe der Labelbox (*labelHeight*). Dies führt zu $value_{1,m}$, $value_{2,n}$, $value_{3,o}$ und $value_{4,p}$ – die „1“ im Index steht für die konfliktfreien Beschriftungsräume in der oberen horizontalen Bewegungsrichtung, „2“ linke vertikale, „3“ untere horizontale und „4“ rechte vertikale Bewegungsrichtung. Die Indizes m , n , o und p zeigen an, dass es mehr als einen konfliktfreien Beschriftungsraum pro Bewegungsrichtung gibt.

```

1  Input: Ein zu beschriftendes punkthafes Objekt mit x- und y-Vektoren der Gebiete
      des Beschriftungsraum in denen die Konflikte zu Konfliktpartnern bzw.
      deren Beschriftungsräume berücksichtigt sind zum vorgegebenen Maßstab.
Output: Bewertung der konfliktfreien Bewegungsräume, hier begrenzt auf den oberen
5  horizontalen Bewegungsraum, durch die normierten value-Werte zum
      vorgegeben Maßstab (vom Typ Float) sowie die Anzahl der zusammenhängenden
      konfliktfreien Bewegungsräume m1.
      // für die horizontalen Bewegungsräume oben:
wenn i > 0 // i ≙ Anzahl der Gebiete oberhalb der Signatur
10  m1 = 0 // Zähler für konfliktfreie Flächen
      wenn (x1 + x5,1,1) > 0 // konfliktfreie Fläche zwischen Gebiet 1 und 5,1
          m1 = 1
          value1,m1 = x1 + x5,1,1
      für n = 2, n <= i, n++
15  wenn x5,n-1,2 = labelWidth
          // es existiert kein Konflikt in Gebiet 5,n-1. Das Gebiet ist also durch
          // x5,n-1,1 in value1,m1 aufgenommen. Es muss nur noch x5,n,1 hinzugenommen
          // werden.
          value1,m1 = value1,m1 + x5,n,1
20  sonst
          wenn (x5,n-1,2 + x5,n,1) > 0
              // das letzte Gebiet war nicht konfliktfrei und es kommt ein neuer
              // konfliktfreier Raum hinzu. Zuvor vorherigen value normieren.
              value1,m1 = value1,m1/labelWidth
25  m1++
          value1,m1 = (x5,n-1,2 + x5,n,1)
          // sonst es gibt keinen konfliktfreien Raum zwischen Gebiet 5,m1-1 & 5,m1
wenn x5,i,2 = labelWidth
          value1,m1 = value1,m1 + x2
30  sonst
          wenn (x5,i,2 + x2) > 0
              // das letzte Gebiet war nicht konfliktfrei und es kommt ein neuer
              // konfliktfreier Raum hinzu. Zuvor vorherigen value normieren.
              value1,m1 = value1,m1/labelWidth
35  m1++
          value1,m1 = (x5,i,2 + x2)
          // sonst es gibt keinen konfliktfreien Raum zwischen Gebiet 5,i und 2
sonst // die Signatur hat keine Ausdehnung - keine Gebiete oberhalb der Signatur
          m1 = 1
40  value1,m1 = (x1 + x2)/labelWidth

```

Algorithmus 3 Bestimmung der oberen horizontalen Bewegungsräumen bzw. der normierten Bewertungen $value_{1,m}$. Die Ermittlung der Werte, für die untere horizontale sowie die beiden vertikalen Bewegungsrichtungen, verläuft analog. Die Bestimmung der Konfliktflächen und damit der x -Werte (x_1 , x_2 , $x_{5,n,1}$ bzw. $x_{5,n,2}$) erfolgt in Algorithmus 4, der die Bewertungen für die Bewegungsräume (diesen Algorithmus) aufruft. In den Zeilen 9 bis 37 wird der Fall behandelt, dass die Signatur eine horizontale Ausdehnung besitzt und somit mindestens ein fünfer Gebiet¹² zwischen den Gebieten 1 und 2 liegen ($i > 0$) – Zeilen 38 bis 40 den anderen Fall (vgl. Abbildung 39). Zeile 11 bis 13 behandelt den Übergang von Gebiet 1 zu den fünfer Gebieten; Zeile 14 bis 27 die fünfer Gebiete; Zeile 28 bis 37 den Übergang von den fünfer Gebieten zu Gebiet 2. $m1$ liefert die Anzahl der gefundenen zusammenhängenden Bewertungsbereiche und $value_{1,m}$ die zugehörigen normierten Bewertungen.

Definition (Bewertung der konfliktfreien Bewegungsräume – value – verallgemeinerter Fall):

Für jeden Bewegungsraum werden die konfliktfreien Bewegungsräume normiert durch $value_{1,m}$ für den oberen horizontalen, $value_{2,n}$ für den linken vertikalen, $value_{3,o}$ für den unteren horizontalen und $value_{4,p}$ für den rechten vertikalen Bewegungsraum wie folgt bewertet:

$$0 \leq value_{1,m} \leq boundingBoxWidth / labelWidth \quad (4.28)$$

$$0 \leq value_{2,n} \leq boundingBoxHeight / labelHeight \quad (4.29)$$

$$0 \leq value_{3,o} \leq boundingBoxWidth / labelWidth \quad (4.30)$$

$$0 \leq value_{4,p} \leq boundingBoxHeight / labelheight \quad (4.31)$$

Wenn einer der *values* den Wert Eins oder größer besitzt existiert eine konfliktfreie Beschriftungsposition:

$$\max(value_{1,m}, value_{2,n}, value_{3,o}, value_{4,p}) \geq 1 \Leftrightarrow \text{es existiert eine konfliktfreie Beschriftungsposition} \quad (4.32)$$

Das führt in Anlehnung an die zuvor entwickelten Bewertungsfunktionen zu:

$$\begin{aligned} labelScore_{\max} &= \max(value_{1,m}, value_{2,n}, value_{3,o}, value_{4,p}) \\ 0 \leq labelScore_{\max} &\leq \max(boundingBoxWidth / labelWidth, boundingBoxHeight / labelHeight) \end{aligned} \quad (4.33)$$

bzw. von $labelScore_{sum}$:

$$\begin{aligned} labelScore_{sum,1} &= \sum_{v=1}^m value_{1,v}, & labelScore_{sum,2} &= \sum_{v=1}^n value_{2,v}, \\ labelScore_{sum,3} &= \sum_{v=1}^o value_{3,v}, & labelScore_{sum,4} &= \sum_{v=1}^p value_{4,v}, \end{aligned} \quad (4.34)$$

wenn $m, n, o, p \neq 0$

und folglich:

$$labelScore_{sum} = labelScore_{sum,1} + labelScore_{sum,2} + labelScore_{sum,3} + labelScore_{sum,4} \quad (4.35)$$

Für die kombinierte Bewertung ergibt sich unter Verwendung von $maxLabelPositions = 2 * boundingBoxWidth / labelWidth + 2 * boundingBoxHeight / labelHeight$:

$$labelScore_{combination} = \begin{cases} \frac{labelScore_{sum}}{maxLabelPositions}, & \text{wenn } labelScore_{\max} < 1 \\ 1 + \frac{labelScore_{sum}}{maxLabelPositions}, & \text{wenn } labelScore_{\max} \geq 1 \end{cases}, \quad 0 \leq labelScore_{combination} \leq 2 \quad (4.36)$$

Das entwickelte Verfahren zur Ermittlung und Bewertung konfliktfreier Bewegungsräume bzw. konfliktfreier Beschriftungspositionen ist in Algorithmus 4 umgesetzt. Dieses für die Ermittlung der Beschriftungsschwierigkeit entwickelte Verfahren wird in Abschnitt 4.3.11 zentral für alle Beschriftungsklassen in das Verfahren zur Erzeugung des reaktiven Konfliktgraphen integriert. Es bietet sich auch eine Kombination mit dem Kriterium maximale Anzahl an Konflikten mit Objekten höherer Priorität (4.3.3.1) an, da das Kriterium nicht differenzieren kann sobald kein konfliktfreier Bewegungsraum verbleibt. Dies wird im nächsten Abschnitt beschrieben.

Die Laufzeit für die Ermittlung der Beschriftungsschwierigkeit eines punkthaften Objekts, Algorithmus 4, kann wie folgt nach oben abgeschätzt werden ($g \equiv$ Anzahl der Gebiete; $p \equiv$ Anzahl der Konflikte, d.h. Anzahl der zu beschriftenden punkthaften Objekten):

- Zeile 1 - 20 (Initialisierung): $O(g)$
- Zeile 21 - 33 (Bestimmung der Konfliktflächen und der x - und y -Werte): $O(p * g)$
- Zeile 34 - 43 (Bewertung der vier Bewegungsräume – Algorithmus 3 und analoge) $O(g)$
- Zeile 44 - 46 (Schleife zur Bestimmung von $labelScore_{\max}$ und $labelScore_{sum}$): $O(g)$

Daraus ergibt sich eine obere Schranke für die Laufzeit für das Verfahren von $O(p * g)$.

```

1  Input: Ein zu beschriftendes punkthafes Objekt mit Beschriftungsraum und Labelbox
        sowie Konfliktpartner und deren Beschriftungsräume zum vorgegebenen
        Maßstab, zu dem die Beschriftungsschwierigkeit bestimmt werden soll.
        (Verwendung von Algorithmus 3)
5  Output: Bewertungen der Beschriftungsschwierigkeit des betrachteten punkthaften
        Objekts zu vorgegebenem Maßstab mit dem Kriterium freie
        Beschriftungsfläche und den drei entwickelten Bewertungsfunktionen:
        labelScoremax, labelScoresum und labelScorecombination (vom Typ Float)
// Initial für das betrachtete zu beschriftende Objekt zu einem Maßstab:
10 x1..4 = labelWidth
    y1..4 = labelHeight
    i = aufgerundet(boundingBoxWidth/labelWidth) - 2
    j = aufgerundet(boundingBoxHeight/labelHeight) - 2
    x5,n,1 = x5,n,2 = x7,n,1 = x7,n,2 = labelWidth    für  $1 \leq n < i$ 
15 x5,i,1 = x5,i,2 = x7,i,1 = x7,i,2 = boundingBoxWidth%labelWidth
    y6,n,1 = y6,n,2 = y8,n,1 = y8,n,2 = labelHeight    für  $1 \leq n < j$ 
    y6,j,1 = y6,j,2 = y8,j,1 = y8,j,2 = boundingBoxHeight%labelHeight
// für jeden x- und y-Wert muss der Ursprung bestimmt werden (Bezugspunkt). Die
// Gebiete 1..4 in den Ecken werden gesondert behandelt.
20 bestimme Bezugspunkte bpg,1 & bpg,2 für jedes Gebiet g (bpg,1==bpg,2 für  $1 \leq g \leq 4$ )
    für jeden Konfliktpartner
        bestimme konfliktfläche
        bestimme Konfliktgebiet bzw. Konfliktgebiete
        für jedes beteiligte Konfliktgebiet
25     case Konfliktgebiet g = 1, 2, 3, oder 4
            xg = minimum(xg, horizontaleDistanz(bpg,1, konfliktfläche))
            yi = minimum(yi, vertikaleDistanz(bpg,1, konfliktfläche))
            case Konfliktgebiet g = 5 oder 7
                xg,1 = minimum(xg,1, horizontaleDistanz(bpg,1, konfliktfläche))
30             xg,2 = minimum(xg,2, horizontaleDistanz(bpg,2, konfliktfläche))
                case Konfliktgebiet g = 6 oder 8
                    yg,1 = minimum(yg,1, vertikaleDistanz(bpg,1, konfliktfläche))
                    yg,2 = minimum(yg,2, vertikaleDistanz(bpg,2, konfliktfläche))
// für jede Bewegungsrichtung Anzahl der konfliktfreien Bewegungsräume (m) und
35 // normierten Werte (value) bestimmen
// für den horizontalen Bewegungsraum oben:
bestimme m1 und value1,n mit Algorithmus 3
// für den vertikalen Bewegungsraum links:
bestimme m2 und value2,n mit Verfahren analog zu Algorithmus 3
40 // für den horizontalen Bewegungsraum unten:
bestimme m3 und value3,n mit Verfahren analog zu Algorithmus 3
// für den vertikalen Bewegungsraum rechts:
bestimme m4 und value4,n mit Verfahren analog zu Algorithmus 3
für jedes s, t, u, v ( $s \leq m1$ ,  $t \leq m2$ ,  $u \leq m3$ ,  $v \leq m4$ ):
45     labelScoremax = maximum(value1,s, value2,t, value3,u, value4,v)
        labelScoresum = summe(value1,s, value2,t, value3,u, value4,v)
maxLabelPositions =  $2 * \text{boundingBoxWidth} / \text{labelWidth} + 2 * \text{boundingBoxHeight} / \text{labelHeight}$ 
wenn labelScoremax < 1
    dann labelScorecombination = labelScoresum / maxLabelPositions
50 sonst labelScorecombination =  $1 + \text{labelScore}_{\text{sum}} / \text{maxLabelPositions}$ 

```

Algorithmus 4 Bestimmung der Beschriftungsschwierigkeiten eines punkthaften Objekts zu einem festen Maßstab für den allgemeinen Fall, dass die Signatur eine Ausdehnung hat. In den Zeilen 9 bis 17 erfolgt die Initialisierung der x- und y-Werte jedes Gebiets (soweit vorhanden), bevor in den Zeilen 18 bis 33 die Konfliktflächen berücksichtigt werden. Dazu wird zu jeder Bewegungsrichtung die Anzahl der konfliktfreien Bewegungsräume bestimmt (*m1*, *m2*, *m3* und *m4*) sowie die zugehörigen normierten Bewertungen (*value_{1,0..m1}*, *value_{1,0..m2}*, *value_{1,0..m3}* und *value_{1,0..m4}*) (Zeilen 34 bis 43). Zur Ermittlung dieser Werte wird Algorithmus 3 genutzt bzw. analoge Verfahren. In den Zeilen 44 bis 47 werden die Bewertungen *labelScore_{max}* und *labelScore_{sum}* bestimmt. Dazu werden das Maximum aller *value*-Werte gebildet bzw. alle *value*-Werte aufaddiert, wie durch die Schleife in Zeile 44 angedeutet. In den Zeilen 47 bis 50 erfolgt die Bestimmung von *labelScore_{combination}*.

Zu dem Verfahren „Freie Beschriftungsfläche“ ist eine kurze Bemerkung angebracht: Auf den ersten Blick wirkt die Ermittlung der \bar{x} und \bar{y} -Vektoren umständlich und unnötig. Man könnte z.B. auf der Grenze des Beschriftungsraums entlang gehen und immer, wenn ein Bereich der Grenze konfliktfrei ist, dieses dementsprechend deklarieren. Konfliktfreie Bereiche werden in Abhängigkeit, ob sie horizontal oder vertikal liegen, mit der Breite bzw. Höhe der Labelbox getestet. Ist der Test erfolgreich, der Bereich ist größer als die entsprechende Labelbox-Seite, kann eine Labelbox konfliktfrei in diesem Bereich platziert werden (Abbildung 44 a)). Wie in Abbildung 44 b) dargestellt, ist dieses Verfahren fehlerhaft, da die Konfliktfreiheit nicht für Breite und Höhe der zu platzierenden Labelbox garantiert wird. Die Erweiterung des Verfahrens auf Berücksichtigung beider Dimensionen führt zu dem in diesem Abschnitt vorgestellten Verfahren, das Vektoren verwendet.

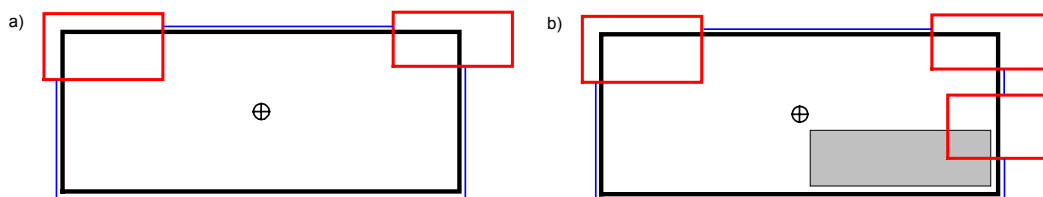


Abbildung 44 Die Ermittlung einer konfliktfreien Beschriftungsposition durch Durchlaufen der Grenze des Beschriftungsraums ist in einfachen Fällen, wie in a) dargestellt, erfolgreich. Obwohl in b) der Bereich der unteren Grenze als konfliktfrei erkannt ist, kommt es bei der Platzierung der Labelbox unten rechts zu einem Konflikt.

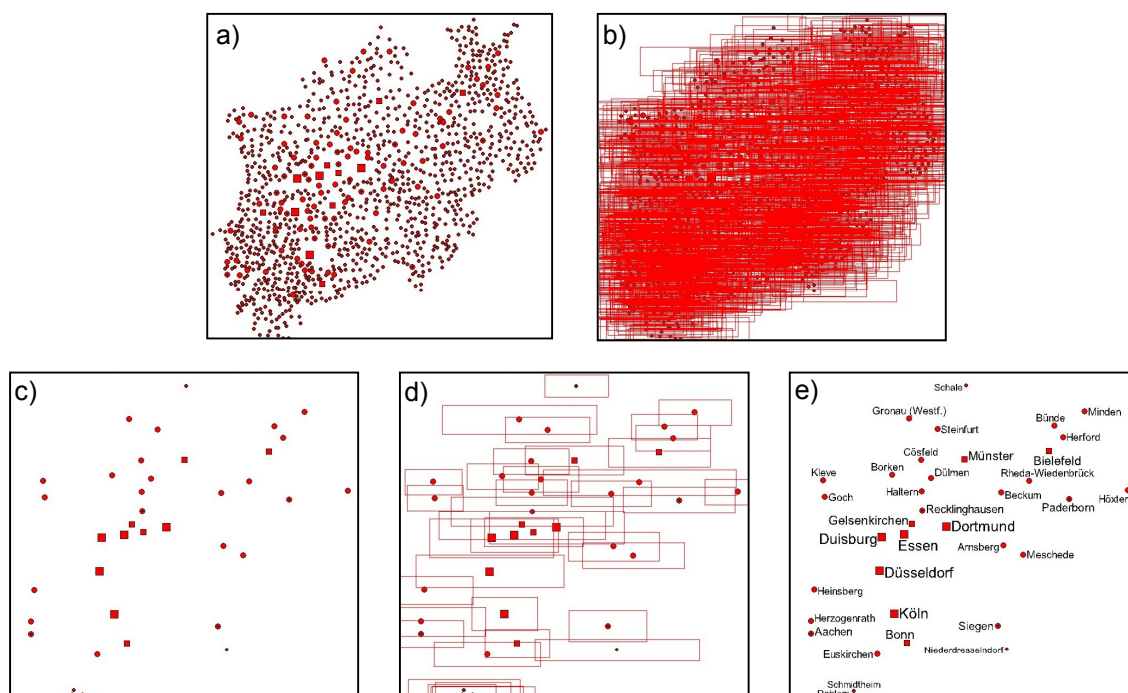


Abbildung 45 Vorselektion zu beschriftender Objekte: alle zu beschriftende Objekte sind in a) und mit Beschriftungsraum in b) dargestellt; die Selektion in c) bis e).

4.3.3.4 Zusammenfassung – Verfahren zur Bestimmung der Beschriftungsschwierigkeit

In den Abschnitten 4.3.3.1 bis 4.3.3.3 wurden verschiedene Kriterien zur Bewertung der Beschriftungsschwierigkeit vorgestellt und ihre Vor- und Nachteile diskutiert ohne sie zu ver-

gleichen. In diesem Abschnitt erfolgt ein Überblick und ein Vergleich zwischen ihnen, bevor ein Fazit gezogen wird. Die Beschriftungsschwierigkeit dient als Heuristik im Beschriftungsprozess für Bildschirmkarten, um die Anzahl der zu beschriftenden Objekte für den zu einem vorgegebenen Maßstab aufgerufenen Auswahlalgorithmus auf ein angemessenes Maß zu reduzieren (Abbildung 45). Eine Diskussion darüber erfolgte in Abschnitt 4.3.3. Eine Zusammenstellung der im folgenden beschriebenen Eigenschaften ist in Tabelle 3 zu finden.

	Knotengrad – Anzahl der Konflikte (4.3.3.1)	größter freier Winkel (4.3.3.2)	freie Beschriftungsfläche (4.3.3.3)
Aussage über Anzahl der Konfliktpartner	ja	nein	nein
Aussage über verbleibende konfliktfreie Beschriftungsfläche(n)	nein	ja	ja
Berücksichtigung der geometrischen Lage der Konflikte	nein	ja	ja
Aussage bei vollständiger Überlappung des Beschriftungsraums	ja	nein	nein
Laufzeit	+	–	0
Integration und Nutzung im Auswahlalgorithmus	–	–	+
Erweiterbarkeit auf andere Beschriftungsklassen	–	–	+

Tabelle 3 Vergleich der entwickelten Kriterien zur Bestimmung der Beschriftungsschwierigkeit (Bewertung: +(gut), 0(zufriedenstellend) und –(schlecht).

▪ Knotengrad

Die Vorteile des Kriteriums Knotengrad (4.3.3.1) ist im Vergleich zu den beiden anderen Kriterien größter freier Winkel (4.3.3.2) und freie Beschriftungsfläche (4.3.3.3) die geringe Laufzeit ($O(n)$, wobei n die Anzahl der Kanten der inzidenten Kanten des zu beschriftenden Objekts im reaktiven Konfliktgraphen ist) und, dass auch bei einer vollständigen Überlappung des Beschriftungsraums eine Quantifizierung erfolgen kann.

Die geometrische Lage der Konflikte bzw. Konfliktflächen und eine Aussage über den verbleibenden Beschriftungsraum kann mit diesem Kriterium nicht gewonnen werden, wie einleitend in Abschnitt 4.3.3.2 dargestellt wurde (vgl. Abschnitt 6.6 und Abbildung 84). Eine Verwendung des Kriteriums für die Ableitung einer Beschriftungsposition im Auswahlalgorithmus in der Interaktionsphase ist wegen der fehlenden Berücksichtigung der Geometrie nicht möglich.

Wegen der fehlenden Einbeziehung der räumlichen Ausdehnung ist der Einsatz des Kriteriums Knotengrad für die Beschriftung linienhafter Objekte ungeeignet. Die Anzahl der zulässigen Konflikte könnte zwar in Abhängigkeit z.B. des Verhältnisses der Länge des linienhaften Objekts zur Länge des zu platzierenden Schriftobjekts gewählt werden. Jedoch schließt das keine Gleichverteilung der Konflikte ein. Z.B. gibt es am Rhein im Ruhrgebiet, Frankfurter-Raum und Baseler-Raum eine hohe Anzahl an Konflikten, während an anderen Stellen keine Konflikte auftreten und somit trotz einer insgesamt hohen Anzahl an Konflikten

konfliktfreie Beschriftungsmöglichkeiten existieren. Eine mögliche Segmentierung des linienhaften Objekts führt auch nicht zum Ziel. In den Abschnitten 4.3.4 und 4.3.4.2 wird diese Thematik ausführlich behandelt.

Die anderen beiden entwickelten Kriterien machen im Unterschied zum Kriterium Knotengrad nicht eine Aussage über die Konflikte sondern über den verbleibenden konfliktfreien Beschriftungsraum. Sie stellen also einen reziproken Wert dar. Es wird deutlich, dass die Bewertung der Beschriftungsschwierigkeit vom verwendeten Verfahren abhängig ist.

- größter freier Winkel

In dem folgenden Ansatz, dem des größten freien Winkels (4.3.3.2), geht die Lage der Konflikte in Form von Konfliktsektoren ein. Dabei treten drei wesentliche Probleme auf. In der Regel werden Flächen als Konfliktflächen markiert die keine sind, da der Konfliktsektor immer von der Mitte des Beschriftungsraums bis zu seiner Grenze verläuft. Da die (Konflikt-)Sektoren und somit auch der größte freie Winkel nicht zwangsläufig achsenparallel bzw. parallel zur platzierenden Labelbox sind, kommt es in der Regel zu Überlappungen zwischen Sektoren und möglichen Beschriftungspositionen. Diese Eigenschaften führen dazu, dass eine Überlappung nicht automatisch einen Konflikt zwischen Labelbox und Konfliktflächen impliziert. Außerdem ist die Beurteilung des größten freien Winkels von seiner Orientierung, wie in 4.3.3.2 diskutiert, abhängig. Durch die Konstruktion der Winkel werden insbesondere die Beschriftungspositionen in den vier Ecken des Beschriftungsraums bevorzugt.

Sobald es zu einer vollständigen Überlappung des Beschriftungsraums kommt, kann das Kriterium des größten freien Winkels nicht weiter differenzieren. Eine Kombination mit dem Kriterium Knotengrad ist daher sinnvoll (s. Ende des Abschnitts). Um aus den ermittelten Konfliktsektoren den größten freien Winkel abzuleiten, müssen diese nach ihrer Orientierung sortiert werden, bevor der größte freie Winkel bestimmt werden kann, was zeitaufwendig ist ($O(n \log n)$, wobei n die Anzahl der Kanten der inzidenten Kanten des zu beschriftenden Objekts im reaktiven Konfliktgraphen sind).

- freie Beschriftungsfläche

Das Kriterium der freien Beschriftungsfläche (4.3.3.3) ist eine Weiterentwicklung des Kriteriums größter freier Winkel. Im Unterschied zu dem letztgenannten Kriterium wird durch die Quadranten (einfacher Fall) bzw. die Gebiete (allgemeiner Fall) explizit der verbleibende konfliktfreie Beschriftungsraum in Abhängigkeit der zu platzierenden Labelbox modelliert und nicht die Konfliktflächen und daraus abgeleitet der verbleibende Beschriftungsraum. Dadurch ist die Aussage des Kriteriums in Hinblick auf Beschriftungsmöglichkeiten präziser als die des freien Winkels. Das Kriterium ist so präzise, dass es, wie in Kapitel 5 beschrieben, zur Bestimmung der (endgültigen) Beschriftungsposition im Auswahlalgorithmus verwendet bzw. adaptiert wird. Die Erweiterbarkeit für die Bestimmung der Beschriftungsschwierigkeit linienhafter Objekte ist, wie in 4.3.6 gezeigt, möglich. Die Laufzeit ist von der Anzahl der Konfliktpartner und von der Anzahl der Gebiete abhängig, wie in 4.3.3.3.2 dargestellt. Wenn die Anzahl der Gebiete als konstant angesehen wird, ergibt sich somit eine Laufzeit von $O(n)$, wobei n die Anzahl der Kanten der inzidenten Kanten des zu beschriftenden Objekts im reaktiven Konfliktgraphen sind. Wegen dem konstanten Faktor ist die Laufzeit schlechter als die des Knotengrads.

Sobald es zu einer vollständigen Überlappung des Beschriftungsraums kommt, kann auch dieses Kriterium, analog zum Kriterium größter freier Winkel, nicht weiter differenzieren. Eine Lösung wird im Folgenden dargestellt.

- Kombination von Kriterien

Wie dargestellt hat das Kriterium freie Beschriftungsfläche einige Vorteile gegenüber den beiden anderen entwickelten Kriterien. Eine Schwachstelle ist die Bewertung bei vollständiger Überlappung des Beschriftungsraums. Dabei stellen sich zwei Fragen: Wann tritt dieser Fall ein bzw. wie wird das Kriterium eingesetzt? Wie kann dieses Problem gelöst werden?

Das Kriterium zur Bestimmung der Beschriftungsschwierigkeit wird bei der Generierung des reaktiven Konfliktgraphen verwendet (4.3.11). Dabei werden nicht alle Konflikte zu dem betrachteten zu beschriftenden Objekt berücksichtigt, sondern nur die zu Objekten mit einer höheren Priorität. Das bedeutet, dass zu einem zu beschriftenden Objekt zu allen Konflikten in dem betrachteten Maßstab z.B. fünf Konflikte (Knotengrad) und eine *labelScore*-Bewertung von 0,0 (freie Beschriftungsfläche) vorliegt. Da der Algorithmus zur Generierung des reaktiven Konfliktgraphen aber nur Konflikte zu Objekten mit höherer Priorität berücksichtigt, wird eine Bewertung für diese Situation von z.B. drei Konflikten und eine *labelScore*-Bewertung von 0,7 ermittelt. Es tritt also eine „Entspannung“ für das Bewertungskriterium freie Beschriftungsfläche ein.

Durch eine Kombination des Kriteriums freie Beschriftungsfläche mit dem Kriterium Knotengrad können diese Bewertungsprobleme vollständig gelöst werden. Am sinnvollsten ist folglich ein zweistufiges Bewertungskriterium:

$$\text{Beschriftungsschwierigkeit} = \begin{cases} -\text{Bewertung}_{\text{Knotengrad}} & , \text{ wenn } \text{Bewertung}_{\text{freieFläche}} = 0 \\ \text{Bewertung}_{\text{freieFläche}} & , \text{ sonst} \end{cases} \quad (4.37)$$

Wenn die so ermittelte Beschriftungsschwierigkeit größer eins ist, existiert mindestens eine konfliktfreie Beschriftungsposition. Bei einem Wert unter eins existiert keine konfliktfreie Beschriftungsposition mehr und bei einem Wert kleiner gleich Null ist der Beschriftungsraum vollständig überlappt. Diese Bewertungsfunktion fließt in die Generierung des reaktiven Konfliktgraphen in 4.3.11 ein.

4.3.4 Linienhafte Objekte

Analog zu den punkthafte zu beschriftenden Objekten wird im nächsten Abschnitt der Beschriftungsraum für zu beschriftende linienhafte Objekte in Anlehnung an den für punkthafte Objekte behandelt bevor die Bestimmung von Schnittmaßstäben in Abschnitt 4.3.5.1 erfolgt. Für die Bewertung der Konflikte bzw. für das Deselektionskriterium (Abschnitte 4.3.5.2 und 4.3.6) sowie die Integration in den reaktiven Konfliktgraphen werden die für punkthafte Objekte entwickelten Techniken adaptiert.

Linienhafte zu beschriftende Objekte werden, wie punkthafte als Knoten im Konfliktgraphen, sowohl im reaktiven als auch im statischen, repräsentiert. Dabei geht die Anschaulichkeit, die in der intuitiven Assoziation zwischen der Geometrie des reaktiven Konfliktgraphenknotens und der Geometrie des zu beschriftenden Objekts liegt, verloren (Abbildung 46).

In Abbildung 46 ist der „Globalisierungseffekt“, der auf Grund der geometrischen Ausdehnung linienhafter Objekte und ihrer Beschriftungsräume auftreten kann, dargestellt. Diese Globalisierung von Konflikten führt zu einem Zusammenschmelzen von lokalen Verdichtungen. Um diesem entgegenzuwirken, wird in Abschnitt 4.3.4.2 die Segmentierung zu beschriftender linienhafter Objekte sowie die dadurch entstehenden Vor- und Nachteile diskutiert.

Für die Konflikte zwischen linienhaften Objekten (4.3.5) und zwischen punkt- und linienhaften Objekten (4.3.7) gilt analog die gleiche Motivation, wie für die Konflikte zwischen punkthafte Objekten. Diese wurde ausführlich in Abschnitt 4.3.2 besprochen und

dargestellt. Daher ist die Motivation in den nächsten Abschnitten auf spezifische Unterschiede beschränkt.

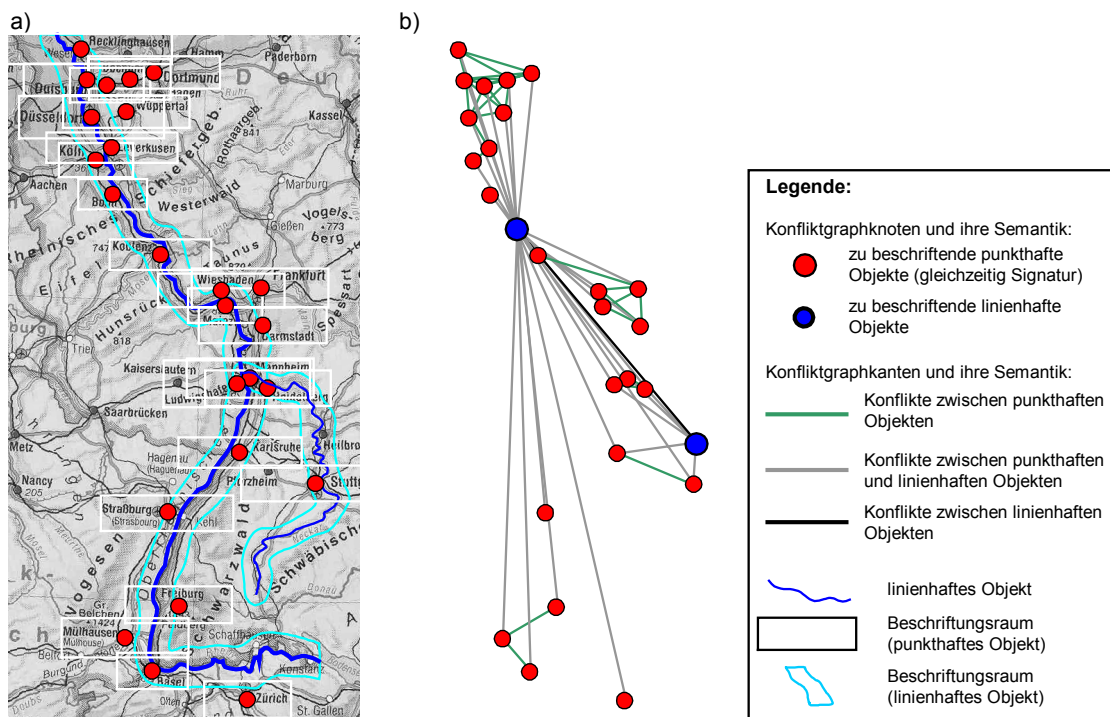


Abbildung 46 Konfliktgraph mit zu beschriftenden linienhaften Objekten. Durch die Integration zu beschriftender linienhafter Objekte geht die assoziative Zuordnung zwischen Geometrie des zu beschriftenden Objekts und der Repräsentation im Konfliktgraphen verloren. (Karte in a) [DIE76])

4.3.4.1 Beschriftungsraum

Bei der Beschriftung linienhafter Objekte werden die Vorteile des Konzepts des Beschriftungsraums deutlich. Bisherige Verfahren (z.B. [ECM96]) erzeugen eine Menge möglicher diskreter Beschriftungspositionen (in Abbildung 47 a) die fünf Labelboxen mit Schrift), aus der der Auswahlalgorithmus die lokal beste Position wählt. Im Gegensatz zu den bisherigen Verfahren ist dieses nicht auf einen diskreten Maßstab beschränkt, sondern unterstützt ein Kontinuum, das sinnvoller- aber nicht notwendigerweise durch einen oberen und einen unteren Maßstab beschränkt ist, auf Grund der zu Grunde liegenden Daten bzw. ihres Volumens und damit der Handhabung. Durch die Änderung des Maßstabs erfolgt gleichzeitig eine Änderung der Schrift in Höhe und Breite. Die Änderung der Höhe der Schrift erfolgt auf der vom linienhaften Objekt abgewandten Seite. Es stellen sich verschiedene Fragen: Wie kann eine Breitenänderung berücksichtigt werden? Soll die Labelbox zu ihrer Lage im Ursprungsmaßstab linksbündig, rechtsbündig oder zentriert ausgerichtet werden? Wie verhält es sich, wenn es dadurch zu „unschönen“ Knicken kommt? Gibt es keine diskreten, sondern kontinuierliche Beschriftungspositionen, kann in diesem Fall die Lage der Schrift „etwas verschoben“ werden, bis eine passende Stelle gefunden ist. Diese Idee lässt sich über das Konzept des Beschriftungsraums realisieren.

Die Schrift eines zu beschriftenden linienhaften Objekts wird über oder unter dem Linienzug in einem festen Abstand platziert, wie in Abschnitt 2.1.1.3 beschrieben. Eine Beschriftung auf dem linienhaften Objekt kann erfolgen, wenn dieses eine ausreichende Signaturbreite im gewählten Maßstab besitzt. Dieses ist in der Regel bei großmaßstäbigen Karten, wie z.B. Stadtplänen, der Fall. Da das Verfahren des reaktiven Konfliktgraphen für mittel- und

kleinmaßstäbige Karten entwickelt wurde, wird auf eine explizite Berücksichtigung dieses Falls verzichtet. Eine entsprechende Erweiterung des hier vorgestellten Konzepts und Verfahrens ist offensichtlich.

Abbildung 47 zeigt ein linienhaftes Objekt, das mehrfach mit dem zu platzierenden Schriftobjekt beschriftet ist, und den zugehörigen Beschriftungsraum.

Definition (linienhaftes Objekt):

Ein (zu beschriftendes) linienhaftes Objekt ist eine Polylinie, die sich aus geraden Liniensegmenten zusammensetzt.

Auf eine Modellierung mit Splines wurde verzichtet, da diese zu einer Verwendung von komplexeren und laufzeitintensiveren Algorithmen führt und sie zum anderen durch feine Segmentierung approximiert werden können. Der Beschriftungsraum eines linienhaften Objekts ist äquivalent zu dem Puffer des Objekts mit der Schrift- bzw. Labelboxhöhe und dem Abstand zwischen Objekt und Schrift (Abbildung 47 a) und b)). Die Schrift „schlängelt“ sich dabei am linienhaften Objekt entlang. Der Puffer und somit auch der Beschriftungsraum umschließt vollständig den Anfang und das Ende des Linienzugs, obwohl dort keine Schrift platziert wird. Für eine effiziente Ermittlung der Konflikte wird dieser Nachteil eines zu großen Beschriftungsraums in Kauf genommen. Bei der Bestimmung der Beschriftungsschwierigkeit und bei der Beschriftung wird dies jedoch berücksichtigt.

Der Puffer mit Höhe h eines linienhaften Objekts ist die Vereinigung der Flächen aller Kreise, deren Mittelpunkte auf dem Objekt liegen und deren Radien h sind. Diese Definition wird später zur Bestimmung der Beschriftungskonflikte verwendet.

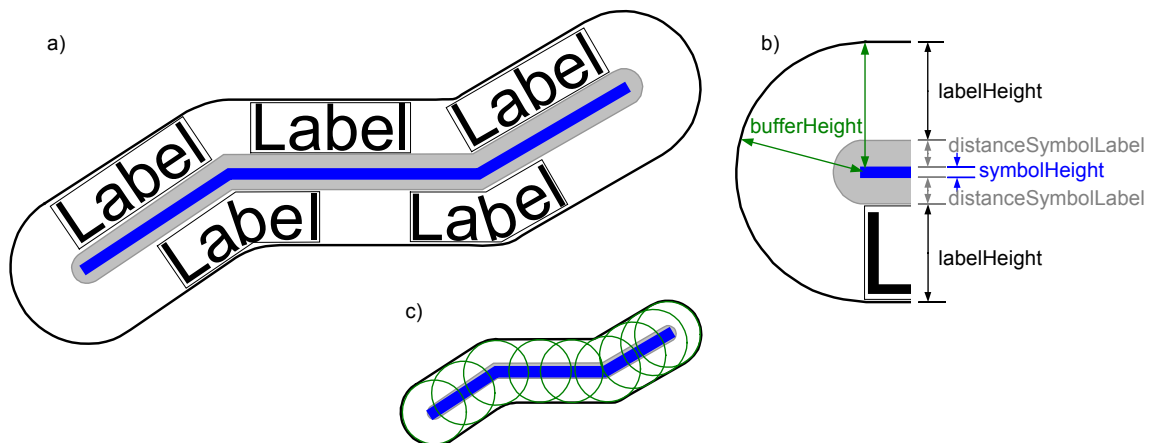


Abbildung 47 Beschriftungsraum eines linienhaftes Objekts.

In Anlehnung an den Beschriftungsraum punkthafter Objekte (Abschnitt 4.3.1.1) und mit der in 2.1.2 entwickelten Skalierungsfunktion kann die für die Verwendung im reaktiven Konfliktgraph nötige Parametrisierung mit dem Maßstab erfolgen. Im Folgenden wird die für den Beschriftungsraum punkthafter Objekte in Abschnitt 4.3.1.1 adaptierte Skalierung verwendet.

Ermittlung des Beschriftungsraums für linienhafte Objekte (parametrisiert mit dem Maßstab):

- Gegeben (in Realwelt-Koordinaten):
 - Positionen der i Stützpunkte der $i-1$ Liniensegmente: $p_{i.x}, p_{i.y}$
 - Referenzmaßstab für Labelbox, Signatur und Abstand: $fixScale$
 - Referenzbreite der Signatur: $fixSymbolHeight$
 - Referenzabstand zwischen Signatur und Schrift: $fixDistanceSymbolLabel$
 - Referenzlabelbox passend zum Referenzmaßstab der zu platzierenden Schrift: $fixWidth, fixHeight$
- Skalierungsfunktionen (vgl. Abschnitt 2.1.2 und 4.3.1.1):
Die folgenden Skalierungsfunktionen können aus Abschnitt 4.3.1.1 übernommen werden und sind mit ihrer Gleichungsnummer aufgelistet:

$scaling(scale, fixScale)$ aus (4.1)

$referenceScaling(scale, fixScale)$ aus (4.3)

$labelWidth(scale, fixScale, fixWidth)$ aus (4.4)

$labelHeight(scale, fixScale, fixHeight)$ aus (4.5)

$symbolWidth(scale, fixScale, fixSymbolWidth)$ aus (4.6)

$symbolHeight(scale, fixScale, fixSymbolHeight)$ aus (4.7)

$distanceSymbolLabel(scale, fixScale, fixDistanceSymbolLabel)$ aus (4.8)

- Höhe des Puffers:
 - Höhe:
 - Die Höhe des Puffers im Zielmaßstab ($scale$) in Abhängigkeit der Referenzlabelbox, -signatur und -abstand sowie dem zugehörigen Referenzmaßstab ergibt sich aus der halben Signaturhöhe plus dem Abstand zwischen Signatur und Schriftobjekt plus der Schrifthöhe für den gewählten Maßstab jeweils abhängig vom Referenzwert und Referenzmaßstab.
 - Formal ergibt sich:

$$bufferHeight = \frac{symbolHeight}{2} + distanceSymbolLabel + labelHeight \quad (4.38)$$

durch einsetzen der Funktionen (4.5), (4.6), (4.3) und (4.1):

$$bufferHeight = \left(fixHeight + fixDistanceSymbolLabel + \frac{fixSymbolHeight}{2} \right) * \left(\frac{scale}{fixScale} \right)^{value} \quad (4.39)$$

4.3.4.2 Segmentierung des Beschriftungsraums

Cluster sind Gebiete lokaler Verdichtungen, wie z.B. das Ruhrgebiet, der Frankfurter-Raum und der Baseler-Raum, die in Abbildung 48 a) dargestellt sind. Sie können bis zu einem gewissen Grad unabhängig voneinander behandelt werden und erlauben den Einsatz von „Divide and Conquer“- und Parallelverarbeitungstechniken¹⁵, die zur Reduktion der Laufzeit beitragen können.

¹⁵ Um aus der Segmentierung Vorteile in der Beschriftungsgeschwindigkeit während der Interaktionsphase erzielen zu können, muss eine Mehrprozessor-Hardware zur Verfügung stehen und der Auswahlalgorithmus parallele Abläufe zulassen. Dadurch können die Cluster parallel bearbeitet und beschriftet werden.

Wie einleitend in Abschnitt 4.3.4 angesprochen, kommt es jedoch durch die geometrische Ausdehnung von zu beschriftenden linienhaften Objekten zu einer „Globalisierung“ der Konflikte, was zu einer Verschmelzung von Clustern führt (Abbildung 48 b)). Um dem entgegenzuwirken, kann das linienhafte Objekt segmentiert werden. Dabei wird das linienhafte Objekt nicht mehr durch einen Knoten repräsentiert sondern jedes Segment durch einen, wie in Abbildung 48 c) dargestellt. Im Folgenden werden Vor- und Nachteile der Segmentierung diskutiert.

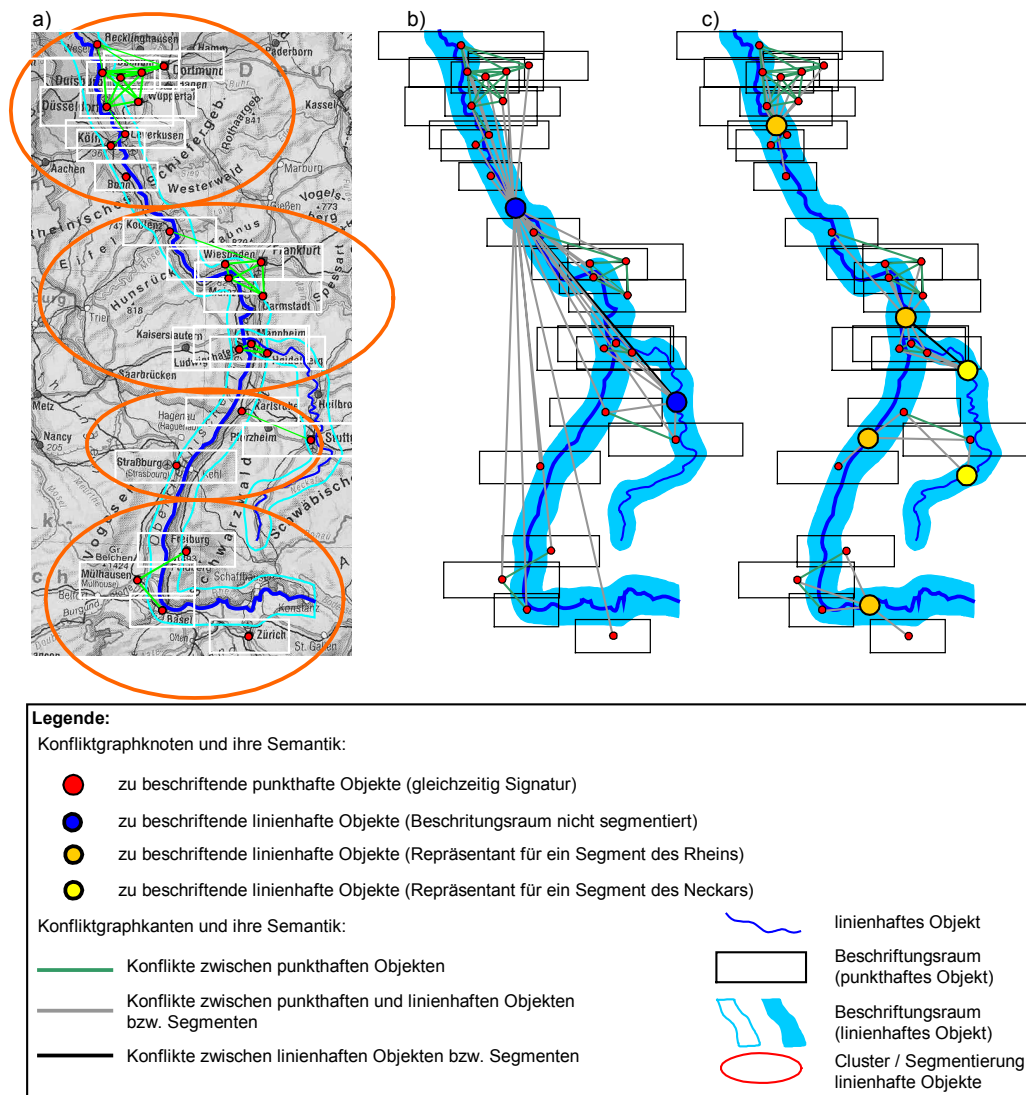


Abbildung 48 Integration von zu beschriftenden linienhaften Objekten in den Konfliktgraphen ohne Segmentierung b) und mit Segmentierung c). (Karte in a) entnommen aus [DIE76])

Es stellt sich die Frage, wie bzw. wann die Segmentierung erfolgen kann. Abbildung 48 c) zeigt die Segmentierung und den zugehörigen Konfliktgraphen zu einem spezifischen Maßstab. Bei einem anderen Maßstab oder einem anderen Ausschnitt muss die Segmentierung, damit sie die gewünschten Effekte erzielen kann, erneut durchgeführt werden. Das bedeutet, dass die Segmentierung während der Interaktionsphase erfolgen muss, die eigentlich entlastet werden soll.

Wie sieht eine „gute“ Segmentierung und was bedeutet das für die Beschriftung des linienhaften Objekts? Abbildung 48 a) zeigt eine scheinbar „gute“ Segmentierung der linienhaften

Objekte Rhein und Main, um eine maximale Anzahl an Clustern zu erhalten, die in diesem Beispiel nur durch die beiden segmentierten linienhaften Objekte verbunden sind.

Wie bzw. wo soll nun die Beschriftung der segmentierten linienhaften Objekten erfolgen? Ein Ansatz wäre für jedes Segment von zu beschriftenden linienhaften Objekten die beste Beschriftungsposition zu ermitteln und von diesen die beste für das zu beschriftende Objekt auszuwählen. Bei diesem Vorgehen wird die Beschriftung in den Clustern, also den Konfliktzentren, die eigentlich bei der Beschriftung vermieden werden sollten, bevorzugt. Die aus kartographischer Sicht besser geeigneten Beschriftungspositionen liegen genau auf den Grenzen dieser Segmentierung. Die Segmentgrenzen linienhafter Objekte sollten daher in den Konfliktzentren liegen, so dass konfliktarme Bereiche nicht „zerlegt“ werden. Das steht aber im Widerspruch zur Cluster-Segmentierung.

Was ändert sich für die zu beschriftenden punkthaften Objekte bei einer Segmentierung? Der Konflikt und die Lage des Konflikts mit dem linienhaften Objekt bleibt bestehen, unabhängig davon, ob er mit dem Objekt an sich oder einem Segment des segmentierten Objekts erfolgt.

Bei näherer Betrachtung und dem Vergleich der Konfliktgraphen in Abbildung 48 b) und c) ist ein weiterer Nachteil erkennbar: die Anzahl der Konfliktgraphknoten nimmt zu. Das bedeutet gleichzeitig eine Erhöhung der Laufzeit in der Interaktionsphase, wie in Kapitel 5 dargestellt werden wird.

Aus diesen Gründen wurde auf eine weitere Vertiefung der Segmentierung verzichtet.

4.3.5 Beschriftungskonflikt: linienhaftes – linienhaftes Objekt

Der Beschriftungsraum linienhafter Objekte ist mit denen im vorletzten Abschnitt entwickelten Formeln definiert. Dieser ist, wie der Beschriftungsraum punkthafter Objekte, abhängig vom Maßstab. Analog zu diesem entstehen bei abnehmenden Maßstab Überlappungen der Beschriftungsräume und damit potentielle Beschriftungskonflikte. Abbildung 49 zeigt an einem Kartenbeispiel Beschriftungskonflikte zwischen linienhaften Objekten zu einem festen Maßstab.

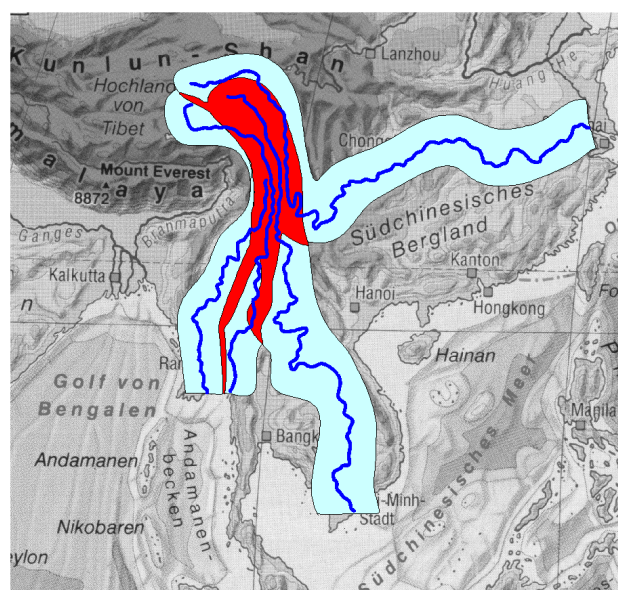


Abbildung 49 Beschriftungsräume linienhafter Objekte (hellblau) und ihre Konflikte (rot) am Beispiel der Flüsse Irawadi, Saluen, Mekong und Jangtsekiang in Südostasien. (Karte [DIE96])

Die Bestimmung des Schnittmaßstabs, der Maßstab, ab und unter dem es zwischen den Räumen zu einem Konflikt kommt, erfolgt im nächsten Abschnitt. Dazu wird die Puffer-eigenschaft des Beschriftungsraums des linienhaften Objekts genutzt, sowie Formeln, die für die Ermittlung von Konflikten zwischen Beschriftungsräumen punkthafter Objekte entwickelt wurden.

4.3.5.1 Schnittmaßstab

Der Schnittmaßstab ist der Maßstab, ab und unterhalb dem es zu einem Konflikt zwischen den Beschriftungsräumen bzw. den Puffern zweier linienhafter Objekte kommt (vgl. Abschnitt 4.3.2.1).

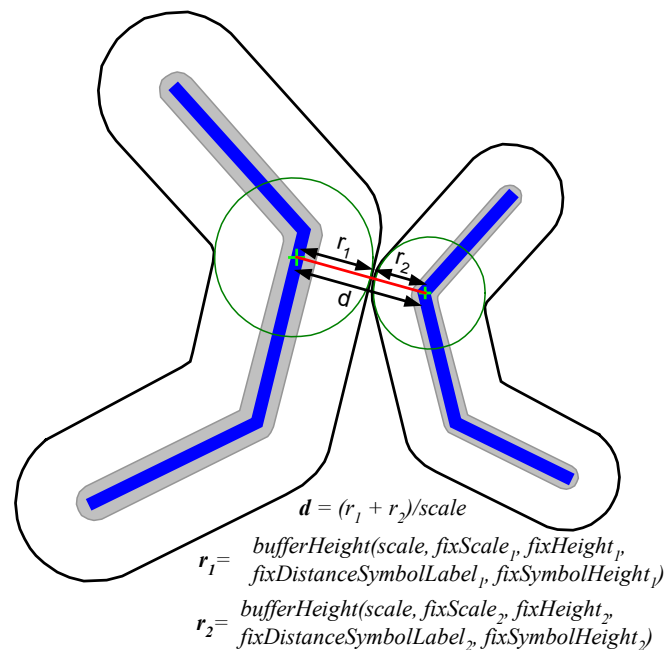


Abbildung 50 Berechnung des Schnittmaßstabs zwischen zwei linienhaften zu beschriftenden Objekten (Abbildungsmaßstab gleich Schnittmaßstab).

Analog zur Bestimmung des Schnittmaßstabs punkthafter Objekte (Abschnitt 4.3.2.1) wird der Maßstab berechnet, bei dem es zu einer Berührung [EGH90] zwischen den beiden Beschriftungsräumen kommt (Abbildung 50). Dazu wird zunächst der kleinste Abstand im Ursprungsmaßstab zwischen den beiden linienhaften Objekten ermittelt (Abbildung 50 Distanz d), wofür Standardverfahren der Computergeometrie verwendet werden können ([GLA90] und [ARV91]). Nun kann die Eigenschaft ausgenutzt werden, dass die beiden Beschriftungsräume jeweils aus Kreisen gleicher Radien konstruiert sind (4.3.4.1). Die Lage sowie die Anzahl der Berührungen zwischen den Beschriftungsräumen sind somit ohne Auswirkung. In Abbildung 50 ist die Situation der Berührung der beiden Beschriftungsräume bzw. der Kreise grün visualisiert. Die Summe der beiden Radien r_1 und r_2 , die abhängig vom Maßstab sind, werden auf die Distanz d gesetzt und nach dem Maßstab aufgelöst.

Mit der Distanz im Ursprungsmaßstab (Realwelt) gilt analog zu den punkthaften Objekten (4.3.2.1) mit Gleichung (4.39), die Parameter der Funktionen r_1 und r_2 sind aus Gründen der Übersichtlichkeit weggelassen:

$$d = \frac{r_1 + r_2}{scale} = \frac{\left(fixHeight_1 + fixDistanceSymbolLabel_1 + \frac{fixSymbolHeight_1}{2} \right) * \left(\frac{scale}{fixScale_1} \right)^{value}}{scale} + \frac{\left(fixHeight_2 + fixDistanceSymbolLabel_2 + \frac{fixSymbolHeight_2}{2} \right) * \left(\frac{scale}{fixScale_2} \right)^{value}}{scale} \quad (4.40)$$

durch Umformung und aufgelöst nach $scale$ ergibt sich:

$$scale = \left(\frac{d}{\frac{fixHeight_1 + fixDistanceSymbolLabel_1 + \frac{fixSymbolHeight_1}{2}}{fixScale_1^{value}} + \frac{fixHeight_2 + fixDistanceSymbolLabel_2 + \frac{fixSymbolHeight_2}{2}}{fixScale_2^{value}}} \right)^{\frac{1}{value-1}} \quad (4.41)$$

Im Unterschied zur Berechnung des Schnittmaßstabs zwischen punkthaften Objekten muss nicht zwischen der x- und y-Richtung differenziert werden. Die Berechnung des Schnittmaßstabs ist ansonsten identisch.

4.3.5.2 Deselektionsmaßstab

Für den Deselektionsmaßstab von Konflikten zwischen zwei linienhaften Objekten trifft die gleiche Motivation und Vorgehensweise zu, wie für den Deselektionsmaßstab zwischen punkthaften Objekten in Abschnitt 4.3.2.2. Die Bestimmung des Deselektionsmaßstabs ist direkt abhängig von der Beschriftungsschwierigkeit für linienhafte Objekte, die im nächsten Abschnitt vorgestellt wird.

Die Integration linienhafter Objekte und der Konflikte zwischen linienhaften Objekten in den Konfliktgraphen erfolgt zentral in Abschnitt 4.3.11.

4.3.6 Beschriftungsschwierigkeit und Deselektion: linienhaftes Objekt

Das Verfahren zur Ermittlung der Beschriftungsschwierigkeit linienhafter Objekte ist eine Weiterentwicklung des Verfahrens „freie Beschriftungsfläche“ zur Bestimmung der Beschriftungsschwierigkeit punkthafter Objekte aus Abschnitt 4.3.3.3.

Es wird keine Unterscheidung zwischen linienhaften oder punkthaften Konfliktpartnern gemacht, da es bei beiden zu Mehrfachüberlappungen kommen kann. Das Beispiel in Abbildung 51 ist auf einen Konflikt mit einem punkthaften Objekt beschränkt, das analog ohne Einschränkung ein linienhaftes Objekt sein kann. Es können mehrere Konfliktflächen zwischen den beiden Konfliktpartnern eintreten. Jede Konfliktfläche ist die minimal umschließende Fläche des betrachteten Konflikts, wobei die Längen der Seitenstücke dieser Fläche, die den Beschriftungsraum durchqueren, die minimale Distanz zwischen dem Objekt und dem Rand des Beschriftungsraums besitzen und die anderen Seitenstücke am Rand des Beschriftungsraums bzw. am Objekt entlanglaufen. Wie in Abbildung 51 b) zu erkennen, steht nicht der vollständige Puffer als Beschriftungsraum zur Verfügung, sondern ein um die orangenen Flächen am Anfang und Ende des linienhaften Objekts reduzierter, da sich die Beschriftung nicht um die Enden des linienhaften Objekts „herumschlängelt“. Der Nachteil des etwas zu großen Puffers im Vergleich zum Beschriftungsraum wird in Kauf genommen, um eine

analytische Berechnung der Schnittmaßstäbe ohne die Behandlung zu vieler Sonderfälle durchzuführen (vgl. auch Abschnitt 4.3.7.1).

```

1  Input: Ein zu beschriftendes linienhaftes Objekt mit Beschriftungsraum und Labelbox
    sowie Konfliktpartner und deren Beschriftungsräume zum vorgegebenen
    Maßstab, zu dem die Beschriftungsschwierigkeit bestimmt werden soll.
Output: Bewertungen der Beschriftungsschwierigkeit des betrachteten linienhaften
5  Objekts zum vorgegeben Maßstab mit den entwickelten Bewertungsfunktionen:
    ohne Bewertung der Beschriftungsstelle:
         $labelScore_{max}$ ,  $labelScore_{sum}$  und  $labelScore_{sumLabelWidth}$  (Typ Float)
    mit Bewertung der Beschriftungsstelle:
         $labelScore_{maxQuality}$ ,  $labelScore_{sumQuality}$  und  $labelScore_{sumLabelWidthQuality}$  (Typ Float)
10 // Konfliktfläche = Schnitt der Beschriftungsflächen der Konfliktpartner
    // Start-/Enddistanz = Wegstrecke zwischen Startpunkt des linienhaften Objekts und dem
    // darauf projizierten Punkt
    // SeiteA = Beschriftungsraum bzw. Puffer rechts des linienhaften Objekts
    // SeiteB = Beschriftungsraum bzw. Puffer links des linienhaften Objekts
15 // stripeA und stripeB sind Listen
    für jeden Konfliktpartner
        für jeden Konflikt bzw. Konfliktfläche mit SeiteA
            berechne Start- und Enddistanz der aktuell betrachteten Konfliktfläche
            mit SeiteA projiziert auf Linienobjekt
20         füge Tupel (Startdistanz, Enddistanz) zu stripeA hinzu
        für jeden Konflikt bzw. Konfliktfläche mit SeiteB
            berechne Start- und Enddistanz der aktuell betrachteten Konfliktfläche
            mit SeiteB projiziert auf Linienobjekt
            füge Tupel (Startdistanz, Enddistanz) zu stripeB hinzu
25 durchlaufe beide stripe-Listen und ermittle konfliktfreie Bereiche, teile sie
    durch labelWidth und speichere sie in dem Array conflictFree[] und die
    Bewertung der Beschriftungsstelle in dem Array conflictFreeQuality[]
    // conflictFree[] ist ein Array von Tripeln (Startdistanz, Enddistanz, Seite)
    // conflictFree[] und conflictFreeQuality[] haben jeweils beide length Einträge
30 // Beschriftungsschwierigkeit ohne Bewertung der Beschriftungsstelle
     $labelScore_{max}$  = Maximum von conflictFree
     $labelScore_{sum}$  = Summe von conflictFree
     $labelScore_{sumLabelWidth}$  = Summe der conflictFree-Einträge, die größer 1 sind
    // Beschriftungsschwierigkeit mit Bewertung der Beschriftungsstelle
35  $labelScore_{maxQuality}$  =  $\max_{j=1}^{length} (conflictFree[j] * conflictFreeQuality[j])$ 
    , wobei  $conflictFree[j] > 1$ 
     $labelScore_{sumQuality}$  =  $\sum_{j=1}^{length} conflictFree[j] * conflictFreeQuality[j]$ 
     $labelScore_{sumLabelWidthQuality}$  =  $\sum_{j=1}^{length} conflictFree[j] * conflictFreeQuality[j]$  , wobei  $conflictFree[j] > 1$ 

```

Algorithmus 5 Verfahren zur Bestimmung der Beschriftungsschwierigkeit eines linienhaften Objekts. Der Typ des Konfliktpartners ist unerheblich, da für jeden Konflikt die Konfliktfläche(n) bestimmt werden und diese in die Bestimmung eingehen (Abbildung 51).

Algorithmus 5 zeigt ein Verfahren zur Bestimmung der Beschriftungsschwierigkeit und damit der Deselektion für linienhafte Objekte. Im Folgenden wird dieser Algorithmus besprochen.

- Algorithmus 5 – Zeile 1 bis 24

Die Bestimmung der Konfliktbereiche erfolgt in Zeile 17 und 21. Es können mehrere Bereiche pro Konfliktpartner existieren, da wegen der Geometrie des linienhaften Objekts und seines Beschriftungsraums mehrere nicht zusammenhängende Konfliktbereiche auftreten können. Nur die Konflikte mit dem Beschriftungsraum sind dabei von Interesse, nicht die mit dem Objekt bzw. der Signatur. Dabei wird unterschieden, auf welcher „Seite“ des linienhaften Objekts sie liegen. Die Seite eines linienhaften Objekts ist durch die Reihenfolge der Seg-

mente bzw. der Anfangs- und Endpunkte bestimmt. Die Konfliktflächen werden auf das linienhafte Objekt projiziert, wie in Abbildung 51 dargestellt. Die orange eingefärbten Flächen am Anfang und Ende des Puffers des linienhaften Objekts, in denen keine Beschriftung erfolgen soll (Abbildung 51 b)), werden durch die getrennte Betrachtung von *SeiteA* und *SeiteB* berücksichtigt und brauchen daher nicht explizit gemacht werden (Abbildung 51 c) und d)). In den beiden Listen *stripeA* und *stripeB* werden die Konfliktbereiche mit ihrem Anfangs- und Endpunkt mittels der Distanz zum Startpunkt des („abgewickelten“) linienhaften Objekts (Abbildung 51 c) und d)) als Tupel (*stripeStart*, *stripeEnd*) gespeichert.

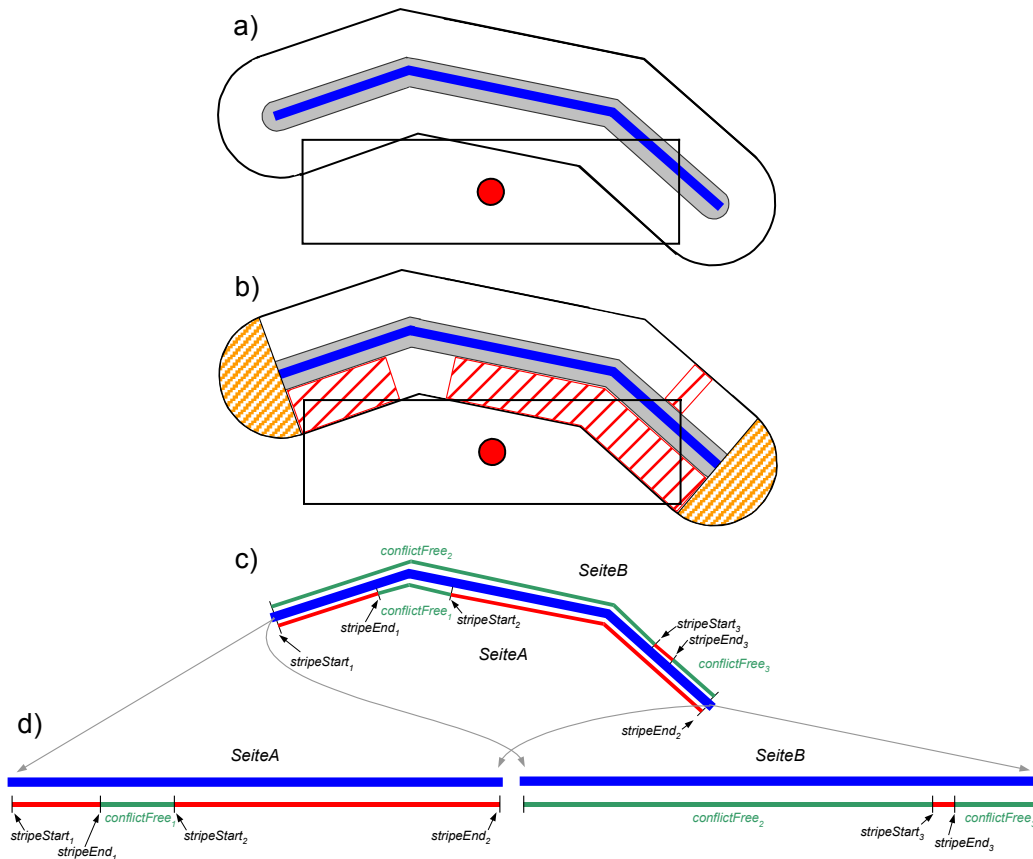


Abbildung 51 Konflikt eines zu beschriftenden linienhaften Objekts mit einem zu beschriftenden punkthaften Objekt: a) Konflikt zwischen den beiden Beschriftungsräumen; b) Konfliktflächen des linienhaften Objekts sind rot und die Flächen des Puffers, die nicht zur Beschriftung zur Verfügung stehen, orange dargestellt; c) auf das linienhafte Objekt projizierte Konfliktflächen (rot) und konfliktfreie Bereiche (grün); d) „abgewickelte“ linienhafte Objekt getrennt nach *SeiteA* und *SeiteB*.

▪ Algorithmus 5 – Zeile 25 bis 29

Aus den ermittelten *Stripe*-Werten werden die zusammenhängenden konfliktfreien Bereiche ermittelt (*conflictFree*). Das sind die Bereiche, die nicht in einem Tupel von *stripeStart* und *stripeEnd* liegen (grün in Abbildung 51 c) und d)). Diese Bereiche werden analog zu den *Stripe*-Listen in dem Array *conflictFree* durch ihre Anfangs- und Endpunkte mittels der Distanz zum Startpunkt des linienhaften Objekts als Tripel zusammen mit der Seite als Tripel (*Start*, *Ende*, *Seite*) gespeichert. Für die Bewertungsfunktionen werden in den Zeilen 25 bis 27 die Längen der konfliktfreien Bereiche durch die Länge der zu platzierenden Schrift geteilt und dann in dem Array *conflictFree* gespeichert für beide Seiten. *conflictFree*-Werte größer Eins repräsentieren Beschriftungspositionen in dem Beschriftungsraum des linienhaften Objekts, in die die zu platzierende Schrift vollständig und konfliktfrei passt.

Zu jedem konfliktfreien Bereich wird eine Bewertung der Beschriftungsposition erstellt. Diese bewertet den betrachteten konfliktfreien Bereich zum zu beschriftenden Objekt und die geometrischen Eigenschaften des Bereichs. In die Bewertung geht die Lage zum Start- und Endpunkt des linienhaften Objekts (Zentrierung), die Position zum linienhaften Objekt (unter, über bzw. rechts, links), die Krümmung des Beschriftungsraums an dieser Stelle ein. Detaillierte Informationen zu diesen Bewertungen können [ECM96], [PET96], [PEP97] und [ELL01] entnommen werden.

- Algorithmus 5 – Zeile 30 bis 33

Die drei Bewertungsfunktionen $labelScore_{max}$, $labelScore_{sum}$ und $labelScore_{sumLabelWidth}$ berücksichtigen nur den freien Raum, in dem die Höhe des zu platzierenden Labels hineinpasst. Es bleibt zu testen, ob das zu beschriftende Objekt hinsichtlich der Länge passt. Da die *conflictFree*-Werte normiert sind, durch Teilung durch die Länge der zu platzierenden Schrift, repräsentiert jeder Wert größer Eins einen ausreichend großen Beschriftungsraum (analog zu Abschnitt 4.3.3.3). Aus $labelScore_{max}$ und $labelScore_{sumLabelWidth}$ kann eindeutig gefolgert werden, ob eine konfliktfreie Beschriftungsposition existiert: Das ist genau dann der Fall, wenn sie einen Wert größer Eins liefern. Während $labelScore_{max}$ nur die längste Beschriftungsstelle berücksichtigt, gehen in $labelScore_{sumLabelWidth}$ alle Beschriftungsstellen ein, die mindestens die Länge der Schrift haben. Aus der Bewertung der einfachen Summe der normierten Werten ($labelScore_{sum}$) kann nicht auf eine konfliktfreie Beschriftungsposition geschlossen werden, auch wenn der Wert größer Eins ist: Dies wird in Abschnitt 4.3.3.3 diskutiert.

- Algorithmus 5 – Zeile 34 bis 38

Die Eignung der Beschriftungsstelle geht in die Bewertungen $labelScore_{maxQuality}$, $labelScore_{sumQuality}$ und $labelScore_{sumLabelWidthQuality}$ ein. Wie in dem Fall zuvor werden in $labelScore_{maxQuality}$ und $labelScore_{sumLabelWidthQuality}$ nur die Beschriftungsstellen berücksichtigt, die mindestens die Länge der zu platzierenden Schrift haben, kombiniert mit der „Qualität“ der Beschriftungsstelle, die in *conflictFreeQuality* abgelegt ist.

Die Laufzeit für die Ermittlung der Beschriftungsschwierigkeit eines linienhaften Objekts in Algorithmus 5 ist von der Anzahl der Konflikte abhängig (Doppelschleife Zeile 17 bis 20 bzw. 21 bis 24). Im Worst-Case besteht das betrachtete zu beschriftende linienhafte Objekt aus der maximalen Anzahl an (Linien-)Segmenten s . Es kommt zu Konflikten mit allen $l-1$ anderen zu beschriftenden linienhaften Objekten, die auch jeweils die maximale Anzahl an (Linien-) Segmenten (s) besitzen. Dadurch entstehen bei jedem Konflikt $O(s^2)$ Konfliktflächen bzw. konfliktfreie Flächen. Das ergibt eine obere Schranke der Laufzeit von $O(l*s^2)$ ($l \equiv$ Anzahl der zu beschriftenden linienhaften Objekte und $s \equiv$ maximale Anzahl der (Linien-) Segmente zu beschriftender linienhafter Objekte).

4.3.7 Beschriftungskonflikt: punkthafes – linienhaftes Objekt

Neben Konflikten zwischen Beschriftungsräumen punkthafter Objekte und zwischen Beschriftungsräumen linienhafter Objekte kann es abhängig vom Maßstab zu Konflikten zwischen Beschriftungsräumen punkthafter und linienhafter Objekte kommen. Dies ist beispielhaft in Abbildung 52 dargestellt. Analog zu den anderen beiden Konfliktarten gibt es auch hier einen ausgezeichneten Maßstab, den Schnittmaßstab, bei und unter dem es zu einer Überlappung zwischen den beiden beteiligten Beschriftungsräumen kommt. Die Bestimmung des Schnittmaßstabs erfolgt im nächsten Abschnitt.



Abbildung 52 Der Konflikt zwischen Beschriftungsräumen punkthafter und linienhafter Objekte illustriert am Beispiel der Hauptstadt des Bundesstaates Mississippi, Jackson, und dem Mississippi. (Karte [RMN98]).

4.3.7.1 Schnittmaßstab

Die Ermittlung des Schnittmaßstabs zwischen punkthaftern und linienhaften Objekten ist wegen den unterschiedlichen Ausprägungen der Beschriftungsräume diffiziler und führt zu Fallunterscheidungen. Die Ursache liegt in der unterschiedlichen x- und y-Ausdehnung der Beschriftungsräume punkthafter Objekte.

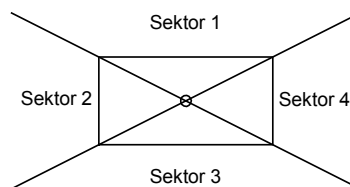


Abbildung 53 Das zu beschriftende punkthafte Objekte mit Beschriftungsraum und Sektoren für die Bestimmung des Schnittmaßstabs mit linienhaften Objekten.

Für die Fallunterscheidung wird die Ebene ausgehend vom punkthaftern Konfliktpartner und seinem Beschriftungsraum in vier Sektoren aufgeteilt, wie in Abbildung 53 dargestellt. Die Sektorengrenzen sind die Verlängerungen der Diagonalen des Rechtecks, das den Beschriftungsraum beschreibt, ausgehend vom Mittelpunkt des punkthaftern Objekts. Da der Beschriftungsraum in x- und in y-Richtung proportional zueinander wächst, sind die Sektorengrenzen fest. Das linienhafte Objekt kann nun in einem oder mehreren Sektoren liegen. Um den Schnittmaßstab, dies ist auch hier der Maßstab, ab und unterhalb dem es zu einem Konflikt zwischen den Beschriftungsräumen kommt, zu ermitteln muss dieser für jeden Sektor separat bestimmt werden. Der größte Schnittmaßstab der Sektoren ist der gesuchte Schnittmaßstab zwischen den Konfliktpartnern.

Dieses Verfahren ist in Algorithmus 6 beschrieben. Es werden dazu zunächst die Schnittmaßstäbe der Sektoren bestimmt. Innerhalb der Sektoren erfolgt eine weitere Differenzierung bevor der (endgültige) Schnittmaßstab bestimmt werden kann. Algorithmus 6 greift dabei auf Funktionen zurück, die im Folgenden hergeleitet und anhand von Abbildung 53 erläutert werden.

```

1  Input: Ein zu beschriftendes linienhaftes Objekt mit Beschriftungsraum und
    ein zu beschriftendes punkthafte Objekt zwischen denen der Schnitt-
    maßstab bestimmt werden soll.
    Output: Schnittmaßstab von punkt- und linienhaften zu beschriftenden Objekt
5  // ermittle Konfliktsektoren  $confSector_i$  des punkthaften Objekts
    für jeden  $confSector_i$ 
        ermittle nächstliegenden Punkt  $p_i$  des linienhaften Objekts
        berechne Schnittmaßstab  $cutScale_i$  mit  $calculateCutScale_1$ 
        berechne Beschriftungsraum punkthafte Objekt mit  $cutScale_i$ 
10 wenn  $p_i$  nicht in der Verlängerung des Beschriftungsraums liegt
        dann
            berechne Schnittmaßstab  $cutScale_i$  mit  $calculateCutScale_2$ 
            ermittle Sektordistanz  $a''$ 
            wenn  $a'' \leq$  Sektordistanz von Liniensegment
15             berechne Schnittmaßstab  $cutScale_i$  mit  $calculateCutScale_3$ 
    Schnittmaßstab =  $\max(cutScale_i)$ 

```

Algorithmus 6 Verfahren zur Berechnung des Schnittmaßstabs zwischen den Beschriftungsräumen punkthafter und linienhafter Objekte. Der Text präzisiert die zum Teil unscharfen Ausdrücke des Algorithmus.

- Algorithmus 6 – Zeile 1 bis 6

Zunächst wird ermittelt, in welchen Sektoren (Abbildung 53) das linienhafte Objekt liegt. Für jeden dieser Sektoren erfolgt die Bestimmung des Schnittmaßstabs in den Zeilen 7 bis 15 zunächst separat.

- Algorithmus 6 – Zeile 7 bis 9

Nun kann die Eigenschaft ausgenutzt werden, dass der Beschriftungsraum des linienhaften Objekts aus Kreisen gleichen Radien konstruiert ist (4.3.4.1). Für die Bestimmung des Schnittmaßstabs für den jeweiligen Sektor ist es ausreichend den nächstliegenden Punkt des linienhaften Objekts zu betrachten von dem ausgehend der Puffer den ersten „Kontakt“ mit dem Beschriftungsraum des punkthaften Objekts hat (Abbildung 54 a) bis c) roter Punkt). Dabei wird nicht die euklidische Distanz zwischen dem linienhaften und punkthaften Objekt verwendet. Da der Beschriftungsraum des punkthaften Objekts ein achsenparalleles Rechteck ist, ist für den ersten und dritten Konfliktsektor der Punkt der am nächsten liegende, der den kleinsten vertikalen Abstand hat, für den zweiten und vierten derjenige mit dem kleinsten horizontalen Abstand (Abbildung 54 a) bis c) für die drei Konfliktsektoren – graugestrichelte parallele Geraden zeigen den Abstand zwischen dem Punkt des linienhaften Objekts, der am nächsten zum punkthaften liegt und dem des punkthaften Objekts).

Wegen der unterschiedlichen x- und y-Ausdehnung des Beschriftungsraums des punkthaften Objekts, wird abhängig vom Sektor der Schnittmaßstab mit folgenden Gleichungen – repräsentiert durch den Ausdruck $calculateCutScale_1$ in Algorithmus 6 Zeile 8 – bestimmt, die aus der Schnittmaßstabberechnung zwischen zwei punkthaften bzw. zwei linienhaften Objekten in 4.3.2.1 und 4.3.5.1 bzw. den Gleichungen (4.14), (4.15) bzw. (4.41) abgeleitet sind.

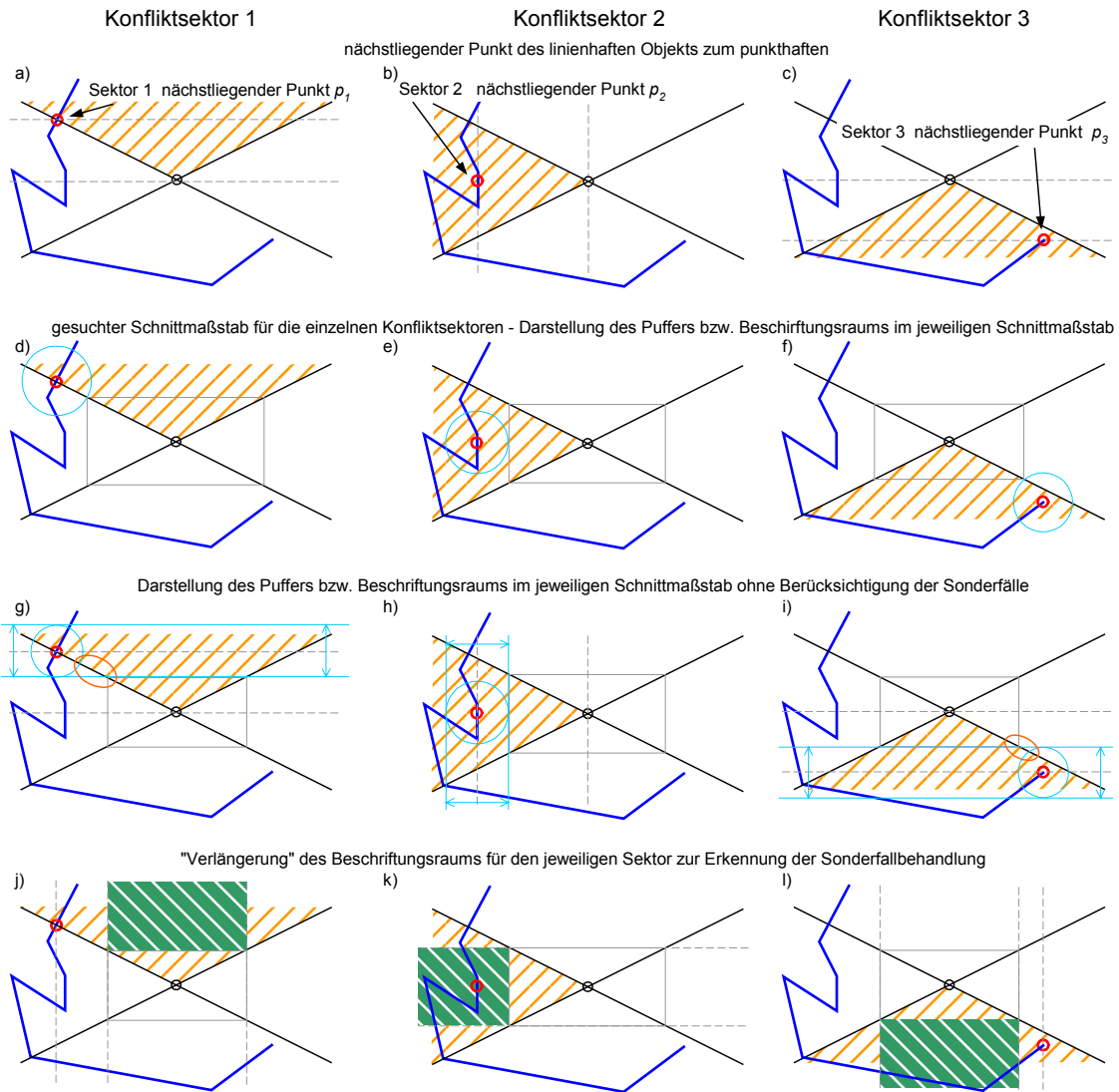


Abbildung 54 Beispiel zur Bestimmung des Schnittmaßstabs zwischen punkt- und linienhaften Objekten (Algorithmus 6). Das linienhafte Objekt liegt in dem ersten (linke Spalte), zweiten (mittlere Spalte) und dritten Konfliktsektor (rechte Spalte) des punkthaften Objekts. Liegt der nächstliegende Punkt des linienhaften Objekts (rot) nicht innerhalb der grünschraffierten Fläche j) bis l), liegt ein Sonderfall vor.

Für die Sektoren eins und zwei ergibt sich für $calculateCutScale_1$ – Index „1“ repräsentiert das punkthafte Objekt und „2“ das linienhafte Objekt, d ist die vertikale Distanz zwischen dem punkthaften Objekt und dem nächstliegenden Punkt des linienhaften (Differenz der y-Koordinaten):

$$cutScale_1 = \left(\frac{d}{\frac{fixHeight_1 + fixDistanceSymbolLabel_1 + \frac{fixSymbolHeight_1}{2}}{fixScale_1^{value}} + \frac{fixHeight_2 + fixDistanceSymbolLabel_2 + \frac{fixSymbolHeight_2}{2}}{fixScale_2^{value}}} \right)^{\frac{1}{value-1}} \quad (4.42)$$

und für die Sektoren zwei und drei ergibt sich für $calculateCutScale_1$ – Index „1“ repräsentiert das punkthafte Objekt und „2“ das linienhafte Objekt, d ist die horizontale Distanz zwischen dem punkthaften Objekt und dem nächstliegenden Punkt des linienhaften (Differenz der x-Koordinaten):

$$cutScale_i = \left(\frac{d}{\frac{fixWidth_1 + fixDistanceSymbolLabel_1 + \frac{fixSymbolWidth_1}{2}}{fixScale_1^{value}} + \frac{fixHeight_2 + fixDistanceSymbolLabel_2 + \frac{fixSymbolHeight_2}{2}}{fixScale_2^{value}}} \right)^{\frac{1}{value-1}} \quad (4.43)$$

Die möglichen Ergebnisse der mit diesen beiden Gleichungen berechneten Schnittmaßstäbe (*calculateCutScale₁* in Algorithmus 6 Zeile 8) sind in Abbildung 54 g) bis h) dargestellt. Die Abbildung macht deutlich, dass der Schnittmaßstab, ausgehend von der horizontalen (Sektor 1 und 3) bzw. vertikalen (Sektor 2 und 3) Geraden, die durch den nächstliegenden Punkt des linienhaften Objekts führt, bestimmt wird. Die Abbildung 54 g) und i) zeigen, dass es im ermittelten Schnittmaßstab nicht zu einer Berührung zwischen den Beschriftungsräumen bzw. Puffer kommen muss, orange dargestellt.

▪ Algorithmus 6 – Zeile 10 bis 13

Um diese Fehler abzufangen wird zunächst der Beschriftungsraum des punkthaften Objekts zu dem ermittelten Schnittmaßstab berechnet (Algorithmus 6 Zeile 9). In den Sektoren eins und drei treten diese Fehler auf, da der nächstliegende Punkt des linienhaften Objekts nicht über bzw. unter dem Beschriftungsraum liegt, sondern links oder rechts neben ihm (Abbildung 54 j) und l) die nächstliegenden Punkte des linienhaften Objekts liegen nicht in den grünschraffierten Flächen; k) der nächstliegende Punkt im Sektor zwei liegt in der grünschraffierten Fläche, d.h. der Schnittmaßstab ist ermittelt und es müssen keine Sonderfälle betrachtet werden).

Es können zwei Sonderfälle bei der Ermittlung des Schnittmaßstabs eintreten. Zunächst wird der erste Sonderfall betrachtet (Algorithmus 6 – Zeile 12). Führt dieser nicht zum Erfolg (Algorithmus 6 – Zeile 14) trifft der zweite Sonderfall zu (Algorithmus 6 – Zeile 15).

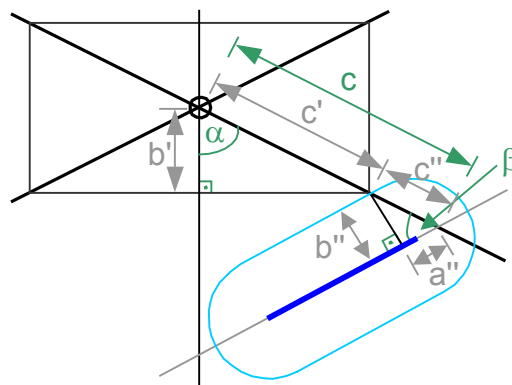


Abbildung 55 Erster Sonderfall (Algorithmus 6 – Zeile 12 *calculateCutScale₂*): der Punkt des linienhaften Objekts, der am nächsten dem punkthaften Objekt ist, liegt nicht in der Verlängerung des Beschriftungsraums des punkthaften Objekts (vgl. Abbildung 54). Die Beschriftungsräume (schwarz und hellblau) sind in dem gesuchten Schnittmaßstab dargestellt (d.h. es gibt eine Berührung). Grün sind die festen, vom Maßstab unabhängigen und als bekannt vorausgesetzten Größen dargestellt, grau die vom Maßstab abhängigen.

Im Folgenden werden die Sonderfälle diskutiert und Lösungen dazu vorgestellt. Es wird dabei exemplarisch angenommen, dass die Sonderfälle in Sektor 3 vorkommen, wie in Abbildung 55 dargestellt. Die Lösungen gelten für den dritten als auch den ersten Sektor, sowohl rechts als auch links neben dem Bereich des Beschriftungsraums des punkthaften Objekts. Für die Sektoren zwei und vier müssen die „Höhen“-Werte des punkthaften Objekts durch die

„Breiten“-Wert ersetzt werden, d.h. $fixHeight_{point}$ durch $fixWidth_{point}$, $fixSymbolHeight_{point}$ durch $fixSymbolWidth_{point}$. Die Berechnungen sind im übrigen analog und werden daher nicht weiter ausgeführt.

Abbildung 55 zeigt den ersten Sonderfall mit allen zur Lösung erforderlichen Größen. Grün sind die bekannten vom Maßstab unabhängigen Größen eingetragen, die geometrisch ermittelbar sind, und grau die vom Maßstab abhängigen. Für den Beschriftungsraum des punkthaften Objekts ergibt sich:

$$\cos \alpha = \frac{b'}{c'} \quad (4.44)$$

Dabei entspricht b' der halben Höhe des Beschriftungsraums. Die Funktion, die in Gleichung (4.12) dargestellt ist, liefert die Höhe des Beschriftungsraums im Schnittmaßstab. Da im Folgenden die Größen im Ursprungsmaßstab (Realwelt) angegeben werden, muss der Wert transformiert werden und es ergibt sich:

$$\cos \alpha = \frac{\left(fixHeight_{point} + fixDistanceSymbolLabel_{point} + \frac{fixSymbolHeight_{point}}{2} \right) * \left(\frac{cutScale}{fixScale_{point}} \right)^{value} * cutScale^{-1}}{c'} \quad (4.45)$$

aufgelöst nach dem Schnittmaßstab:

$$c' = \frac{\left(fixHeight_{point} + fixDistanceSymbolLabel_{point} + \frac{fixSymbolHeight_{point}}{2} \right) * \frac{cutScale^{value-1}}{fixScale_{point}^{value}}}{\cos \alpha} \quad (4.46)$$

Wie in Abbildung 55 erkennbar ist, ergibt sich analog für den Beschriftungsraum bzw. Puffer des linienhaften Objekts:

$$\sin \beta = \frac{b''}{c''} \quad (4.47)$$

Dabei entspricht b'' der Höhe des Beschriftungsraums bzw. Puffers des linienhaften Objekts. Mit Gleichung (4.39) und der Umrechnung auf den Ursprungsmaßstab ergibt sich:

$$\sin \beta = \frac{\left(fixHeight_{line} + fixDistanceSymbolLabel_{line} + \frac{fixSymbolHeight_{line}}{2} \right) * \left(\frac{cutScale}{fixScale_{line}} \right)^{value} * cutScale^{-1}}{c - c'} \quad (4.48)$$

mit dem in (4.46) hergeleiteten c' ergibt sich:

$$\sin \beta = \frac{\left(fixHeight_{line} + fixDistanceSymbolLabel_{line} + \frac{fixSymbolHeight_{line}}{2} \right) * \frac{cutScale^{value-1}}{fixScale_{line}^{value}}}{c - \left(\frac{\left(fixHeight_{point} + fixDistanceSymbolLabel_{point} + \frac{fixSymbolHeight_{point}}{2} \right) * \frac{cutScale^{value-1}}{fixScale_{point}^{value}}}{\cos \alpha} \right)} \quad (4.49)$$

Zur übersichtlichen Darstellung werden die konstanten Terme für die Beschriftungsräume punkthafter und linienhafter Objekte zusammengefasst und durch je eine Variable repräsentiert:

$$fix_{point} = fixHeight_{point} + fixDistanceSymbolLabel_{point} + \frac{fixSymbolHeight_{point}}{2} \quad (4.50)$$

$$fix_{line} = fixHeight_{line} + fixDistanceSymbolLabel_{line} + \frac{fixSymbolHeight_{line}}{2} \quad (4.51)$$

Dadurch ergibt sich für Gleichung (4.49):

$$\sin \beta = \frac{fix_{line} * \frac{cutScale^{value-1}}{fixScale_{line}^{value}}}{c - \left(\frac{fix_{point} * \frac{cutScale^{value-1}}{fixScale_{point}^{value}}}{\cos \alpha} \right)} \quad (4.52)$$

dieser Ausdruck nach dem Schnittmaßstab $cutScale$ aufgelöst:

$$cutScale = e^{\left(\frac{\ln \left(\frac{c \cos \alpha \sin \beta}{fix_{point} * fixScale_{point}^{-value} * \sin \beta + fix_{line} * fixScale_{line}^{-value} * \cos \alpha} \right)}{value-1} \right)} \quad (4.53)$$

Dabei entspricht $cutScale$ der Funktion $calculateCutScale_2$ in Algorithmus 6 – Zeile 12. Wie angesprochen muss untersucht werden, ob der ermittelte Schnittmaßstab stimmt. Bei diesem betrachteten Fall wird vorausgesetzt, dass die kürzeste Distanz zwischen dem linienhaften Objekt (dunkelblau) und dem „Touch“-Punkt, b'' in Abbildung 56, in einem rechten Winkel zum linienhaften Objekt steht. Der Beschriftungsraum des punkthaftern Objekts ist schwarz und der des linienhaften Objekts hellblau eingezeichnet. In Abbildung 56 a) bis c) wird durch die unterschiedlichen Distanzen a'' auf dem linienhaften Objekt und seiner Verlängerung zwischen dem Schnittpunkt des rechten Winkels und der festen, vom Maßstab unabhängigen verlängerten Diagonalen des Beschriftungsraums des punkthaftern Objekts, der „Freiraum“ des Schnitts des rechten Winkels deutlich.

Abbildung 56 d) bis f) zeigen, dass nicht der Schnittmaßstab ausgehend von einem speziellen Punkt auf dem linienhaften Objekt berechnet wurde, sondern von der Geraden, die das linienhafte Objekt und seine Verlängerung bildet. Der zugehörige Beschriftungsraum bzw. Puffer ist eine Verschiebung der Geraden.

Wenn der Schnittpunkt der kürzesten Distanz b'' , die im rechten Winkel zum linienhaften Objekt steht (Abbildung 56), nicht mehr auf dem linienhaften Objekt liegt, sondern auf der Verlängerung, tritt der nächste Sonderfall ein. Um dieses festzustellen, muss a'' berechnet (Algorithmus 6 – Zeile 13) und ausgewertet werden (Zeile 14).

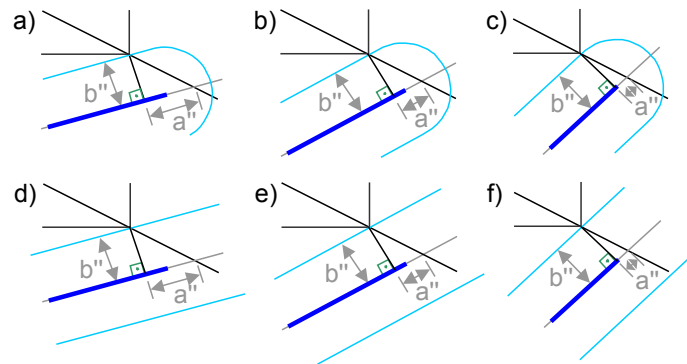


Abbildung 56 Bestimmung der Sektordistanz a'' aus Algorithmus 6 – Zeile 13.

▪ Algorithmus 6 – Zeile 14 bis 15

Aus Abbildung 55 folgt $\cos \beta = a''/c''$, wobei β bekannt ist und c'' abhängig vom ermittelten Schnittmaßstab ist. Mit dem in Gleichung (4.46) hergeleiteten c' folgt $c'' = c - c'$ also $a'' = (c - c') \cos \beta$ (β und c bekannt und fest, c' abhängig vom Maßstab).

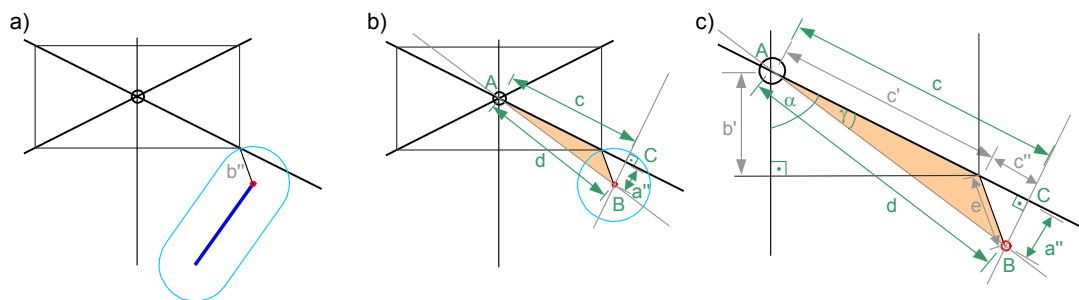


Abbildung 57 Zweiter Sonderfall (Algorithmus 6 – Zeile 15 *calculateCutScale₃*): Der Beschriftungsraum bzw. Puffer des linienhaften Objekts geht vom rot markierten Punkt B aus a). Bild c) zeigt einen vergrößerten Ausschnitt der Konstruktion aus b). Grün sind die festen, vom Maßstab unabhängigen und als bekannt vorausgesetzten Größen dargestellt, grau die vom Maßstab abhängigen.

Liegt der Schnittpunkt nicht auf dem linienhaften Objekt (Algorithmus 6 – Zeile 14), dann tritt der zweite Sonderfall ein (Zeile 15), wie in Abbildung 57 a) dargestellt. Im Gegensatz zum Standardfall und dem ersten Sonderfall wird nun der Beschriftungsraum bzw. Puffers des linienhaften Objekts zu einem bestimmten Punkt (Abbildung 57 b) hellblauer Kreis), dem Punkt, der am nächsten zum punkthaften Objekt liegt, ermittelt. Dieser ist entweder ein Endpunkt eines Liniensegments oder liegt auf einer Sektorgrenze (Abbildung 57 rot markierter Punkt B). In Abbildung 57 sind die bekannten, vom Maßstab unabhängigen Größen grün eingetragen und die maßstabsabhängigen grau. Mit Hilfe des Kosinussatzes, angewandt auf das gelbbraun eingefärbte Dreieck in Abbildung 57, lässt sich der Schnittmaßstab wie folgt bestimmen:

$$e^2 = c^2 + d^2 - 2c'd \cos \gamma \quad (4.54)$$

Dabei entspricht e der Höhe des Beschriftungsraums bzw. des Puffers des linienhaften Objekts, c' der halben Länge der Diagonalen des Beschriftungsraums des punkthaften Objekts, d der Distanz zwischen dem Mittelpunkt des punkthaften Objekts und dem Punkt, von dem der Beschriftungsraum bzw. Puffer des linienhaften ausgeht sowie γ dem eingeschlossenen Winkel zwischen d und c' .

c' ist in Gleichung (4.46) hergeleitet und auf den Ursprungsmaßstab (Realwelt) umgerechnet. Mit der in Gleichung (4.50) eingeführten Repräsentation für den konstanten Teil des Ausdrucks ergibt sich:

$$c' = \frac{fix_{point} * fixScale_{point}^{-value} * cutScale^{value-1}}{\cos \alpha} \quad (4.55)$$

Für die Höhe e des linienhaften Beschriftungsraums bzw. Puffers ergibt sich unter Verwendung von Gleichung (4.39) und der Umrechnung auf den Ursprungsmaßstab:

$$e = \left(fixHeight_{line} + fixDistanceSymbolLabel_{line} + \frac{fixSymbolHeight_{line}}{2} \right) * fixScale_{line}^{-value} * cutScale^{value-1} \quad (4.56)$$

und durch Repräsentation des konstanten Terms mit der in (4.51) eingeführten Variablen ergibt:

$$e = fix_{line} * fixScale_{line}^{-value} * cutScale^{value-1} \quad (4.57)$$

Für den Kosinussatz aus Gleichung (4.54) folgt:

$$\left(fix_{line} * fixScale_{line}^{-value} * cutScale^{value-1} \right)^2 = \left(\frac{fix_{point} * fixScale_{point}^{-value} * cutScale^{value-1}}{\cos \alpha} \right)^2 + d^2 + 2 \frac{fix_{point} * fixScale_{point}^{-value} * cutScale^{value-1}}{\cos \alpha} d \cos \gamma \quad (4.58)$$

Diese Gleichung nach dem Schnittmaßstab $cutScale$ aufgelöst führt zu zwei Lösungen:

$$cutScale_a = \left(\frac{-2 fixScale_{point}^{-value} fix_{point} \cos \gamma + 2 \sqrt{fix_{point}^2 fixScale_{point}^{-2value} \cos(\gamma)^2 + fix_{line}^2 fixScale_{line}^{-2value} \cos(\alpha)^2 - fix_{point}^2 fixScale_{point}^{-2value}}}{2 (fix_{line}^2 fixScale_{line}^{-2value} \cos(\alpha)^2 - fix_{point}^2 fixScale_{point}^{-2value})} d \cos \alpha \right)^{\left(\frac{1}{value-1} \right)}$$

$$cutScale_b = \left(\frac{-2 fixScale_{point}^{-value} fix_{point} \cos \gamma - 2 \sqrt{fix_{point}^2 fixScale_{point}^{-2value} \cos(\gamma)^2 + fix_{line}^2 fixScale_{line}^{-2value} \cos(\alpha)^2 - fix_{point}^2 fixScale_{point}^{-2value}}}{2 (fix_{line}^2 fixScale_{line}^{-2value} \cos(\alpha)^2 - fix_{point}^2 fixScale_{point}^{-2value})} d \cos \alpha \right)^{\left(\frac{1}{value-1} \right)}$$

(4.59)

Die zwei Lösungen resultieren aus den Quadraten in Gleichung (4.58) und dem zum Auflösen notwendigen Ziehen der Quadratwurzel. Abbildung 58 a) und b) zeigen die beiden möglichen Lösungen bzw. Dreiecke, die durch die nach dem Schnittmaßstab aufgelöste Gleichung (4.58) entstehen – Gleichung (4.59). Grund für die Existenz zweier Lösungen ist, dass es zwei Schnittpunkte (D – hellgrün dargestellt) zwischen dem hellblauen Kreis, dem Beschriftungsraum bzw. Puffer des linienhaften Objekts, der Diagonale des Beschriftungsraums des punkthaften Objekts und seiner Grenze gibt. Dabei ist der eingeschlossene Winkel zwischen c und d in beiden Abbildungen gleich, genauso wie die Kante d und die Lage der Punkte A und B . Da die Beschriftungsräume und damit die Kanten c und e vom Maßstab abhängig sind, haben diese in b) eine größere Ausdehnung.

Das Maximum von $cutScale_a$, $cutScale_b$ ist der gesuchte Schnittmaßstab. Die Funktion aus Algorithmus 6 – Zeile 15 ist somit $calculateCutScale_3 = \max(cutScale_a, cutScale_b)$.

Somit kann für jeden Konfliktsektor $confSector_i$ der Schnittmaßstab $cutScale_i$ ermittelt werden. Der Schnittmaßstab zwischen den Beschriftungsräumen des punkthaften und des linienhaften Objekts ist somit der maximale der Sektoren, Algorithmus 6 – Zeile 16.

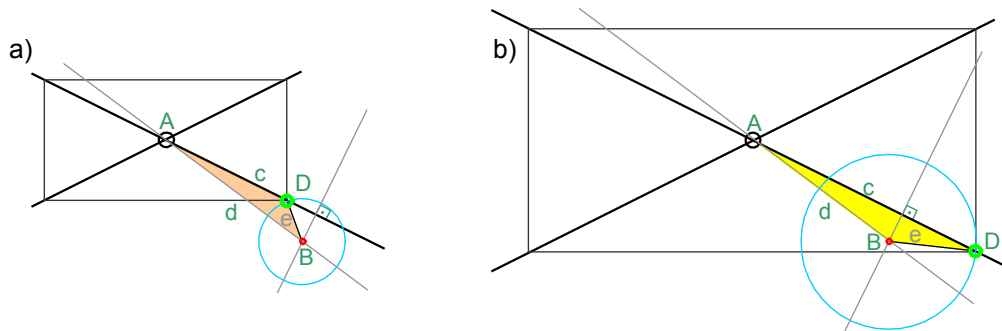


Abbildung 58 Zwei mögliche Lösungen (Algorithmus 6 – Zeile 15 bzw. (4.59)), da es zwei Schnittpunkte (D – hellgrün dargestellt) zwischen dem hellblauen Kreis, dem Beschriftungsraum bzw. Puffer des linienhaften Objekts, der Diagonale des Beschriftungsraums des punkthaften Objekts und seiner Grenze gibt. (beide Abbildungen sind im gleichen Maßstab)

4.3.7.2 Deselektionsmaßstab

Die Motivation und Vorgehensweise für die Bestimmung des Deselektionsmaßstabs von Konflikten zwischen punkthaften und linienhaften Objekten ist unverändert zu der zwischen punkthaften Objekten in Abschnitt 4.3.2.2. Der Deselektionsmaßstab ist abhängig von der Beschriftungsschwierigkeit der beteiligten Objekte. Die Bestimmung der Beschriftungsschwierigkeit punkthafter Objekte bei Konflikten mit linienhaften Objekten verlangt eine Adaption, die im nächsten Abschnitt vorgestellt wird. Der umgekehrte Fall, die Berücksichtigung von Konflikten mit punkthaften Objekten aus Sicht der linienhaften Objekte, ist, wie in den Beispielen in Abbildung 51 gezeigt, durch Abschnitt 4.3.6 abgedeckt.

Durch die Berücksichtigung von Konflikten mit zu beschriftenden punkthaften Objekten ergibt sich eine obere Schranke der Laufzeit der Beschriftungsschwierigkeit zu beschriftender linienhafter Objekte von $O(l*s^2+p*s)$ ($l \equiv$ Anzahl der zu beschriftenden linienhaften Objekte; $p \equiv$ Anzahl der zu beschriftenden punkthaften Objekte und $s \equiv$ maximale Anzahl der (Linien-) Segmente zu beschriftender linienhafter Objekte).

Die Integration dieser Konflikte in den reaktiven Konfliktgraphen erfolgt zentral in Abschnitt 4.3.11.

4.3.8 Erweiterung Beschriftungsschwierigkeit: punkthafes Objekt

Für punkthafte Objekte wurde in 4.3.3.3 ein Verfahren zur Ermittlung der Deselektion und Beschriftungsschwierigkeit im Konflikt mit anderen punkthaften Objekten vorgestellt. Dieses Verfahren kann für Konflikte mit linienhaften Objekte adaptiert werden. Für jedes Gebiet des Beschriftungsraums des punkthaften Objekts (vgl. Abschnitt 4.3.3.3 und Abbildung 39) wird die Konfliktfläche durch ein achsenparalleles Rechteck approximiert, wie in Abbildung 59 dargestellt. Diese approximierten Konfliktflächen können, wie die Konfliktflächen von anderen punkthaften Konflikten, mit dem Verfahren aus Abschnitt 4.3.3.3 behandelt werden.

Die Laufzeit der Beschriftungsschwierigkeit zu beschriftender punkthafter Objekte kann bei der Berücksichtigung von Konflikten mit zu beschriftenden linienhaften Objekten mit $O(p*g+l*s*g)$ nach oben abgeschätzt werden ($p \equiv$ Anzahl der zu beschriftenden punkthaften Objekte; $l \equiv$ Anzahl der zu beschriftenden linienhaften Objekte; $g \equiv$ maximale Anzahl der Gebiete eines zu beschriftenden punkthaften Objekts und $s \equiv$ maximale Anzahl der (Linien-)Segmente zu beschriftender linienhafter Objekte). Jedes linienhafte Objekt kann s Konflikte mit den g Gebieten des betrachteten zu beschriftenden punkthaften Objekt haben.

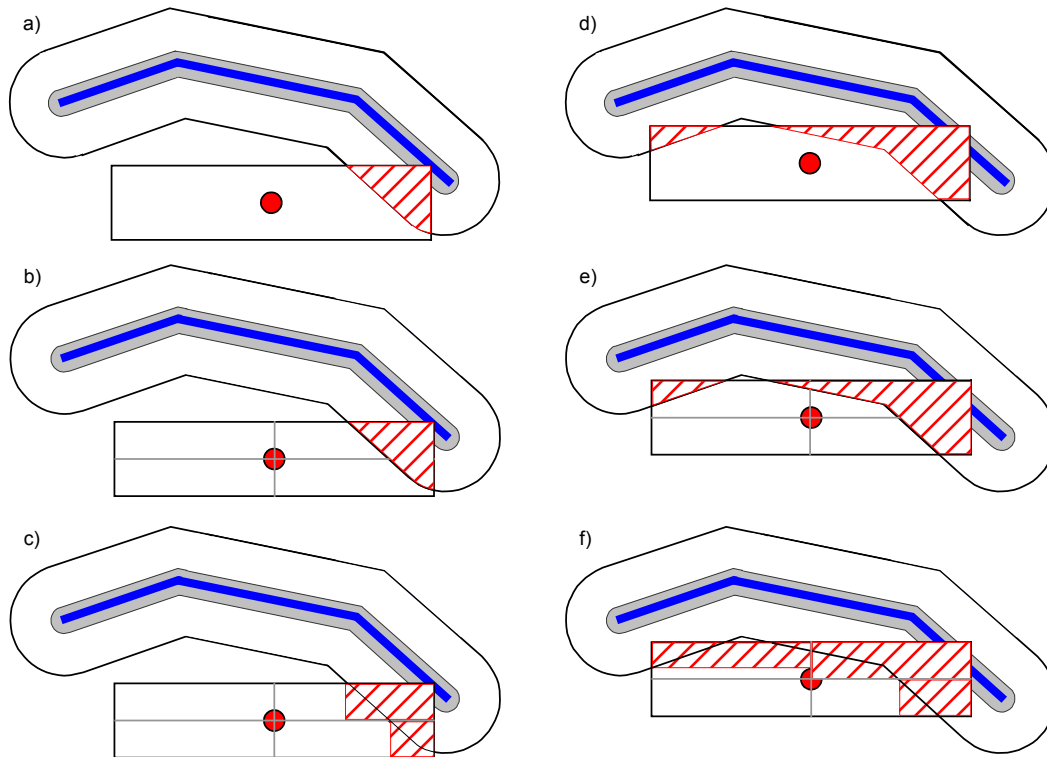


Abbildung 59 Rückführung des Konflikts und damit des Deselektionskriteriums von einem punkthaften zu beschriftenden Objekt mit einem linienhaften (a) und d)) auf den mit einem punkthaften durch Approximation der Konfliktschatten. Hier für den einfachen Fall dargestellt, der allgemeine Fall (Abschnitt 4.3.3.3.2) ergibt sich analog.

4.3.9 Flächenhafte Objekte

Die Integration zu beschriftender flächenhafter Objekte in den (reaktiven) Konfliktgraphen ist schwierig, da eine Einschränkung des Beschriftungsraums nicht möglich ist. Der Beschriftungsraum eines flächenhaften Objekts ist die Fläche selbst, dieser ist somit unabhängig vom Maßstab. Damit steht das zu beschriftende flächenhafte Objekt mit allen zu beschriftenden Objekten, die in der Fläche liegen oder diese schneiden, in Konflikt. Anschaulich kann man sich eine Karte Deutschlands vorstellen, in der alle größeren Städte eingetragen und beschriftet sind und in die man nun die Bundesländer mit Beschriftungen hinzunimmt (Abbildung 60). Die explosive Zunahme der Anzahl potentieller Konflikte ist offensichtlich. Neben diesem Problem, das (schon) zu einem festen Maßstab auftritt, kommen bei dem reaktiven Konfliktgraphen die Maßstabsübergänge und die dazu nötigen Formalisierungen hinzu.

Um eine angemessene kartographische Beschriftung zu erhalten und gleichzeitig eine Integration in den reaktiven Konfliktgraphen, der zu einer effizienten Beschriftung führt, muss der Beschriftungsraum eingegrenzt werden. Die flächenhafte Beschriftung lässt sich in zwei Klassen unterteilen, abgesehen von Siedlungsflächen (s. Abschnitt 4.3.10).

Der Unterschied der beiden flächenhaften Beschriftungsklassen liegt in der Ausrichtung der Schrift. Zum einen kann die Schrift achsenparallel platziert werden (Abbildung 60 und Abbildung 61 b)) oder der Schriftverlauf kann die Ausdehnung der Fläche widerspiegeln, wie in Abbildung 61 a) dargestellt.



Abbildung 60 Zunahme der Anzahl potentieller Konflikte bei der Beschriftung flächenhafter Objekte am Beispiel der Bundesländer der Bundesrepublik Deutschland. Die Schrift muss innerhalb des zugehörigen Bundeslandes platziert werden und verursacht dabei potentielle Konflikte mit allen Objekten innerhalb des Bundeslandes. (Karte [WES96])



Abbildung 61 Die flächenhafte Beschriftung, die die Ausdehnung der Fläche widerspiegelt, ist in a) dargestellt. b) zeigt eine achsenparallele Beschriftung von Flächen. (Karte a) [DIE76] und b) [NAT95])

Die Bestimmung der für diese beiden Gattungen nötigen Standlinien kann maßstabsunabhängig erfolgen und ist nicht auf eine Standlinie beschränkt. Für den Fall, wo der Schriftverlauf die Ausdehnung der Fläche wiedergibt, kann auf zwei Verfahren zurückgegriffen werden. Beide basieren auf der Ermittlung des Skeletts, in der Literatur oft mit Medial Axis bezeichnet, das anschaulich die Fläche auf einen (nicht notwendiger Weise einfachen) Linienzug projiziert. Das Verfahren von Freeman ermittelt das Skelett mit Hilfe von „Wellenlinien / Wellenfronten“, die vom Flächenrand ins Innere laufen [AHF84a] (Abbildung 62 a)). Alternativ kann das Skelett mit Hilfe von Voronoi-Diagrammen abgeleitet werden. Dieses Verfahren wird in [PET96] und [PEP97] detailliert behandelt, so dass hier nicht weiter darauf eingegangen wird (Abbildung 62 b)).

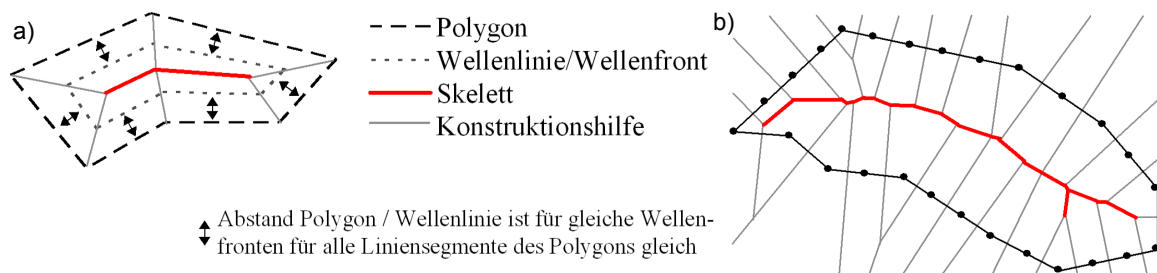


Abbildung 62 a) Erzeugung des Skeletts einer Fläche mit Hilfe von Wellenlinien/Wellenfronten nach [AHF84a]; b) alternative Erzeugung des Skeletts mit Hilfe des Voronoi-Diagramms nach [PET96].

Die Ermittlung der Standlinien für die achsenparallele Beschriftung ist weniger aufwendig und kann [ELL01] entnommen werden.

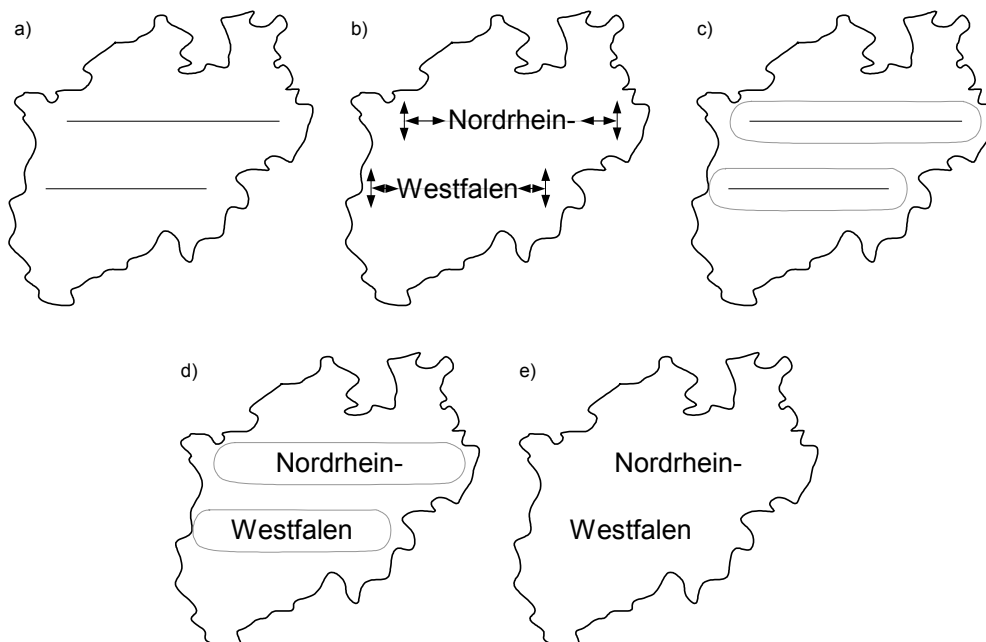


Abbildung 63 Die achsenparallele Beschriftung flächenhafter Objekte am Beispiel Nordrhein-Westfalens. Der Beschriftungsraum wird dabei auf den linienhafter Objekte zurückgeführt wie in a) bis e) illustriert.

Mit den vorgestellten Verfahren werden für einen Maßstab Standlinien bestimmt. Die Standlinien und der Maßstab, zu dem sie bestimmt wurden, werden wie die Referenzbeschriftung und der Referenzmaßstab am Objekt gespeichert, um eine Maßstabsunabhängigkeit zu

erzielen. Durch Skalierung mit dem (Ziel-)Maßstab können so zu jedem Maßstab die Standlinien abgeleitet werden, ohne sie aufwendig neu zu berechnen. Die so bestimmten Standlinien werden wie zu beschriftende linienhafte Objekte beschriftet (Abbildung 63). Im Unterschied zu den bisher besprochenen linienhaften Objekten können diese aus mehreren separaten Teilen zusammengesetzt sein; eine besondere Behandlung ist jedoch nicht erforderlich.

Durch diese Vorgehensweise ist der Beschriftungsraum eingeschränkt und die Beschriftung flächenhafter Objekte auf die von linienhaften zurückgeführt.

4.3.10 Beschriftung von Siedlungsflächen

Die in den Abschnitten 4.3.1 und 4.3.3.3 entwickelten Verfahren zur Beschriftung punkthafter Objekte werden in diesem Abschnitt für die Beschriftung von Siedlungsflächen erweitert. Siedlungsflächen bestehen aus einer oder mehreren Teilflächen, zu denen ein gemeinsamer Schriftzug gehört und die wegen der Ausdehnung des flächenhaften Kartenzeichens nicht als punkthafte Objekte beschriftet werden können. Die Ausrichtung der Schrift ist achsenparallel und kann auf oder neben dem Kartenzeichen erfolgen.

Da die Zuordnung der Schrift zu ihrem Objekt bzw. Objekten eindeutig sein soll, kann der Beschriftungsraum, wie in Abbildung 64 b) dargestellt, eingeschränkt werden. Analog zum Beschriftungsraum punkthafter Objekte kann dieser für Siedlungsflächen als Rechteck definiert werden, der sich aus dem als Rechteck approximierten flächenhaften Kartenzeichens ergibt, vergrößert nach oben und unten um jeweils die Höhe und nach links und rechts um jeweils die Breite der zu platzierenden Labelbox. Dieser Beschriftungsraum wird analog zu den Gebieten für punkthafte Objekte Abschnitt 4.3.3.3 Abbildung 39 in Gebiete zerlegt, die maximal die Größe der zu platzierenden Labelbox haben (Abbildung 64 b)). Durch diese Vorgehensweise können Schnittmaßstäbe zu Konfliktpartnern mit den Verfahren für punkthafte ohne weitere Adaption genutzt werden. Die Bestimmung der Beschriftungsschwierigkeit wird am Ende diesen Abschnitts angesprochen.

Hat die Siedlungsfläche eine L-förmigen Geometrie, ist die Approximation durch ein Rechteck ungünstig, da die Distanz zwischen dem Kartenzeichen und der möglichen Beschriftungsposition in der oberen rechten Ecke des Beschriftungsraums zu groß ist. In diesen Fällen kann der Beschriftungsraum individuell dem flächenhaften Kartenzeichen angepasst werden, der auch in Gebiete, die maximal die Größe der zu platzierenden Labelbox haben, zerlegt wird. Da die Geometrie des Beschriftungsraums nicht mehr einem Rechteck entspricht, können die Verfahren zur Bestimmung der Schnittmaßstäben punkthafter Objekte nicht ohne Adaption verwendet. Der Beschriftungsraum muss in Rechtecke zerlegt werden so dass die entwickelten Verfahren für punkthafte Objekte verwendet werden können.

Im Gegensatz zur punkthafter Beschriftung, wo die Schrift nur einen horizontalen Bewegungsraum über, einen unter sowie vertikalen links und rechts von der Signatur besitzt, kann bei Siedlungsflächen die Schrift überall im Beschriftungsraum platziert werden. Das führt zu einer Änderung des Verfahrens aus Abschnitt 4.3.3.3, dass sowohl die Beschriftungsschwierigkeit bestimmt, als auch zur Platzierung des Schriftobjekts verwendet wird. Im Unterschied zu der Gebietsaufteilung in Abbildung 40 wird bei Siedlungsflächen der gesamte Beschriftungsraum in Gebiete, die maximal die Größe der zu platzierende Schrift besitzen, aufgeteilt, wie in Abbildung 64 b) dargestellt.

Um konfliktfreie Bewegungsräume aufzufinden besitzen alle Gebiete, die nicht an das Äußere grenzen, jeweils vier \bar{x} und \bar{y} -Vektorenpaare, deren Ursprung in den vier Ecken des Gebiets liegen. Die Gebiete am Rand des Beschriftungsraums besitzen \bar{x} und \bar{y} -Vektoren analog zu dem Verfahren der freien Beschriftungsfläche, das in Abschnitt 4.3.3.3.2 für die Beschriftung

punkthafter Objekte entwickelt wurde (Abbildung 40). Die Bestimmung konfliktfreier Bereiche und ihrer Bewertung erfolgt analog zu Algorithmus 3 und Algorithmus 4. Wegen der engen Analogie wird an dieser Stelle auf eine detaillierte Darstellung verzichtet (s. Abbildung 64).

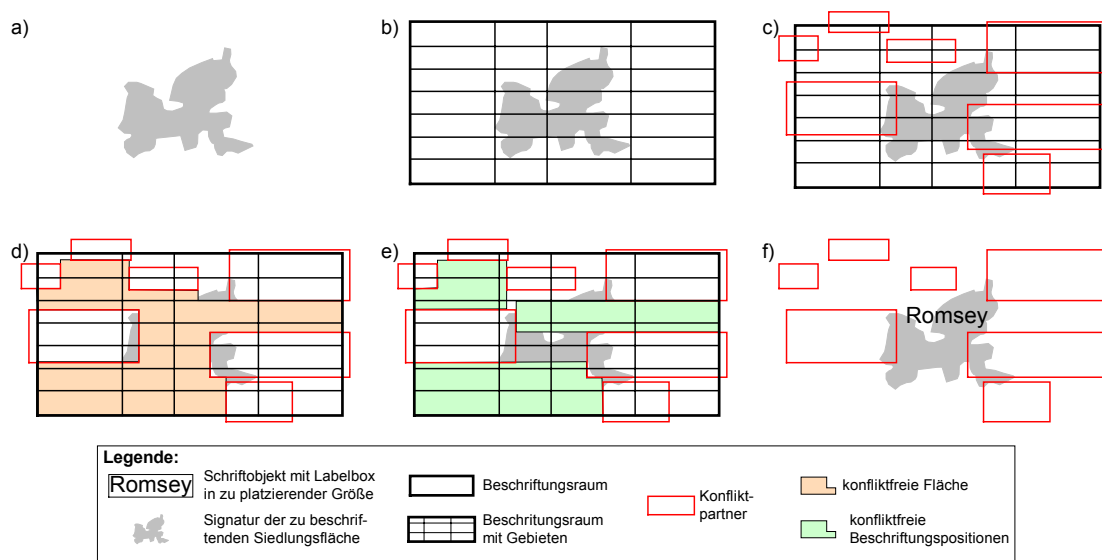


Abbildung 64 Siedlungsflächen: die zu beschriftende Siedlungsfläche ist in a) dargestellt und in b) der Beschriftungsraum und die Zerlegung in Gebiete; die Platzierung der Schrift unter Berücksichtigung der Konflikte aus c) ist in d) bis f) illustriert.

4.3.11 Erzeugung des reaktiven Konfliktgraphen

Ziel dieses Abschnitts ist die Entwicklung eines Verfahrens zur Generierung eines homogenen reaktiven Konfliktgraphen, der die unterschiedlichen zu beschriftenden Objekte und die zwischen ihnen möglichen Konflikte berücksichtigt. Wie in den Abschnitten zur Ermittlung der Deselektionsmaßstäbe der Beschriftungskonflikte (Abschnitte 4.3.2.2, 4.3.5.2 und 4.3.7.2) erwähnt, basieren diese auf der Beschriftungsschwierigkeit bzw. Deselektion der beteiligten Objekte. Die Beschriftungsschwierigkeit bzw. Deselektionskriterien der Objekte sind in den Abschnitten 4.3.3.3, 4.3.6 und 4.3.8 beschrieben und sind die Grundbausteine für das Verfahren zur Erzeugung des reaktiven Konfliktgraphen.

Der reaktive Konfliktgraph ist in Abschnitt 4.2 definiert. Er besteht aus Knoten und Kanten, die beide attribuiert sind. Zu beschriftende Objekte werden als Knoten im reaktiven Konfliktgraphen, wie in den vorherigen Abschnitten bereits angesprochen, repräsentiert. Ein Attribut des Knotens speichert die Beschriftungsklasse (punkt-, linien- oder flächenhaft) und ein weiteres den Deselektionsmaßstab (*deselect*), der während der Generierung des reaktiven Konfliktgraphen erzeugt wird. Ferner übernimmt jeder Knoten die Informationen seines assoziierten Objekts. Die während der Erzeugung des Konfliktgraphen ermittelten Konflikte zu anderen Knoten, die in Konfliktgraphkanten abgelegt sind, werden als weitere Attribute an dem Objekt gespeichert.

Die Konfliktgraphkante repräsentiert Konflikte zwischen zu beschriftenden Objekten, die durch Endknoten der Kante repräsentiert sind. Die Erzeugung der Kanten erfolgt während der Generierung des Konfliktgraphen. Zu jeder Kante sind die zugehörigen Knoten, also die Konfliktpartner, assoziiert. Die Konfliktklasse, die von den beteiligten Konfliktpartnern abhängig ist (punkthaft-punkthaft, punkthaft-linienhaft, punkthaft-flächenhaft, linienhaft-linienhaft, linienhaft-flächenhaft und flächenhaft-flächenhaft) wird in einem eigenen Attribut

abgelegt. Die *upperBound*, der Schnittmaßstab zwischen den beiden Konfliktpartnern, und die *lowerBound*, der Maßstab unterhalb dem der Konflikt nicht mehr auftritt, sind weitere Attribute der Kante.

Abschnitt 4.3.2.2 und die Einleitung von Abschnitt 4.3.3 motivieren, diskutieren und illustrieren die Zusammenhänge. Aufbauend auf Algorithmus 1 aus Abschnitt 4.3.3.1, einem ersten Ansatz zur Erzeugung des reaktiven Konfliktgraphen für punkthafte Objekte mit dem Deselektionskriterium maximaler Knotengrad zu Objekten höherer Priorität, entsteht ein allgemeineres Verfahren für die Generierung des reaktiven Konfliktgraphen, das in Algorithmus 7 dargestellt ist.

```

1  Input: zu beschriftende Objekte mit Lage, Referenzbeschriftung und Priorität
        sowie maxDifficulty für maximale Beschriftungsschwierigkeit zu
        wichtigeren Objekten
   Output: reaktiver Konfliktgraph (mit Knoten und Kanten)
5  // erzeuge Konfliktgraphkanten zu Objekten mit höherer Priorität und bestimme
   // den Deselektionsmaßstab für den betrachteten Konfliktgraphknoten
   für jedes zu beschriftende Objekt aufsteigend nach Priorität (priorityObject):
        erzeuge Konfliktgraphknoten und Beschriftungsraum zu priorityObject
        berechne Schnittmaßstab zu jedem wichtigeren Objekt (höhere Priorität)
10  sortiere diese Objekte absteigend nach Schnittmaßstab
   durchlaufe Schnittmaßstab-Liste (cutObject):
        bestimme Beschriftungsschwierigkeit zwischen priorityObject und zu cutObject und
        allen vorherigen Elementen aus Schnittmaßstab-Liste
        wenn ermittelte Beschriftungsschwierigkeit größer maxDifficulty:
15         dann erzeuge Konfliktgraphkante:
                zwischen priorityObject und cutObject
                setze upperBound auf Schnittmaßstab zwischen priorityObject und cutObject
                sonst verlasse Schleife
        setze Knoten-Attribut von priorityObject (Deselektionsmaßstab):
20         wenn cutObject existiert (zu dem gerade keine Kante mehr erzeugt wurde)
                dann setze deselect auf Schnittmaßstab von cutObject +  $\epsilon^1$ 
                sonst setze deselect auf 0
        // behandle Sonderfall: Konfliktgraphkanten zwischen Objekten gleicher Priorität
        für jedes Paar an zu beschriftenden Objekten mit gleicher Priorität:
25         berechne zu jedem Paar den Schnittmaßstab
        erzeuge Konfliktgraphkante zu jedem Paar, wenn der Schnittmaßstab größer ist als
        die beiden deselect-Maßstäbe der Objekte
        und setze upperBound auf den ermittelten Schnittmaßstab
        // bestimme die fehlenden Attribute der Konfliktgraphkante und
30 // teste die Konfliktgraphkante bzw. Maßstabsintervall auf (Ir-)Relevanz
   für jede Konfliktgraphkante:
        setze Attribut lowerBound der Kante auf den größeren deselect-Maßstab
        der beiden inzidenten Knoten (Objekte)
        eliminiere Kante deren Attribut upperBound kleiner ist als das Attribut
35         lowerBound

```

Algorithmus 7 Erzeugung des reaktiven Konfliktgraphen: Ermittlung der Konflikte und Generierung der Konfliktgraphkanten mit Beschriftungsschwierigkeit und Priorität als Deselektionskriterium. Neben der Integration der entwickelten Verfahren werden alle Klassen an zu beschriftenden Objekte berücksichtigt (im Unterschied zu Algorithmus 1).

Das Verfahren zur Generierung des reaktiven Konfliktgraphen ist dreigeteilt durch die *for*-Schleifen (Zeile 1 bis 22, Zeile 23 bis 28 und Zeile 29 bis 35), wie in Algorithmus 7 dargestellt und teilweise in Abschnitt 4.3.3.1 beschrieben.

- Algorithmus 7 – Zeile 1 bis 10

Alle zu beschriftenden Objekte werden nach ihrer Priorität abgearbeitet Algorithmus 7 – Zeile 7. Die Objekte mit der niedrigsten Priorität zuerst und am Ende die mit der höchsten Priorität. Für jedes zu beschriftende Objekt – *priorityObject* ist das zurzeit aktive – wird zunächst der

zugehörige reaktive Konfliktgraphknoten erzeugt (Zeile 8). Jeder Knoten besitzt eine Referenz zum zugehörigen Objekt.

Die Schnittmaßstäbe zu allen Objekten mit einer höheren Priorität werden in Zeile 9 berechnet. Dabei wird in Abhängigkeit von beiden Konfliktpartnern das entsprechende Verfahren gewählt. Der Schnittmaßstab zwischen zwei punkthaften Objekten ist das Minimum der Ergebnisse der Gleichungen (4.14) und (4.15) (Abschnitt 4.3.2.1). Für den Schnittmaßstab zwischen zwei linienhaften Objekten werden zunächst die beiden Punkte auf den Objekten bestimmt, die die kleinste Distanz zueinander haben. Gleichung (4.41) liefert dann den Schnittmaßstab zwischen den beiden linienhaften Objekten (Abschnitt 4.3.5.1). Der Schnittmaßstab zwischen punkthaften und linienhaften Objekt wird mit Algorithmus 6 bestimmt (Abschnitt 4.3.7.1). Da zu beschriftende flächenhafte Objekte auf linienhafte bzw. Siedlungsflächen auf punkthafte Objekte zurückgeführt werden, braucht dieser Fall nicht extra betrachtet werden.

Die Sortierung der so gewonnenen Objektliste erfolgt absteigend danach (Zeile 10). Dadurch werden die Objekte der Liste implizit nach ihrer Distanz (aus Konfliktsicht) zum aktuell betrachteten Objekt *priorityObject* geordnet.

- Algorithmus 7 – Zeile 11 bis 18

Die so erzeugte, nach dem Auftreten der Konflikte sortierte Liste der Objekte mit höherer Priorität wird im Folgenden durchlaufen (Zeile 11) – das jeweils betrachtete Objekt dieser Liste ist *cutObject*. Die Zeilen 12 bis 18 befinden sich somit in einer geschachtelten Doppelschleife. Das aktuell betrachtete Objekt der äußeren Schleife ist *priorityObject* und das der inneren Schleife *cutObject*.

Für das aktuelle Objekt der äußeren Schleife *priorityObject* wird die Beschriftungsschwierigkeit zu *cutObject* bestimmt (Zeile 12). Dabei wird als Kriterium „freie Beschriftungsfläche“ verwendet, das für alle Klassen an zu beschriftenden Objekten entwickelt wurde. Das verwendete Kriterium der Beschriftungsschwierigkeit ist zwar eindeutig – die Reihenfolge in der die Konfliktpartner betrachtet werden ist beliebig aber nicht additiv. Die einzelnen Beschriftungsschwierigkeiten zu zwei unterschiedlichen Objekten sind nicht zwangsläufig gleich der Beschriftungsschwierigkeit zu beiden Objekten gemeinsam (vgl. Fußnote⁸). Daher wird die Beschriftungsschwierigkeit von Objekt *priorityObject* zu *cutObject* sowie allen Objekten bestimmt, die in der Liste, die nach dem Schnittmaßstab sortiert ist (Zeile 9 und 10), vor dem Element *cutObject* liegen.

Wie schon beschrieben, ist die Sortierung der Liste nach dem Schnittmaßstab äquivalent zur Reihenfolge, in der die Konflikte bei einer Verkleinerung des Maßstabs auftreten. Das bedeutet, dass, wenn der Konflikt zu *cutObject* bei dem Schnittmaßstab eintritt, die Konflikte zu allen Objekten, die in der Liste vor *cutObject* liegen, schon eingetreten sind. Gleichzeitig nimmt die Beschriftungsschwierigkeit beim Durchlaufen der Liste monoton zu.

Die Bestimmung der Beschriftungsschwierigkeit in Zeile 14, die mit der maximal zulässigen Beschriftungsschwierigkeit (*maxDifficulty*) verglichen wird, ist abhängig von der Beschriftungsklasse des aktuell betrachteten zu beschriftenden Objekts.

Für punkthafte Objekte wurde in Abschnitt 4.3.3.1 der Knotengrad und in Abschnitt 4.3.3.3 die freie Beschriftungsfläche als Kriterien für die Beschriftungsschwierigkeit entwickelt. Zusammenfassend wurde in Abschnitt 4.3.3.4 festgestellt, dass eine Kombination dieser beiden Kriterien sinnvoll ist, die in der Funktion *Beschriftungsschwierigkeit* im gleichen Abschnitt umgesetzt ist.

Für die Bestimmung der Beschriftungsschwierigkeit linienhafter Objekte können die entwickelten Funktionen *labelScore*, insbesondere *labelScore_{maxQuality}* und *labelScore_{sumLabelWidthQuality}*, die in Algorithmus 5 Abschnitt 4.3.6 entwickelt und diskutiert wurden, verwendet werden. Die Beschriftung flächenhafter Objekte wurde, wie in Abschnitt 4.3.9 dargestellt, auf die Beschriftung von linienhafter zurückgeführt. Analog gilt das für zu beschriftende Siedlungsflächen, die auf die Beschriftung punkthafter Objekte in Abschnitt 4.3.10 zurückgeführt ist.

Für die oben erwähnten Bewertungsfunktionen gilt: je kleiner der Wert, desto schwieriger ist es das Objekt zu beschriften.

Alternativ können andere Kriterien oder andere Kombinationen und Gewichtungen der Kriterien gewählt werden. Allgemein können die Zeilen 12 bis 14 durch „lasse die Objekte als Konfliktpartner zu, die das gewünschte Kriterium erfüllen“ ersetzt werden.

In Zeile 14 wird die ermittelte Beschriftungsschwierigkeit für das betrachtete zu beschriftende Objekt (*priorityObject*) und denen, die in der sortierten Schnittmaßstabsliste vor ihm liegen, zum Maßstab, der gleich dem Schnittmaßstab zum Objekt *cutObject* ist, mit dem Wert *maxDifficulty* verglichen. *MaxDifficulty* ist ein vorgegebener Parameter, der eine obere Schranke zu der Menge an Konflikten zu Objekten mit höherer Priorität ist.

Fällt das betrachtete Objekt der inneren Schleife *cutObject* in die Menge der zulässigen Konflikte, d.h. ist die Beschriftungsschwierigkeit zu diesem Objekt und allen, die vor diesem in der sortierten Schnittmaßstabsliste liegen, größer als der Parameter *maxDifficulty*, dann wird zwischen diesem Objekt und *priorityObject* eine Kante des reaktiven Konfliktgraphen erzeugt (Zeile 15).

Das Attribut *upperBound* der neu erzeugten reaktiven Konfliktgraphkante wird auf den ermittelten Schnittmaßstab zwischen *priorityObject* und *cutObject* gesetzt (Zeile 17).

Ist es schwieriger das Objekt zu beschriften, als der Schwellenwert des Parameters *maxDifficulty* zulässt, dann wird keine Konfliktgraphkante erzeugt und die innere Schleife abgebrochen (Zeile 18). Wegen der monotonen Zunahme der Beschriftungsschwierigkeit beim Durchlaufen der inneren Schleife bzw. der sortierten Schnittmaßstabsliste und der Zunahme des Schnittmaßstabs können keine weiteren Konfliktkanten hinzukommen.

- Algorithmus 7 – Zeile 19 bis 22

Abschließend wird zu dem in Zeile 8 erzeugten Konfliktgraphknoten, der das Objekt *priorityObject* repräsentiert, das Attribut *deselect*, der Maßstab, ab dem das zu beschriftende Objekt nicht mehr beschriftet wird, gesetzt (Zeile 19). Dieser Maßstab muss so gewählt werden, dass *priorityObject* beim letzten zulässigen Konfliktpartner noch beschriftet wird aber nicht mehr bei Konfliktpartnern, deren Schnittmaßstab kleiner ist. Daher wird der *deselect*-Maßstab von *priorityObject* auf den Schnittmaßstab mit dem ersten nicht mehr zulässigen Konfliktpartner (*cutObject*) gesetzt und dieser um einen kleinen Betrag (ϵ -Wert) erhöht, damit *priorityObject* bei diesem Maßstab nicht mehr beschriftet wird. Wenn die Schnittmaßstabsliste zuvor bis zum Ende durchlaufen wurde, d.h. es gibt keine Konflikte, die ausgeschlossen werden, da die Beschriftungsschwierigkeitsschwelle *maxDifficulty* nicht überschritten wird, ist der *deselect*-Maßstab Null. Das ist äquivalent zu der Aussage, dass das Objekt immer beschriftet wird.

- Algorithmus 7 – Zeile 23 bis 28

In diesem Abschnitt des Algorithmus werden die Konflikte zwischen Objekten gleicher Priorität behandelt, die in den Zeilen 1 bis 22 nicht berücksichtigt wurden. Angenommen Zeile 9 schließt Objekte gleicher Priorität ein, so würden Konflikte zwischen Objekten gleicher Priorität durch zwei Konfliktkanten repräsentiert und bei der Beschriftungsschwierigkeit berücksichtigt.

Nach Ermittlung der Objekte gleicher Priorität werden diese paarweise¹⁶ durchlaufen (Zeile 24). Wenn der Schnittmaßstab zwischen den beiden betrachteten Objekten größer ist als die beiden Deselektionsmaßstäbe der Objekte (*deselect*), dann wird eine Konfliktgraphkante zwischen ihnen erzeugt und das Attribut *upperBound* auf den Schnittmaßstab gesetzt.

- Algorithmus 7 – Zeile 29 bis 35

In diesem Abschnitt werden alle erzeugten Konfliktgraphkanten durchlaufen (Zeile 31) und deren Attribut *lowerBound* gesetzt. *LowerBound* ist der größere der beiden Maßstäbe, unterhalb dem einer der Konfliktpartner (einer der inzidenten Knoten) nicht mehr beschriftet wird und damit der Konflikt nicht mehr existent ist (Abbildung 26) (Zeile 32/33). Anschließend wird anhand der Attribute *upperBound* und *lowerBound* überprüft, ob es sich um eine irrelevante Kante handelt oder nicht. Bei irrelevanten Kanten ist die *upperBound* kleiner als die *lowerBound*, die ansonsten ein gültiges Intervall bilden (Abbildung 26). Irrelevante Kanten werden gelöscht (Zeile 34/35).

4.3.11.1 Laufzeitanalyse

Für die Laufzeitanalyse wird Algorithmus 7 in drei Teile zerlegt und getrennt betrachtet: Zeile 1 bis 22; Zeile 23 bis 28 und Zeile 29 bis 35. ($n \equiv$ Anzahl der zu beschriftenden punkthaften Objekte; $l \equiv$ Anzahl der zu beschriftenden linienhaften Objekte; $g \equiv$ maximale Anzahl der Gebiete eines zu beschriftenden punkthaften Objekts und $s \equiv$ maximale Anzahl der (Linien-) Segmente zu beschriftender linienhafter Objekte)

Die Sortierung der Liste der zu beschriftenden Objekte für die Schleife in Zeile 7 kann mit $O(n \log n)$ nach oben abgeschätzt werden. Innerhalb der Schleife in Zeile 7 gelten folgende obere Schranken:

- Zeile 8 (Erzeugung der Konfliktgraphknoten): $O(l)$
- Zeile 9 (Schnittmaßstab): $O(n*s^2)$ Im Worst-Case ist das betrachtete Objekt linienhaft ebenso wie die Konfliktpartner, und alle bestehen aus s Liniensegmenten. Da der Schnittmaßstab zwischen zu beschriftenden linienhaften Objekte segmentweise bestimmt wird, ergibt sich $O(s^2)$. Die Berechnung des Schnittmaßstabs zwischen zwei punkthaften Objekten kostet $O(1)$. Zur Berechnung des Schnittmaßstabs zwischen einem punkthaften und einem linienhaften Objekt wird der Schnittmaßstab jedes Liniensegments mit dem eines Konfliktsektors bestimmt, d.h. es ergibt sich $O(s*4)=O(s)$.
- Zeile 10 (Sortierung): $O(n \log n)$
- Zeile 19 bis 22 (Attributierung Konfliktgraphknoten): $O(l)$

¹⁶ Jedes Paar wird einmal behandelt, d.h. es gilt $(a,b) = (b,c)$.

Innerhalb der zweiten Schleife (Zeile 11 bis 18) gilt:

- Zeile 12 und 13 (Beschriftungsschwierigkeit): $O(n*s^2+n*s*g)$ Diese Laufzeit für die Beschriftungsschwierigkeiten sind in den Abschnitten 4.3.3.3.2, 4.3.7.2 und 4.3.8 hergeleitet.
- Zeile 14 bis 18 (Erzeugung Konfliktgraphkante): $O(1)$

Für die innere Schleife (Zeile 11 bis 18) ergibt sich eine Abschätzung der Laufzeit nach oben von $O(n^2*s^2 + n^2*s*g)$.

Der erste Teil, die Schleife von Zeile 1 bis 22, hat somit eine obere Schranke von $O(n^2 \log n + n^3*s^2 + n^3*s*g)$.

Im Worst-Case kommt es in der Schleife, Zeile 24 bis 28, im zweiten Teil zu $O(n^2)$ Paaren an Objekten mit gleicher Beschriftungspriorität. Die Berechnung des Schnittmaßstabs kostet im Worst-Case $O(n*s^2)$ (s. Zeile 9) und die Erzeugung der Konfliktgraphkanten $O(1)$. Damit ergibt sich für den zweiten Teil, Zeile 23 bis 28, eine obere Schranke der Laufzeit von $O(n^3*s^2)$.

Die Laufzeit des dritten Teils, Zeile 29 bis 35, ist abhängig von der Anzahl der Kanten des reaktiven Konfliktgraphen, die mit $O(n^2)$ nach oben abgeschätzt werden kann (vgl. Zeile 15 mit den Schleifen Zeile 7 und Zeile 11).

Die obere Schranke der Laufzeit für die Generierung des reaktiven Konfliktgraphen ist somit $O(n^2 \log n + n^3*s^2 + n^3*s*g)$. Unter der Annahme, dass die Anzahl der Liniensegmente und die Anzahl der Gebiete konstant ist und die Laufzeit in Abhängigkeit der Anzahl der zu beschriftenden Objekte (n) relevant ist, kann die obere Schranke mit $O(n^3)$ abgeschätzt werden.

4.4 Speicherung des reaktiven Konfliktgraphen

Für das vorgestellte Konzept zur Beschriftung von Bildschirmkarten in Echtzeit ist neben der Struktur des Konfliktgraphen seine Speicherung, insbesondere der effiziente Zugriff, entscheidend. Ohne Kapitel 5 und den Anforderungen der Interaktionsphase etwas vorwegzunehmen, wird im Folgenden kurz die Speicherung des reaktiven Konfliktgraphen angerissen.

Um die Anfrage nach Kartenausschnitten zu unterstützen, bieten sich geometrische Datenstrukturen (s. Abschnitt 2.2.2.2) an. Der reaktive Konfliktgraph besteht aus Knoten und Kanten. Im Unterschied zu „normalen“ Graphen besitzen seine Knoten eine geometrische Einbettung durch das Objekt, das sie repräsentieren. Dabei ist zu berücksichtigen, dass die Knoten an sich, die intuitiv mit punkthaften Objekten assoziiert werden, weder Geometrie noch eine Lage im Raum besitzen. Sie repräsentieren unterschiedliche geometrische (Karten-) Objekte, wie z.B. Flüsse, Orte, Länder etc. Damit eine Anfrage nach einem Kartenausschnitt alle Objekte erfasst, ist der Schlüssel die Geometrie des Objekts, die der Knoten repräsentiert im Maßstab 1:1, d.h. in Realwelt-Koordinaten. Um aus den Schlüsseln wieder den Graphen zu gewinnen, erhält jedes Objekt einen Zeiger auf den ihn repräsentierenden Knoten im Graphen. Es gibt somit eine 1:1 Verknüpfung zwischen Objekt und reaktiven Konfliktgraphknoten. Geometrische Datenstrukturen, die auf punkthafte Objekte beschränkt sind, wie der k-d-Baum, sind daher ungeeignet.

Jeder reaktive Konfliktgraphknoten lässt sich als ein Volumen mit fester Lage im dreidimensionalen Raum beschreiben. Neben der flächenhaften Ausdehnung der Signatur – Knoten repräsentieren Kartenobjekte und diese repräsentieren Realweltobjekte, die eine feste Lage im Raum haben – besitzen Knoten durch das Attribut *deselect* eine dritte Dimension. *Deselect* repräsentiert den Maßstab, unterhalb dem das Objekt nicht mehr beschriftet bzw. dargestellt wird (Deselektionsmaßstab). Anschaulich bildet die (flächenhafte) Geometrie des

zum Knoten gehörenden Objekts bzw. die Signatur die „Bodenfläche“ des Volumens und die Höhe ergibt sich aus dem Darstellungsbereich des Knotens. Der Darstellungsbereich geht vom Deselektionsmaßstab (*deselect*) bis zum Maßstab 1:1. Siehe dazu auch Abschnitt 5.1 mit Abbildung 68.

Als geeignete geometrische Datenstruktur, die nicht auf Punkte beschränkt ist und dreidimensionale Objekte verwalten kann sowie für die Interaktionsphase effiziente Zugriffsmethoden bereitstellt, bietet sich der R-Baum oder einer seiner „Verwandten“ (R^+ -Baum, R^* -Baum etc.) an (vgl. Diskussion in Abschnitt 2.2.2.2 bzw. [SAM90], [OTW93], [BKO97]).

Die Speicherung der Knoten des reaktiven Konfliktgraphen kann mit $O(n \log n)$ ($n \equiv$ Anzahl der reaktiven Konfliktgraphknoten) nach oben abgeschätzt werden (Abschnitt 2.2.2.2.3). Der für die Speicherung nötige Platz beträgt $O(n)$ [SAM90].

Für die Speicherung der reaktiven Konfliktgraphkanten in einer separaten Datenstruktur bieten sich zwei Möglichkeiten an. In Anlehnung an die Speicherung der Knoten, können alle Kanten in einem 3D-R-Baum abgelegt werden. Dazu erhalten die Kanten als Geometrie die Fläche zwischen den Geometrien der beiden inzidenten Objekte (Abbildung 65). Zusammen mit dem zum Knoten gehörigen Maßstabsintervall (*lowerBound* und *upperBound*) ergibt sich die Höhe und Lage des geometrischen Objekts im dreidimensionalen Raum. Die Unter- und Oberseite von diesem liegen somit konstruktiv in parallelen Ebenen zur x- und y-Achse (Voraussetzung für die Anfragen zur Ableitung des statischen Konfliktgraphen während der Interaktionsphase). Die Speicherung aller reaktiven Konfliktgraphkanten kann mit $O(m \log m)$ und der Speicherplatz mit $O(m)$ nach oben abgeschätzt werden ($m \equiv$ Anzahl der reaktiven Konfliktgraphkanten). Die Abschätzung der Anzahl der reaktiven Konfliktgraphkanten m ist für den Worst-Case $O(n^2)$ und somit $O(n^2 \log n^2)$ bzw. $O(n^2 \log n)$ für die Speicherung und entsprechend $O(n^2)$ für den Speicherplatz.

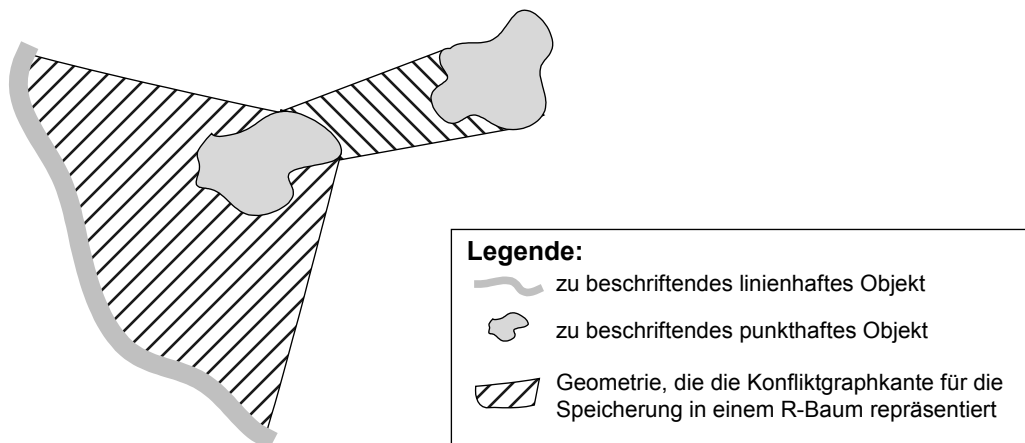


Abbildung 65 Geometrien der zu beschriftenden punkt- und linienhaften Objekte und deren flächenhafte Konflikte zur Speicherung in einer geometrischen Datenstruktur.

Alternativ ist die Speicherung der reaktiven Konfliktgraphkanten an inzidenten Knoten (d.h. eine lokale Speicherung) möglich. Dazu wird jede Kante nur an einem ihrer inzidenten Knoten gespeichert. Das ist ausreichend, da beide Knoten in dem während der Interaktionsphase abgeleiteten statischen Konfliktgraphen liegen müssen, damit der zugehörige Konflikt eintritt. Durch diese Vorgehensweise entfällt die Filterung doppelter Einträge nach der Anfrage. Für einen effizienten Zugriff während der Interaktionsphase bieten sich als geometrische Speicherstrukturen Segment- und Intervall-Bäume (Abschnitt 2.2.2.2

bzw. [SAM90], [OTW93], [BKO97]) an. Diese unterstützen das Aufspieß-Problem [OTW93], das in diesem Fall wie folgt aussieht: liefere alle Kanten, deren zugehörige Maßstabsintervalle den Anfragemastab abdecken. Als Index der reaktiven Konfliktgraphkanten wird ihr zugehöriges Maßstabsintervall verwendet. Die obere Schranke für die Speicherung der reaktiven Konfliktgraphkanten in einem Intervall- oder Segment-Baum ist $O(p \log p)$ ($p \equiv$ Anzahl der inzidenten reaktiven Konfliktgraphkanten des betrachteten reaktiven Konfliktgraphknoten) [BKO97], [SAM90]. Der Speicheraufwand für Intervall-Bäume kann nach oben mit $O(p)$ [BKO97], [SAM90] abgeschätzt werden. Der von Segment-Bäumen liegt mit $O(p \log p)$ [BKO97], [SAM90] höher. Die Anzahl der reaktiven Konfliktgraphkanten an einem Knoten p ist im Worst-Case $O(n)$. Das führt zu einer oberen Schranke bei der Speicherung pro zu beschriftenden Objekt in Segment- oder Intervallbäumen von $O(n \log n)$ bei einem Speicherplatzbedarf von $O(n)$ bei Intervall-Bäumen und $O(n \log n)$ bei Segment-Bäumen. Betrachtet man alle zu beschriftenden Objekte des reaktiven Konfliktgraphen, ergibt sich als Zeit für die Speicherung $O(n^2 \log n)$ für beide Datenstrukturen und für den Speicherplatz $O(n^2)$ bei Intervall-Bäumen und $O(n^2 \log n)$ beim Segment-Baum.

Zusammenfassend gilt für die obere Schranke der Laufzeit zur Speicherung des reaktiven Konfliktgraphen, dessen Knoten in einem R-Baum gespeichert sind und die Kanten in einem:

Speicherstruktur Kanten	obere Schranke Laufzeit		obere Schranke Speicherplatz	
	Knoten	Kanten	Knoten	Kanten
R-Baum	$O(n \log n) + O(n^2 \log n)$		$O(n) + O(n^2)$	
Segment-Baum	$O(n \log n) + O(n^2 \log n)$		$O(n) + O(n^2 \log n)$	
Intervall-Baum	$O(n \log n) + O(n^2 \log n)$		$O(n) + O(n^2)$	

Alternativ zu den hier vorgestellten geometrischen Standarddatenstrukturen können auch andere geeignete Strukturen verwendet werden, wie das Grid-File, der Quadtree, etc. Es wird auf eine ausführliche Vorstellung und Diskussion im Weiteren verzichtet. Einen Überblick geben [SAM90], [BKO97] und [OTW93].

Die Wahl der geometrischen Datenstruktur zur Speicherung der reaktiven Konfliktgraphkanten sollte in Abhängigkeit der Laufzeit der Anfragen in der Interaktionsphase entschieden werden. In Abschnitt 5.1.1 sind die oberen Schranken der Laufzeiten von Anfragen zur Extraktion des statischen Konfliktgraphen aus dem reaktiven mit den verschiedenen Datenstrukturen hergeleitet. Dabei zeigt sich, dass die Worst-Case Laufzeiten für die Extraktion der Kanten gleich sind, wenn sie alle in einem R-Baum gespeichert sind oder lokal an ihren inzidenten Knoten in Segment- oder Intervall-Bäumen.

4.5 Geometriotypwechsel und Generalisierung – Maßstabsunabhängigkeit des reaktiven Konfliktgraphen

Der reaktive Konfliktgraph ist aus Sicht der Algorithmen und Datenstrukturen nicht auf ein Maßstabsintervall beschränkt. In ihm sind alle wichtigen Informationen für die Interaktionsphase enthalten, die für jeden beliebigen Maßstab abgeleitet werden können. Er ist somit maßstabsunabhängig. Das bedeutet, dass maßstabsabhängige Informationen, wie z.B. der Beschriftungsraum durch einfach berechenbare Funktionen aus ihm abgeleitet werden können. Eine maßstabsabhängige Aussage, die nur zu einem konkreten Maßstab gilt, ist z.B. die Beschriftungsschwierigkeit.

Dem reaktiven Konfliktgraph sind trotz der beschriebenen technischen Maßstabsunabhängigkeit Einschränkungen gesetzt aus Gründen der kartographisch gerechten Darstellung und den verwendeten Geodaten.

Der vorgestellte reaktive Konfliktgraph setzt voraus, dass jedes zu beschriftende Objekt die gleiche Signatur in jedem Maßstab besitzt. Nur die Ausdehnung der Signatur wird in Abhängigkeit vom Maßstab geändert. Bei einem großen Maßstab wird die Signatur größer dargestellt als bei einem kleinen Maßstab, analog zur Schriftgröße, die jedoch einer anderen Änderungsfunktion in Abhängigkeit des Maßstabs unterliegt.

Nicht berücksichtigt werden Geometriotypwechsel sowie weitere Änderungen, die bei der Generalisierung eintreten, z.B. Linienglättung oder Verdrängung. Geometriotypwechsel finden z.B. bei der Darstellung von Orten wie folgt statt: Bei einem Maßstab von 1:25.000 ist die Darstellung von Orten flächenhaft und bei einem kleinen Maßstab von 1:500.000 ist sie punkthaft. Linienglättungen treten z.B. bei Flüssen auf, deren Geometrien in einem großen Maßstab vorliegen und nun in einem kleineren Maßstab dargestellt werden sollen. Eine detaillierte Darstellung, Verfahren und Diskussion der Generalisierung und ihrer Begleiterscheinungen kann in [HGM02] nachgelesen werden. Für den Konfliktgraphen entscheidend ist, dass ein Kartenobjekt – unabhängig davon ob es zu beschriften ist oder nicht – bei diskreten Maßstäben seine Geometrie ändert.

Diesem Umstand kann auf zwei Weisen Rechnung getragen werden. Die Anzahl der Maßstäbe, bei denen eine Änderung der Signatur oder ein Generalisierungsschritt eintritt, sollte so klein wie möglich gehalten werden. Für die Maßstabsintervalle zwischen je zwei benachbarten Maßstäben wird jeweils ein eigener reaktiver Konfliktgraph erzeugt. Abhängig von dem Maßstab, zu dem eine Kartenbeschriftung während der Interaktionsphase erzeugt werden soll, wird der passende reaktive Konfliktgraph ausgewählt. Diese Vorgehensweise hat zwei Vorteile. Zum einen werden (geometrische) Änderungen, die die Generalisierung verursacht, berücksichtigt. Zusätzlich kann die Auflösung der Geo-Daten, die dem Konfliktgraphen zu Grunde liegen, abhängig vom jeweiligen Maßstabsintervall gewählt werden.

Ein alternativer Ansatz ist die homogene Berücksichtigung der durch die Generalisierung veränderten Geometrien von Objekten in einem reaktiven Konfliktgraphen, der durchgängig für alle Maßstäbe gilt. Dabei wird die im letzten Abschnitt beschriebene Datenstruktur ausgenutzt. Wie in Abschnitt 4.4 beschrieben, besitzt jeder Knoten, d.h. jedes zu beschriftende Kartenobjekt, einen Darstellungsbereich. Dieser ist durch den zum Knoten gehörigen Deselektionsmaßstab (*deselect*) und dem Maßstab 1:1 vorgegeben. Die 1:1-Zuordnung zwischen Knoten und Objekt, jeder Knoten repräsentiert ein Objekt und jedes Objekt wird durch einen Knoten repräsentiert, wird im Folgenden zu einer 1:*n*-Zuordnung geändert. Das bedeutet, dass jeder Knoten genau ein Objekt für ein bestimmtes Maßstabsintervall repräsentiert und jedes Objekt abhängig vom Maßstab von *n* Knoten repräsentiert wird.

Um Geometrieänderungen in einem Konfliktgraphen realisieren zu können, werden Knoten, analog zu den Konfliktgraphkanten, mit zwei Attributen versehen, die ein Maßstabsintervall bilden, innerhalb dessen der Knoten der Repräsentant des Objekts ist und es dargestellt wird. Das Attribut *lowerBound* ersetzt anschaulich betrachtet das Attribut *deselect* und beschreibt den Maßstab, unterhalb dem das zum Knoten gehörige Objekt nicht mehr dargestellt wird. Das Attribut *upperBound* repräsentiert den Maßstab oberhalb dessen das Objekt bzw. der Knoten nicht mehr gültig ist. Die Speicherung und der Zugriff auf die Knoten im reaktiven Konfliktgraphen erfolgen über die dreidimensionale geometrische Datenstruktur, wie in Abschnitt 4.4 beschrieben.

Die Maßstabsintervalle der Knoten, die ein Objekt repräsentieren, sind disjunkt und grenzen lückenlos aneinander. Die Darstellung des Rheins ist z.B. im Maßstabsintervall 1:1 bis

1:100.000 flächenhaft; im Maßstabsintervall 1:100.000 bis 1:250.000 linienhaft; im Maßstabsintervall 1:250.000 bis 1:750.000 linienhaft mit Linienglättungen; im Maßstabsintervall 1:750.000 bis 1:1.500.000 linienhaft mit weiteren Linienglättungen und Generalisierungsmaßnahmen; usw. Jedes Intervall entspricht einen Knoten des reaktiven Konfliktgraphen mit entsprechender Attributierung.

Die Kanten müssen in beiden Varianten abhängig vom Darstellungsmaßstab der Knoten erzeugt werden.

Dieses Vorgehen der Aufteilung in Maßstabsintervalle ist technisch gesehen identisch mit dem des ersten Ansatzes, wo die reaktiven Konfliktgraphen einen Maßstabsgültigkeitsbereich besitzen. Im ersten Ansatz kann jedoch die Anzahl der reaktiven Konfliktgraphen sehr groß werden. Der vorletzte Absatz beschreibt die Aufteilung des Rheins in z.B. vier Intervalle. Die Darstellung des Mains verlangt z.B. auch eine Aufteilung in vier Intervalle, jedoch liegen die Intervallgrenzen anders als beim Rhein bei 1:1 bis 1:75.000, 1:75.000 bis 1:200.000, 1:200.000 bis 1:500.000; 1:500.000 bis 1:1.000.000. Das führt bei der Modellierung mit dem ersten Ansatz zu acht reaktiven Konfliktgraphen mit den Intervallbereichen 1:1 bis 1:75.000; 1:75.000 bis 1:100.000; 1:100.000 bis 1:200.000; 1:200.000 bis 1:250.000; 1:250.000 bis 1:500.000; 1:500.000 bis 1:750.000; 1:750.000 bis 1:1.000.000; 1:1.000.000 bis 1:1.500.000. In jedem reaktiven Konfliktgraphen werden die beiden Objekte jeweils durch einen Knoten repräsentiert, so dass insgesamt sechzehn Knoten die beiden Objekte repräsentieren. Im ungünstigsten Fall ist die Anzahl der Knoten gleich dem Produkt der Anzahl der Intervalle, in die die Objekte zerfallen und der Anzahl der Objekte. Mit dem zweiten Ansatz repräsentieren acht Knoten (also die Hälfte) die beiden Objekte. Im Gegensatz zum ersten Ansatz entspricht die Anzahl der Knoten der Summe der Anzahl der Intervalle in dem die Objekte zerfallen. Die Anzahl der Knoten ist daher geringer als die im ersten Ansatz.

Der Nachteil des zweiten Ansatzes ist jedoch die Größe, d.h. die Anzahl der Objekte in dem einzigen reaktiven Konfliktgraphen und damit die Laufzeitkosten für die Abfrage der Speicherstruktur. Eine Kombination der beiden Ansätze ist daher sinnvoll.

Ausgehend von einem maximalen Maßstabsbereich, für den Beschriftungen erzeugt werden sollen, wird dieser grob, z.B. in vier Intervalle, unterteilt nach dem ersten Ansatz. Für diese kleineren Maßstabsintervalle wird der zweite Ansatz verwendet. Dadurch kann zum einen die Anzahl der Knoten pro reaktiven Konfliktgraph reduziert werden, als auch die Anzahl der Knoten aller Konfliktgraphen.

4.6 Integration von Informationsverlusten in den Konfliktgraphen

Bisher wurden ausschließlich Konflikte zwischen potentiellen Beschriftungen, d.h. Beschriftungsräumen, behandelt. Informationsverluste treten dagegen bei Verdeckungen von Signaturen durch Schriftobjekte auf, die dieser Abschnitt behandelt.

Im Unterschied zu Konflikten zwischen Schriftobjekten bzw. Beschriftungsräumen untereinander sind diese Verdeckungen bzw. Konflikte, die die Informationsverluste verursachen, lokal. Daher wird, nach der Generierung des reaktiven Konfliktgraphen, jedes zu beschriftende Objekt bzw. Knoten separat betrachtet und die Informationen an dem Knoten gespeichert. Sie sind somit nicht direkt Bestandteil des Konfliktgraphen.

In der Präprocessingphase wird nicht eine Beschriftungsposition für jedes zu beschriftende Objekt festgelegt, sondern potentielle Beschriftungskonflikte ermittelt, die der Beschriftungsraum, der alle möglichen Beschriftungspositionen umfasst, verursacht. Erst während der Interaktionsphase wird eine Beschriftungsposition ausgewählt. Analog ist das Vorgehen bei der Verdeckung von Kartenobjekten durch Schriftobjekte bzw. Beschriftungsräume.

In der Präprocessingphase werden nach der Erzeugung des reaktiven Konfliktgraphen Verdeckungen zwischen Beschriftungsräumen und Kartenobjekten, die keine Schriftobjekte oder Beschriftungsräume sind, ermittelt, wie in Algorithmus 8 dargestellt. Die gewonnenen Informationen werden lokal an dem zugehörigen Konfliktgraphknoten gespeichert.

```

1  Input: Konfliktgraphknoten und deren Deselektionsmaßstäbe (reaktiver Konfliktgraph)
      sowie eine Liste aller Kartenobjekte ohne Schriftobjekte
Output: Konfliktgraphknoten erweitert um eine Liste mit Kartenobjekten und Maßstäben,
      ab denen die Überlappungen zwischen diesen und dem Knoten eintreten
5  // jeder Knoten hat eine (leere) Liste mapConflicts für Tupeleinträge (nicht zu
    // beschriftender Konfliktpartner, Schnittmaßstab)
    // mapObjects = Liste aller Kartenobjekten (ohne Schriftobjekte)
für jeden Knoten des reaktiven Konfliktgraphen (currentNode):
    für jedes Element aus der Liste mapObjects (nonLabelObject):
10     cutScale = Schnittmaßstab zwischen currentNode und nonLabelObject
        wenn cutScale größer Deselektionsmaßstab (deselect) von currentNode
            und cutScale größer Deselektionsmaßstab (deselect) von nonLabelObject
            dann
                speichere (nonLabelObject, cutScale) in mapConflicts von currentNode
15     sortiere Liste mapConflicts von currentNode absteigend nach dem
        Schnittmaßstab (2. Tupeleintrag)

```

Algorithmus 8 Ermittlung der Verdeckungen, die zwischen Beschriftungsräumen und Kartenobjekten, die keine Schriftobjekte sind, entstehen können. Für jedes zu beschriftende Objekt werden die Verdeckungen, die der zugehörige Beschriftungsraum mit anderen Kartenobjekten verursacht, bestimmt. Die Berechnung der Schnittmaßstäbe erfolgt mit dem für die Beschriftungsräumen entwickelten Verfahren. Die ermittelten Konflikte werden zu jedem Objekt in der Liste *mapConflicts* gespeichert.

- Algorithmus 8 – Zeilen 1 bis 9

Der Algorithmus durchläuft alle Knoten des reaktiven Konfliktgraphen und berechnet für jeden Knoten die Schnittmaßstäbe zu allen Kartenobjekten, die keine Schriftobjekte oder Beschriftungsräume sind.

- Algorithmus 8 – Zeile 10 – Schnittmaßstab

Mit den gleichen Verfahren zur Ermittlung von Konflikten zwischen Beschriftungsräumen werden die Schnittmaßstäbe zwischen Beschriftungsräumen und Kartenobjekten (ungleich Schrift/Beschriftungsraum) bestimmt. Im Unterschied zur Schnittmaßstabsbestimmung zwischen zwei Beschriftungsräumen wächst hier nur ein Raum.

Bei der Verwendung der Gleichung (4.14) und (4.15) für die Bestimmung des Schnittmaßstabs zwischen dem Beschriftungsraum eines punkthaften zu beschriftenden Objekts mit einem punkthaften Kartenobjekt und der Gleichung (4.41) für den Schnittmaßstab zwischen dem Beschriftungsraum eines linienhaften zu beschriftenden Objekts mit einem linienhaften Kartenobjekt, wird für das nicht zu beschriftende Objekt die *fixHeight*, *fixWeight* und *fixDistanceSymbolLabel* auf Null gesetzt. Mit den Parametern *fixSymbolHeight*, *fixSymbolWidth* und *fixScale* wird die Größenänderung der Signaturen abhängig vom Maßstab zu einem Referenzmaßstab und Größe berücksichtigt. Für die Bestimmung des Schnittmaßstabs zwischen dem Beschriftungsraum eines linienhaften Objekts und einem punkthaften Kartenobjekts kann ebenfalls auf das Verfahren für die Schnittmaßstabsbestimmung zwischen zwei linienhaften zu beschriftenden Objekten zurückgegriffen werden. Dabei wird ausgenutzt, dass das Verfahren von den zwei nächstgelegenen Punkten auf den linienhaften Objekten ausgeht. Einer dieser Punkte wird durch das punkthafte Objekt ersetzt und die zugehörigen Parameter *fixHeight* und *fixDistanceSymbolLabel* auf Null gesetzt. Der Schnittmaßstab

zwischen einem punkthaften zu beschriftenden Objekt und einem linienhaften Kartenobjekt wird auf die Schnittmaßstabsbestimmung zwischen zwei punkthaften zu beschriftenden Objekten zurückgeführt. Dabei repräsentiert je eine Gleichung (4.14) einen Sektor (vgl. Abbildung 53) des punkthaften Objekts. Für jeden Sektor wird der nächstliegende Punkt des linienhaften Kartenobjekts zum punkthaften Objekt bestimmt, wenn es einen solchen gibt, und in die Gleichung eingesetzt. Wie bei den anderen Fällen erhalten die Parameter *fixHeight*, *fixWeight* und *fixDistanceSymbolLabel* den Wert Null. Das Maximum der vier Gleichungen ist der gesuchte Schnittmaßstab.

Die Beschriftungsräume flächenhafter Objekte werden auf die linienhafter oder punkthafter zu beschriftender Objekte zurückgeführt, wie in Abschnitt 4.3.9 und 4.3.10 beschrieben. Flächenhafte Kartenobjekte, die keine Schriftobjekte sind, werden abhängig von ihrer Größe als punkthafte Objekte behandelt oder, wenn die Informationen in der Flächenbegrenzung liegen, als geschlossenes linienhaftes Objekt. Eine gesonderte Betrachtung flächenhafter Objekte bzw. zu beschriftender flächenhafter Objekte kann daher entfallen.

- Algorithmus 8 – Zeile 11 bis 14

Wie die Schriftobjekte, die unter einem bestimmten Maßstab (*deselect*) nicht mehr dargestellt werden, existiert der Deselektionsmaßstab auch für das Kartenobjekt. Dieser ist vorgegeben und am Kartenobjekt als Attribut abgelegt.

Wenn der ermittelte Schnittmaßstab größer als der Deselektionsmaßstab des zur Zeit betrachteten Knotens (*currentNode*) und größer als der Deselektionsmaßstab des betrachteten Kartenobjekts (*nonLabelObject*) ist, dann wird der berechnete Schnittmaßstab und ein Zeiger auf das Kartenobjekt am Knoten in einer Liste (*mapConflicts*) gespeichert. Die if-Abfrage stellt sicher, dass nur Kartenobjekte mit einer potentiellen Verdeckung gespeichert sind.

- Algorithmus 8 – Zeile 15 bis 16

Für eine effiziente Auswertung und Bestimmung von Verdeckungen in der Interaktionsphase wird die Liste (*mapConflicts*) abschließend absteigend nach dem Maßstab sortiert vor der Bearbeitung des nächsten Knotens.

Maßstabsübergängen und Generalisierung, die in Abschnitt 4.5 diskutiert wurden, sind können problemlos in das vorgestellte Verfahren integriert werden.

4.7 Topologisch korrekte Lage von Schrift zum zu beschriftenden Objekt

Neben der Vermeidung von Schriftüberlappungen und Verdeckungen von Kartenobjekten durch Schrift sollte die Schrift korrekt in Bezug auf Trennlinien platziert werden [ELL01]. Die Stadt Aachen sollte z.B. so beschriftet werden, dass der Schriftzug in dem flächenhaften Objekt „Deutschland“ liegt genauso wie „Frankfurt an der Oder“ (Abbildung 66). In diesem Fall ist die Landesgrenze eine topologische Trennlinie. Sie hilft dem Kartenleser zu erkennen, dass die Stadt innerhalb des flächenhaften Objekts liegt, in diesem Fall innerhalb Deutschlands. Die Beschriftung von Küstenorten erfolgt umgekehrt. Küstenorte (mit Zugang zum Wasser) werden, um diese Eigenschaft explizit darzustellen, im „Wasser“ beschriftet, während Orte die keinen Zugang zum Wasser haben, auf dem „Land“ beschriftet werden (Abbildung 66) [IMH62].

Um die Interaktionsphase, in der die Auswahl der Beschriftungsposition stattfindet, bei der Berücksichtigung von topologischen Trennlinien zu unterstützen, können in der Prä-

processingphase die dafür nötigen Informationen gewonnen werden. Dies geschieht durch eine Modifikation von Algorithmus 8, die in Algorithmus 9 skizziert ist.

Jedes Kartenobjekt erhält eine zusätzliche Information, ob es für eine topologisch korrekte Beschriftung bedeutend ist. Die Knoten, d.h. jedes zu beschriftende Objekt, erhalten eine zusätzliche Liste (*topology*), in die die Objekte gespeichert werden, die für eine topologisch korrekte Beschriftung des Objekts berücksichtigt werden müssen. Algorithmus 9 ist ansonsten unverändert zu Algorithmus 8. Abschnitt 4.6 beschreibt die verbleibenden Details des Algorithmus.

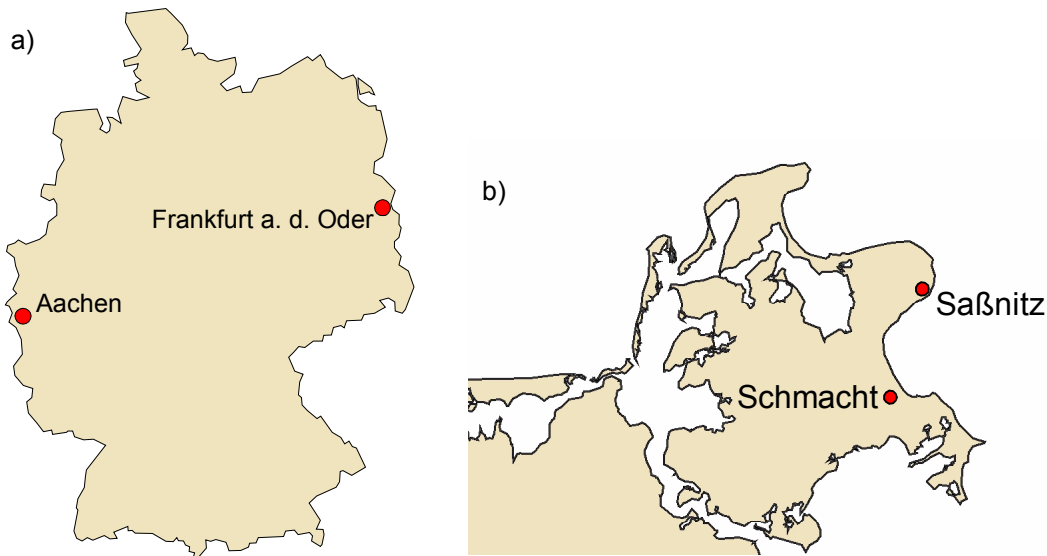


Abbildung 66 Topologisch korrekte Lage der Schrift zum zu beschriftenden Objekt: a) zeigt die topologisch korrekte Beschriftung von in Grenznähe gelegenen Orten und b) von Orten mit bzw. ohne Zugang zum Wasser.

```

1  Input: Konfliktgraphknoten und deren Deselektionsmaßstäbe (reaktiver Konfliktgraph)
      sowie eine Liste aller Kartenobjekte ohne Schriftobjekte
Output: Konfliktgraphknoten erweitert um eine Liste mit Kartenobjekten und Maßstäben,
      ab denen die Überlappungen zwischen diesen und dem Knoten eintreten sowie eine
5     zweite Liste mit Kartenobjekten und Maßstäben ab denen das Objekt für eine
      topologisch korrekte Beschriftung des Knotens berücksichtigt werden muss
// jeder Knoten hat eine (leere) Liste mapConflicts für Tupeleinträge
// jeder Knoten hat eine (leere) Liste topolgy für Tupeleinträge
// mapObjects = Liste aller Kartenobjekten (ohne Schriftobjekte)
10 für jeden Knoten des reaktiven Konfliktgraphen (currentNode):
      für jedes Element aus der Liste mapObjects (nonLabelObject):
          cutScale ≙ Schnittmaßstab zwischen currentNode und nonLabelObject
          wenn cutScale größer Deselektionsmaßstab (deselect) von currentNode
              und cutScale größer Deselektionsmaßstab (deselect) von nonLabelObject
15         dann
              speichere (nonLabelObject, cutScale) in mapConflicts von currentNode
              wenn nonLabelObject topologische Beschriftung beachten
                  dann
                      speichere (nonLabelObject, cutScale) in topology von currentNode
20     sortiere Liste mapConflicts von currentNode absteigend nach dem
          Schnittmaßstab (2. Tupeleintrag)
          sortiere Liste topology von currentNode absteigend nach dem
          Schnittmaßstab (2. Tupeleintrag)

```

Algorithmus 9 Erweiterung von Algorithmus 8 um die Ermittlung der Objekte, die für die topologisch korrekte Lage von Schriftobjekten nötig sind. Die Kartenobjekte besitzen Informationen über ihre Bedeutung in Bezug auf topologische Beschriftung. Wenn es zu einer Verdeckung durch den Beschriftungsraum kommt, muss das Kartenobjekt, wenn es für die topologische Beschriftung wichtig ist, berücksichtigt werden. Dazu wird es in der Liste *topology* mit dem Schnittmaßstab gespeichert. Diese wird für jeden Knoten absteigend nach dem Schnittmaßstab sortiert.

4.8 Zusammenfassung

In diesem Kapitel wurde die Generierung des reaktiven Konfliktgraphen, der die Basis für die Beschriftung in der Interaktionsphase bildet, stufenweise entwickelt. Der reaktive Konfliktgraph (Abschnitt 4.2) besteht aus attribuierten Knoten, die die zu beschriftenden Objekte repräsentieren und attribuierten Kanten, die die potentiellen Schriftkonflikte repräsentieren.

Zunächst wurde die Integration von zu beschriftenden punkthaften Objekten detailliert behandelt bevor systematisch eine Erweiterung um zu beschriftende linien- und flächenhafte Objekte sowie Siedlungsflächen behandelt wurden. In Abschnitt 4.3.9 wird die Beschriftung flächenhafter auf linienhafte Objekte und in Abschnitt 4.3.10 die Beschriftung von Siedlungsflächen auf das Konzept der Beschriftung punkthafter zurückgeführt.

Dabei erfolgte zunächst die Definition der Beschriftungsräume der einzelnen Beschriftungsklassen parametrisiert mit dem Maßstab, wobei berücksichtigt wird, dass sich die Schriftgröße nicht proportional zum Maßstab ändert. Die Beschriftung eines Objekts liegt immer innerhalb seines Beschriftungsraums.

Potentielle Schriftüberlappungen werden im Konfliktgraphen durch Konfliktgraphkanten repräsentiert, deren Konstruktion abhängig von den Beschriftungsklassen der beteiligten Konfliktpartner ist (zwischen zwei punkthaften Objekten; zwischen zwei linienhaften Objekten; zwischen punkt- und linienhaften Objekten). Jede Kante besitzt zwei Maßstabsangaben, die ein Maßstabsintervall bilden. Die obere Intervallgrenze entspricht dem Schnittmaßstab, ab und unterhalb dem es zu einem Konflikt zwischen den beiden zur Kante inzidenten Konfliktpartnern kommt. Dieser wird analytisch bestimmt und ist abhängig von den Beschriftungsklassen der beiden beteiligten Konfliktpartner. Zusammen mit der Einführung der verschiedenen Beschriftungsklassen wird dieser bestimmt: zwischen zwei punkthaften zu beschriftenden Objekten in Abschnitt 4.3.2.1; zwischen zwei linienhaften Objekten in Abschnitt 4.3.5.1, zwischen punkt- und linienhaften Objekten in Abschnitt 4.3.7.1.

Die untere Intervallgrenze der Konfliktgraphkante, der Deselektionsmaßstab, leitet sich aus dem Deselektionsmaßstab des zu beschriftenden Objekts ab. Der Deselektionsmaßstab der Kante ist der Maßstab, ab und unter dem der Konflikt nicht mehr auftritt, da einer der beiden Konfliktpartner nicht mehr beschriftet wird. Er ist also gleich dem Maximum der Deselektionsmaßstäbe der Knoten.

Der Deselektionsmaßstab eines Konfliktgraphknoten ist der Maßstab, ab und unterhalb dem das zu beschriftende Objekt nicht mehr beschriftet wird. Dieser wird in Abhängigkeit der Beschriftungsschwierigkeit bestimmt, die beschreibt, wie schwer es ist ein Objekt zu einem vorgegebenen Maßstab zu beschriften. Für punkthafte Objekte wurden drei verschiedene Kriterien Knotengrad, größter freier Winkel und freie Beschriftungsfläche entwickelt und vorgestellt (Abschnitt 4.3.3).

Das graphtheoretische Kriterium „Knotengrad“ ist die Anzahl der Konflikte, die identisch ist zu der Anzahl der inzidenten Kanten im Konfliktgraphen. Es werden hierbei nur Konflikte mit Objekten mit höherer Priorität berücksichtigt. Es erfolgt eine Beurteilung der Beschriftungsschwierigkeit über die Konflikte im Gegensatz zu den beiden anderen Kriterien.

In dem geometrischen Kriterium „größter freier Winkel“ wird die Lage der Konflikte zu Objekten mit höherer Priorität durch Konfliktsektoren repräsentiert. Über diese kann der konfliktfreie Sektor mit dem größten Winkel bestimmt werden. Je größer dieser größte freie Winkel ist, desto einfacher ist es das Objekt zu beschriften.

Das geometrische Kriterium „freie Beschriftungsfläche“ ermittelt und bewertet die verbleibenden konfliktfreien Beschriftungsräume, die nach Berücksichtigung von Konflikten mit

Konfliktpartnern höherer Priorität verbleiben. Je größer diese Fläche ist, desto einfacher ist es das Objekt zu beschriften.

Die Diskussion der Vor- und Nachteile der Kriterien in Abschnitt 4.3.3.4 ergab, dass eine Kombination der Kriterien freie Beschriftungsfläche und Knotengrad sich dabei als beste herausstellte.

Die Bestimmung der Beschriftungsschwierigkeit für linienhafte Objekte basiert auf einer Erweiterung des Verfahrens freie Beschriftungsfläche für punkthafte Objekte (Abschnitt 4.3.6).

Der Algorithmus zur Erzeugung des reaktiven Konfliktgraphen, der alle Beschriftungsklassen berücksichtigt, ist in Abschnitt 4.3.11 dargestellt während Methoden und Strukturen zu seiner effizienten Speicherung in Abschnitt 4.4 behandelt werden. Eine Diskussion der insbesondere kartographischen Einschränkungen, die durch Maßstabsübergänge und Generalisierung entstehen, erfolgt in Abschnitt 4.5 zusammen mit der Einordnung des Begriffs Maßstabsunabhängigkeit im Kontext des reaktiven Konfliktgraphen.

Die Integration weiterer kartographischer Beschriftungsregeln, die Berücksichtigung von Informationsverlust und die topologisch korrekte Lage von Schrift zum zu beschriftenden Objekt in Bezug auf Trennlinien, in den reaktiven Konfliktgraphen erfolgt abschließend in den Abschnitten 4.6 und 4.7.

Kapitel 5

Vermeidung von Konflikten

Die Interaktionsphase, die in diesem Kapitel beschrieben wird, baut auf den Ergebnissen der Präprocessingphase (Kapitel 4) auf. Die Präprocessingphase findet vor einer Interaktion des Nutzers mit dem System statt. In ihr werden der reaktive Konfliktgraph erzeugt sowie alle Informationen gewonnen, die für die spätere Beschriftung notwendig sind und im Vorhinein bestimmt werden können. Diese werden in einer geometrischen Datenstruktur gespeichert.

Während der Interaktionsphase stellt der Nutzer Anfragen an das System und fordert Kartenausschnitte und damit die zugehörigen Beschriftungen zu beliebigen Maßstäben an. Dies kann entweder durch eine direkte Anfrage – Ausschnittskordinaten und Maßstab – oder durch die Navigation des Nutzers durch Karten erfolgen. Dazu werden die Funktionen Zoomen (Maßstabswahl) und Scrollen (Kartenausschnitt) genutzt. Jede dieser Aktionen führt zu einer neuen Kartendarstellung und erfordert somit eine (vollständige) Neubeschriftung der Karte.

Da sich Schriftgröße und Maßstab nicht im gleichen Verhältnis zueinander ändern, führt eine Änderung des Maßstabs zu einer Änderung des Verhältnisses zwischen Beschriftungsfläche und der Fläche der Schriftobjekte (vgl. Abbildung 25). Dies hat eine Änderung der Konflikte zufolge. Für die Kartendarstellung bedeutet das eine Änderung der zu beschriftenden Objekte und somit eine Neubeschriftung.

Eine Verschiebung des Kartenausschnitts führt ebenfalls zu einer Neubeschriftung, da sich die Beschriftung der am Rand des Ausschnitts liegenden Objekte ändert. Bei punkthaften Objekten kann das zu einem „Umklappen“ der Beschriftung führen. Damit ist die Änderung – anschaulich die Spiegelung – der Beschriftungsposition in Bezug auf die Lage des zu beschriftenden Objekts gemeint, die sich über den ganzen Kartenausschnitt fortpflanzen kann (Abbildung 67). Analoges gilt für die Beschriftung anderer Beschriftungsklassen.

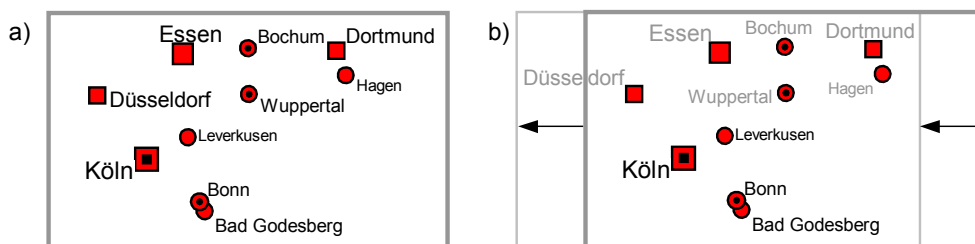


Abbildung 67 Änderung der Beschriftung durch scrollen (verschieben des Kartenausschnitts).

Für die Beschriftung einer Karte bzw. eines Kartenausschnitts zu einem spezifischen Maßstab wird aus dem in der Präprocessingphase gewonnenen reaktiven Konfliktgraphen ein statischer

abgeleitet (Abbildung 15). Dieser enthält alle zu beschriftenden Objekte, ihre Konflikte und zur Beschriftung notwendigen Informationen für den spezifischen Ausschnitt und Maßstab. Die Ableitung des statischen Konfliktgraphen behandelt Abschnitt 5.1. Die (endgültige) Auswahl der zu beschriftenden Objekte und der Beschriftungspositionen erfolgt in Abschnitt 5.2.

Abschließend kann eine Feinplatzierung der Beschriftungen erfolgen mit der Minimierung von Informationsverlusten und der topologisch korrekten Lage der Schrift zum zu beschriftenden Objekt, wie in Abschnitt 5.3 dargestellt.

Das Konzept des Konfliktgraphen ist durch die Zweiteilung in die Präprocessing- und Interaktionsphase geprägt. Die Beschriftung ist somit auf die Objekte und ihre Beschriftungsprioritäten beschränkt, die während der Präprocessingphase im reaktiven Konfliktgraphen berücksichtigt wurden. Bei unterschiedlichen thematischen Karten für den gleichen Kartenausschnitt sind die Beschriftungsprioritäten der Objekte und die Auswahl der zu beschriftenden Objekte unterschiedlich. Eine Möglichkeit wäre für jede mögliche thematische Karte einen eigenen reaktiven Konfliktgraphen zu erzeugen oder einen, in dem alle möglichen Thematiken enthalten sind. Abgesehen von dem hohen Datenvolumen ist der erste Ansatz auf die in der Präprocessingphase vorgesehenen Thematiken beschränkt. Eine Beschriftung einer Karte, die sich aus zwei oder mehreren Thematiken zusammensetzt, ist nicht möglich, wenn sie nicht in der Präprocessingphase berücksichtigt wurde. Im zweiten Ansatz entsteht ein großer gemeinsamer reaktiver Konfliktgraph, aus dem die Ableitung des statischen Konfliktgraphen allerdings laufzeitintensiv ist. In Abschnitt 5.4 wird ein Verfahren vorgestellt, das zwei statische Konfliktgraphen zu einem Neuen zusammenführt. Dies geschieht in der Interaktionsphase.

5.1 Ableitung des statischen aus dem reaktiven Konfliktgraphen

Wie einleitend zu diesem Kapitel beschrieben und in Abbildung 15 dargestellt, wird aus dem reaktiven Konfliktgraphen, der in der Präprocessingphase erzeugt wird, ein statischer Konfliktgraph abgeleitet. Der statische Konfliktgraph ist eine Spezialisierung des reaktiven, da er auf einen (Karten-)Ausschnitt und einen Maßstab eingeschränkt ist.

Der reaktive Konfliktgraph ist in einem dreidimensionalen R-Baum (2.2.2.2.3) gespeichert. Als Schlüssel werden die Geometrien der Objekte (im Maßstab 1:1) verwendet, die der Knoten repräsentiert (vgl. die Diskussion in Abschnitt 4.4). Die ersten beiden Dimensionen speichern die Geometrie der Objekte und die dritte Dimension das Maßstabsintervall, in dem das Objekt existiert. Anschaulich ist das in Abbildung 68 visualisiert. Die z-Achse repräsentiert den Maßstab. Die verschiedenen „Layer“ repräsentieren einige diskrete Maßstäbe des Kontinuums. Mit zunehmendem Maßstab nimmt die Kartenfläche (Größe des Layers) und mit ihr die Anzahl der zu beschriftenden Objekte zu. Die in den Layern der Grafik dargestellte Kartengrundfläche ist die Boundingbox aller Objekte.

Die Extraktion der Knoten des statischen Konfliktgraphen erfolgt durch eine „overlap“-Anfrage an die geometrische Datenstruktur. Die Anfrage umfasst den Kartenausschnitt, durch Realweltkoordinaten (Maßstab 1:1) beschrieben, und den Maßstab. Der Kartenausschnitt liefert die ersten beiden Dimensionen, d.h. die Bereichsanfrage und der Maßstab die dritte Dimension, d.h. das Aufspieß-Problem¹⁷ des „Anfrageraums“. Der Anfrageraum ist ein Quader im dreidimensionalen Raum, der parallel zur x- und y-Achse liegt und dessen Höhe 0 ist.

¹⁷ Eine inverse Bereichsanfrage wird in der Fachliteratur als „Aufspieß-Problem“ oder „Aufspieß-Frage“ bezeichnet [OTW93]. In dem beschriebenen Kontext ist die Frage nach den Objekten, deren zugehöriges Maßstabsintervall den Anfragemmaßstab aufspießt, d.h. der Anfragemmaßstab innerhalb des Maßstabsintervall des Objekts liegt.

Anschaulich, auf die Darstellung bezogen, liefert der Maßstab den „Layer“ und der Kartenausschnitt schränkt die Layerfläche ein.

Die Ergebnisobjekte einer „overlap“-Anfrage müssen bei geometrischen Datenstrukturen, die Boundingboxen als Indizes verwenden, verifiziert werden. Wenn die beiden ersten Dimensionen der Boundingbox eines Ergebnisobjekts nicht vollständig innerhalb des Anfragebereichs liegen, kann es vorkommen, dass die Geometrie des Objekts vollständig außerhalb liegen kann und somit irrelevant für den gesuchten statischen Konfliktgraphen ist (Abbildung 69). Um das zu vermeiden, müssen Ergebnisobjekte, die nicht vollständig innerhalb des Kartenausschnitts liegen, getestet werden.

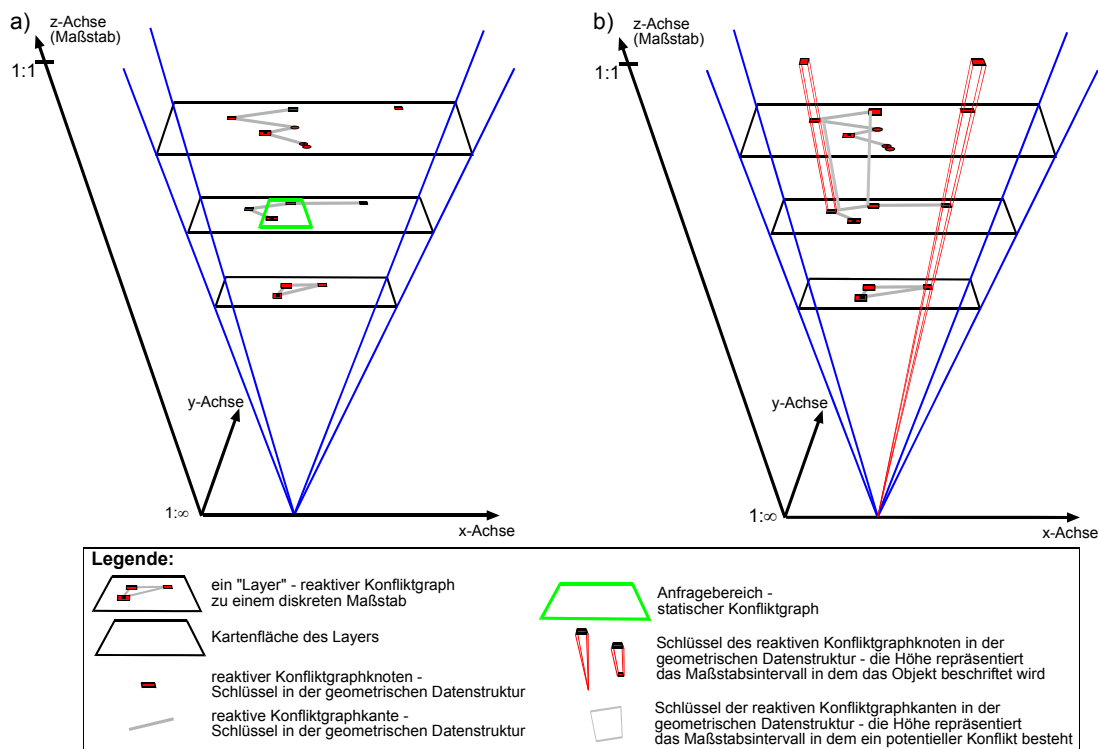


Abbildung 68 Visualisierung der in der geometrischen Datenstruktur gespeicherten Daten. Der Konfliktgraph für drei Maßstäbe ist beispielhaft in a) dargestellt. Objekte in Abhängigkeit vom Maßstab, in dem sie beschriftet werden und ihre Konflikte sind in b) illustriert.

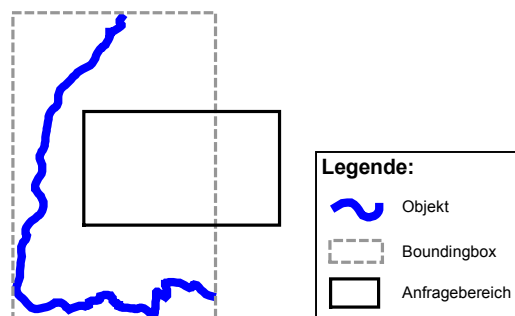


Abbildung 69 Geometrische Objekte, deren Boundingboxen nicht vollständig innerhalb des Anfragebereichs liegen, müssen auf Schnitt getestet werden.

Die Speicherung der Objekte erfolgt im Maßstab 1:1 in der Datenstruktur ohne Signaturierung. Bei einer Anfrage kann der Fall eintreten, dass das Objekt nicht im Anfragebereich liegt, aber die zugehörige Signaturierung im Anfragemassstab diesen Bereich schneidet. Um dieses Problem zu lösen, werden entweder nur Objekte berücksichtigt, die tatsächlich im Anfragebereich liegen oder der Anfragebereich wird etwas vergrößert, die Objekte passend zum Maßstab signaturiert und anschließend auf Schnitt getestet. Im Folgenden wird davon ausgegangen, dass die Anfrage alle Objekte zurückliefert, die berücksichtigt werden müssen.

Wie in Abschnitt 4.4 beschrieben, kann von den gespeicherten Objekten auf die zugehörigen reaktiven Konfliktgraphknoten geschlossen werden. Diese durch die Anfrage selektierten reaktiven Konfliktgraphknoten bilden die Knoten des statischen Konfliktgraphen der Anfrage.

Die Gewinnung der statischen Konfliktgraphkanten aus dem reaktiven Konfliktgraphen kann auf zwei Arten, abhängig von der Speicherung der Kanten, erfolgen. Sind die reaktiven Konfliktgraphkanten in einer geometrischen Speicherstruktur, dem R-Baum, abgelegt, können diese mit einer „overlap“-Anfrage extrahiert werden. Das Vorgehen ist analog zu dem bei der Extraktion der Knoten. Für die Existenz eines Konflikts, den die Konfliktgraphkante repräsentiert, müssen beide inzidenten Knoten in dem abzuleitenden statischen Konfliktgraphen vorhanden sein. Um das sicherzustellen, müssen die aus der Anfrage resultierenden Kanten durchlaufen und es muss getestet werden, ob die inzidenten Knoten in dem statischen Konfliktgraphen enthalten sind (Abbildung 70). Dies kann effizient durch eine Markierung der Knoten des reaktiven Konfliktgraphen erzielt werden: Knoten ist im statischen Konfliktgraphen. Dieser wird gesetzt, sobald der Knoten in der Ergebnismenge der oben beschriebenen Knoten-Anfrage ist.

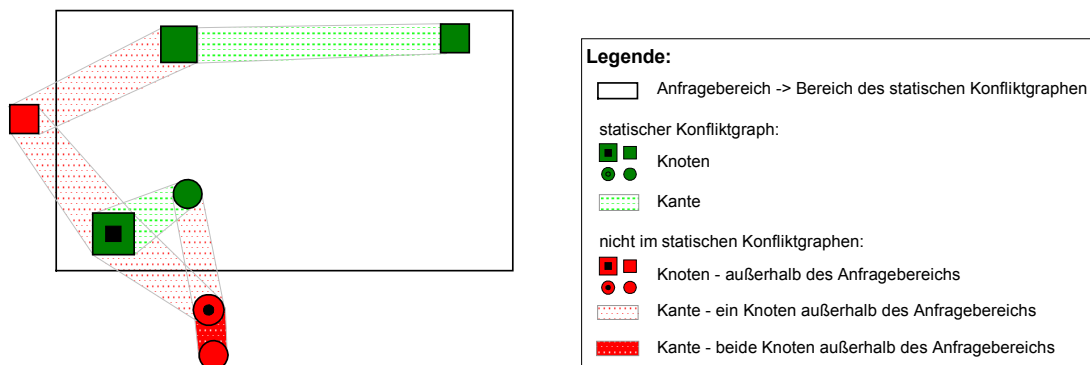


Abbildung 70 Ableitung des statischen Konfliktgraphen aus dem reaktiven Konfliktgraphen. Der reaktive Konfliktgraph umfasst alle Knoten und Kanten der Abbildung, die zur übersichtlicheren Darstellung im Maßstab des statischen Konfliktgraphen dargestellt ist.

Sind die reaktiven Konfliktgraphkanten nicht zusammen in einer Datenstruktur gespeichert, so sind sie indirekt über ihre inzidenten Konfliktgraphknoten erreichbar. Jeder reaktive Konfliktgraphknoten hat Verknüpfungen zu seinen inzidenten Konflikten, d.h. Konfliktgraphkanten. Für eine effiziente Extraktion sind die Verknüpfungen in einer geeigneten geometrischen Datenstruktur abgelegt, wie dem Segment-Baum, mit dem Maßstabsintervall als Schlüssel. Bei der Anfrage handelt es sich um das Aufspieß-Problem: Um alle Kanten des statischen Konfliktgraphen zu erhalten, müssen alle Knoten des statischen Konfliktgraphen durchlaufen werden. Bei jedem Knoten werden die inzidenten Kanten, die zum gesuchten Maßstab passen, mit Hilfe des Aufspieß-Verfahrens extrahiert. Anschließend müssen die extrahierten Kanten durchlaufen werden, um sicherzustellen, dass beide inzidenten Konfliktpartner Knoten im statischen Konfliktgraphen sind. Dies kann analog zum

zuvor beschriebenen Ansatz über Markierungen der Konfliktgraphknoten mit einem Flag effizient geschehen.

5.1.1 Laufzeitanalyse

Die Kosten für die Extraktion der Knoten des statischen Konfliktgraphen aus der geometrischen Datenstruktur des R-Baums können wie folgt abgeschätzt werden: $O(n)$ mit $n \equiv$ Anzahl der Objekte im R-Baum (Anzahl der Knoten des reaktiven Konfliktgraphen). Wie in Abschnitt 2.2.2.2 beschrieben, ist die Abschätzung der Anfragelaufzeit nach oben bei einem R-Baum sehr grob, da sie von den Geometrien der Objekte abhängig ist und von dem verwendeten Verfahren zum Aufteilen von Knoten bei einem Überlauf der Schlüssel während des Einfügens. Die Laufzeit, die der Test verursacht, wenn die Boundingbox nicht vollständig innerhalb des Anfragekartenausschnitts liegt, ist durch n berücksichtigt.

Die Kosten der Extraktion der Kanten ist von ihrer Speicherung abhängig (s. Diskussion in Abschnitt 4.4). Sind die reaktiven Konfliktgraphkanten in einem R-Baum gespeichert, so sind die Kosten analog zu denen der Knoten mit $O(m)$ abzuschätzen, bzw. mit $O(n^2)$ wobei $m \equiv$ Anzahl der Kanten des reaktiven Konfliktgraphen und $n =$ Anzahl der Knoten des reaktiven Konfliktgraphen sind. Der Test, ob die inzidenten Knoten der extrahierten Kanten in dem statischen Konfliktgraphen liegen, kostet $2 * l$ ($l \equiv$ Anzahl der Kanten des statischen Konfliktgraphen) und ist durch das n in der Laufzeitabschätzung berücksichtigt.

Sind die reaktiven Konfliktgraphkanten an ihren inzidenten Knoten in Segment- oder Intervallbäumen gespeichert, beträgt die obere Schranke der Kosten $O(\log p + i)$ mit $p \equiv$ Anzahl der reaktiven Konfliktgraphkanten am betrachteten Konfliktgraphknoten und $i \equiv$ Anzahl der Konfliktgraphkanten, die inzident zum betrachteten Konfliktgraphknoten sind und deren Maßstabsintervall den Anfragemaßstab abdeckt. Der Test, ob die beiden inzidenten statischen Konfliktgraphknoten innerhalb des Kartenausschnitts liegen, kostet mit der Flag-Methode $2 * i$ und ist in der Laufzeitabschätzung enthalten. Im Worst-Case kann der betrachtete Knoten inzident zu allen Kanten des reaktiven Konfliktgraphen sein $p = O(n)$ und deren Maßstabsintervall den Anfragemaßstab abdecken $i = O(n)$, d.h. pro Knoten gilt $O(\log n + m)$. Da im Worst-Case alle Knoten im Anfragekartenausschnitt liegen, kann die Anfrage mit $O(n \log n + n^2) = O(n^2)$ abgeschätzt werden.

Die Ableitung des statischen aus dem reaktiven Konfliktgraphen, dessen Knoten in der geometrischen Datenstruktur eines dreidimensionalen R-Baums gespeichert sind, kann mit $Anfrage_{R-Baum, R-Baum} = O(n) + O(n^2)$, wenn die Kanten in einem R-Baum gespeichert sind bzw. mit $Anfrage_{R-Baum, Segment-/Intervall-Baum} = O(n) + O(n^2)$, wenn die Kanten in Segment- oder Intervall-Bäumen an ihren inzidenten Knoten gespeichert sind, abgeschätzt werden ($n \equiv$ Anzahl der Knoten des reaktiven Konfliktgraphen). Die lokale Speicherung der Kanten an ihren inzidenten Knoten bringt keinen Laufzeitvorteil während der Abfrage, im Vergleich zu einer globalen Speicherung aller Kanten in einem gemeinsamen R-Baum. Ein praktischer Seiteneffekt bei der Verwendung des R-Baums als Speicherstruktur sowohl für die Knoten als auch die Kanten ist die Beschränkung der Implementierung auf eine geometrische Datenstruktur. Anmerkend sollte erwähnt werden, dass Laufzeitanalysen den Worst-Case annehmen. (vgl. auch Abschnitt 4.4)

5.2 Auswahlalgorithmus

Dieser Abschnitt behandelt den Kern der Schriftplatzierung, die Auswahl einer Beschriftungsposition für jedes zu beschriftende Objekt. Ziele sind dabei die Vermeidung von Schriftüberlappungen und die Platzierung in bevorzugten Positionen. In einem zweiten Schritt, der im nächsten Abschnitt beschrieben wird, erfolgt eine Feinplatzierung, deren Ziel die Minimierung der Informationsverluste durch Verdeckung von Kartensignaturen sowie die Berücksichtigung der topologisch korrekten Lage ist.

Im Unterschied zu den in Abschnitt 3.2 vorgestellten heuristischen Verfahren nutzt dieses die entwickelte Methodik des Beschriftungsraums, der nicht auf diskrete Positionen eingeschränkt ist. Der entwickelte Algorithmus 10 basiert auf einem Hill-Climbing-Verfahren [NIL80].

Die Heuristik umfasst neben der Vorselektion durch den statischen Konfliktgraphen die Reihenfolge der zu bearbeitenden Objekte und die Informationen, die zur Wahl einer Beschriftungsposition einfließen.

Die Beschriftung punkthafter und linienhafter Objekte wird explizit berücksichtigt, die Beschriftung flächenhafter Objekte wurde in Abschnitt 4.3.9 auf linienhafte reduziert. Algorithmus 10 ist für die Integration weiterer Beschriftungsarten offen. Hier wird zwischen den Beschriftungen punkthafter und linienhafter Objekte unterschieden. Dabei werden die im Kapitel 4 zuvor entwickelten Algorithmen zur Ermittlung der Beschriftungsschwierigkeit (aus-)genutzt.

Die zu beschriftenden Objekte sind die Knoten des statischen Konfliktgraphen. In Abschnitt 5.1 wurde der statische Konfliktgraph aus dem in der Präprocessingphase aufgebauten reaktiven Konfliktgraphen abgeleitet. Er beinhaltet die zu dem Ausschnitt und Maßstab angemessenen zu beschriftenden Objekte und potentielle Konflikte. Angemessen deshalb, da die Anzahl der zu beschriftenden Objekte in Abhängigkeit des Maßstabs, der geometrischen Lage und daraus resultierenden potentiellen Konflikten vorselektiert sind (Deselektionsmaßstab Kapitel 4). Zusätzlich kennt jedes zu beschriftende Objekt über die Adjazenz im statischen Konfliktgraphen seine potentiellen Konfliktpartner, deren geometrische Ermittlung in dieser Phase wegfällt.

Da die Beschriftung innerhalb des Anfragebereichs liegen muss, werden die Beschriftungsräume, die nicht vollständig innerhalb liegen, dementsprechend verkleinert (Abbildung 71). Technisch gesehen werden die Bereiche des Beschriftungsraums, die außerhalb des Anfragebereichs liegen, als (unvermeidbare) Konflikte behandelt.

Die Auswahl der Beschriftungsposition erfolgt nach der Reihenfolge der Objekte in Abhängigkeit ihrer Priorität, Beschriftungsart und -schwierigkeit.

Die Konfliktpartner für das aktuell betrachtete Objekt werden in drei Klassen eingeteilt, die bei der Positionsfindung unterschiedliche Gewichtungen haben. Absteigend nach Gewichtung sind das die Klasse der schon beschrifteten Objekte (*labeled*), die der noch nicht bearbeiteten Objekte (*untreated*) und die Klasse der Objekte für die keine konfliktfreie Beschriftungsposition ermittelt werden konnte (*unlabeled*). Von Objekten, die den Status *labeled* haben, wird die ermittelte Beschriftungsposition und bei Objekten, die einen der beiden anderen Status haben, wird der Beschriftungsraum berücksichtigt, da keine Beschriftungsposition bekannt ist. Gibt es bei der Berücksichtigung der Beschriftungspositionen der schon beschrifteten Objekte keine konfliktfreie Beschriftungsposition für das aktuell betrachtete Objekt, kann es nicht beschriftet werden und sein Status wird entsprechend auf *unlabeled* geändert. Falls mindestens eine freie Beschriftungsposition existiert, werden die Konflikte mit

den Beschriftungsräumen der anderen Objekte in Abhängigkeit ihrer Klasse bei der Ermittlung der Beschriftungsposition berücksichtigt. Dazu werden Algorithmus 4 und Algorithmus 5 verwendet. In einem zweiten Schritt, der im nächsten Abschnitt behandelt wird, erfolgt eine Feinplatzierung (vgl. auch Abschnitt 4.3.1.1.1)

```

1 Input: statischer Konfliktgraph (passend zum gewählten Maßstab und Ausschnitt).
Output: überlappungsfreie Beschriftungspositionen (soweit möglich).
// Alle Knoten befinden sich immer in einem Status: untreated, unlabeled und labeled
// Initial befinden sich alle Knoten im Status untreated
5 tryItLater =  $\emptyset$  //Liste der Knoten, für die keine konfliktfreie Beschriftungsposition gefunden werden konnte
markiere bei jedem Knoten die Bereiche des Beschriftungsraums als Konflikt, die außerhalb
des Anfragebereichs liegen
bestimme zu jedem Knoten die Beschriftungsschwierigkeit
sortiere Knoten absteigend nach Kombination aus Priorität, Beschriftungsart und -schwierigkeit
10 für jeden Knoten (currentNode) – durchlaufe die sortierte Knoten-Liste:
    setze den Status von currentNode auf labeled
    // berücksichtige (immer) die Beschriftungsbereiche die außerhalb des Anfragebereichs liegen wie
    // Konflikte mit Knoten, die den Status labeled haben
    wähle im Folgenden in Abhängigkeit des Geometrietyps von currentNode (punkt- oder
15 linienhaft) den passenden Algorithmus zur Bestimmung der konfliktfreien Bewegungsräume
    bzw. Bereiche und der Positionsbestimmung
    bestimme konfliktfreie Bereiche freeArea von currentNode zu allen adjazenten Knoten
    wenn einer der freeArea groß genug für konfliktfreie Positionierung
        dann
20         platziere Label in freeArea abhängig von der Positionsqualität
            (→ Algorithmus 11 für punkthafte und Algorithmus 13 für linienhafte)
        sonst
            bestimme konfliktfreie Bereiche remainingFreeArea von
                currentNode zu den adjazenten Knoten, die labeled oder untreated sind
25         wenn einer der remainingFreeArea groß genug für konfliktfreie Positionierung
            dann
                platziere Label in remainingFreeArea unter Berücksichtigung freeArea
                (→ Algorithmus 11 für punkthafte und Algorithmus 13 für linienhafte)
            sonst
30             bestimme konfliktfreie Bereiche labelFreeArea von
                currentNode zu den adjazenten Knoten, die labeled sind
            wenn einer der labelFreeArea groß genug für konfliktfreie Positionierung
                dann
35                 platziere Label in der labelFreeArea unter Berücksichtigung
                    freeArea und remainingFreeArea
                    (→ Algorithmus 11 für punkthafte und Algorithmus 13 für linienhafte)
                sonst //Problem keine freie Beschriftungsposition vorhanden – lokales Minima
                    tryItLater = tryItLater  $\cup$  {adjacentNode}
                    setze den Status von currentNode auf unlabeled
40 wenn Nachbehandlung der im ersten Durchlauf nicht beschrifteten Objekte
    dann
        behandle die Objekte in tryItLater nach (→ Algorithmus 14)
wenn Feinplatzierung
45 dann
    führe Feinplatzierung durch (→ Algorithmus 15 und Algorithmus 16)

```

Algorithmus 10 Der dargestellte Algorithmus führt die eigentliche Beschriftung aus, d.h. die Auswahl einer Beschriftungsposition für jedes zu beschriftende Objekt. Dabei wird ein Hill-Climbing-Verfahren verwendet, das an die Bedürfnisse der Schriftplatzierung und die entwickelten Verfahren adaptiert ist. Bei der Darstellung wurde eine Top-Down-Konstruktion verwendet, in der eine stufenweise Verfeinerung erfolgt.

▪ Algorithmus 10 – Zeile 1 bis 9

Zu Beginn haben alle zu beschriftenden Objekte den Status „unbearbeitet“ (*untreated*) und die Liste *tryItLater* ist initialisiert, die die zu beschriftenden Objekte aufnimmt, für die im ersten Durchlauf des Algorithmus keine konfliktfreie Beschriftungsposition gefunden wird. Wenn noch nicht bei der Ableitung des statischen Konfliktgraphen aus dem reaktiven geschehen,

müssen Bereiche der Beschriftungsräume, die außerhalb des Anfragebereichs, liegen von der Beschriftung ausgeschlossen werden (Abbildung 71). Dies geschieht durch eine Markierung dieser Bereiche als Konflikte.

Anschließend wird vorbereitend für den Hauptteil des Auswahlalgorithmus die Beschriftungsschwierigkeit für jedes zu beschriftende Objekt bestimmt, wobei die zuvor angesprochenen Beschriftungsbereiche außerhalb des Anfragebereichs als nicht beschriftbar berücksichtigt werden (Zeile 8). Dazu werden die Verfahren verwendet, die für die Präprocessingphase in Kapitel 4 entwickelt wurden, Algorithmus 4 (Seite 68) und Algorithmus 5 (Seite 80). Die Bestimmung der Beschriftungsschwierigkeit muss während der Laufzeitphase erneut erfolgen und kann nicht während der Präprocessingphase bestimmt werden, da sie vom gewählten Kartenausschnitt und Kartenmaßstab abhängig ist. Für das Hill-Climbing-Verfahren erfolgt eine (absteigende) Sortierung der zu beschriftenden Objekte (Zeile 9). Der Schlüssel ist eine gewichtete Kombination aus der Wichtigkeit des Objekts, der Beschriftungsart (punkt-, linien- oder flächenhaft) und der zuvor bestimmten Beschriftungsschwierigkeit. Ziel ist die Vorgabe der Beschriftungsreihenfolge für das Hill-Climbing-Verfahren. Wichtigere und schwerer zu beschriftende Objekte werden vor unbedeutenderen oder leichter zu beschriftenden Objekten behandelt in Abhängigkeit ihrer Beschriftungsart.

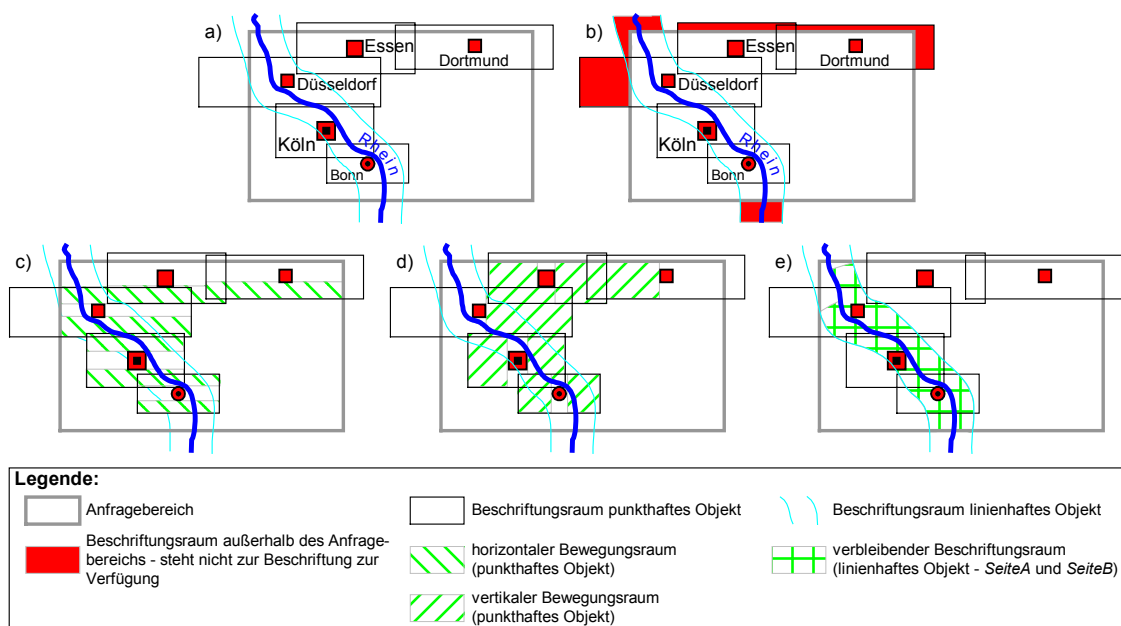


Abbildung 71 Beschriftungsräume, die nicht vollständig innerhalb des Anfragebereichs liegen b), müssen verkleinert werden. Die verbleibenden Beschriftungsräume sind in c) bis e) dargestellt.

▪ Algorithmus 10 – Zeile 10 bis 21

Der Status des aktuell zu beschriftenden Objekts (*currentNode*) wird als bearbeitet und erfolgreich beschriftet markiert (Zeile 11). In dem Fall, dass im Folgenden doch keine konfliktfreie Beschriftungsposition gefunden wird, wird die Markierung dementsprechend geändert (Zeile 39).

Für zu beschriftende Objekte, deren Beschriftungsraum nicht vollständig innerhalb des Anfragebereichs liegt, muss die Einschränkung des Beschriftungsraums, die in Zeile 6 und 7 ermittelt wurde, im Folgenden berücksichtigt werden.

In den Zeilen 17 bis 39 erfolgt eine Einengung des Beschriftungsraum des zur Zeit betrachteten Objekts, um eine konfliktfreie Beschriftungsposition zu erzielen und um später zu be-

handelnde und zu beschriftende Objekte möglichst zu berücksichtigen. Um eine übersichtliche Darstellung zu erreichen, erfolgt eine gemeinsame Darstellung für beide Geometrietypen (punkt- und linienhaft). In den Zeilen 17 bis 39 muss daher abhängig vom Geometrietyp des zur Zeit betrachteten Objekts (*currentNode*) der passende Algorithmus zur Bestimmung der konfliktfreien Bewegungsräume bei punkthaften Objekten bzw. Bereiche bei linienhaften Objekten und der Positionsbestimmung gewählt werden.

Zunächst wird versucht, die schon festgelegten Beschriftungspositionen der Konfliktpartner – Knoten mit dem Status *labeled* – sowie die Beschriftungsräume der noch nicht behandelten Konfliktpartner – Knoten mit dem Status *untreated* – und die erfolglos beschrifteten Konfliktpartner – Knoten mit dem Status *unlabeled* – zu berücksichtigen (Zeile 17 bis 21).

Wenn der aktuell betrachtete Knoten ein zu beschriftendes punkthaftes Objekt ist, werden die konfliktfreien Bewegungsräume ermittelt, die bei der Berücksichtigung der Beschriftungspositionen bzw. -räume der Konfliktpartner verbleiben (Zeile 17 bis 18). Für die Ermittlung der konfliktfreien Bewegungsräume *freeArea* wird Algorithmus 3, der für die Bestimmung der Beschriftungsschwierigkeit entwickelt wurde, verwendet. Er wird dahingehend modifiziert, dass die konfliktfreien Bewegungsräume *value* nicht normiert werden, d.h. die Division durch die Schriftlänge (*labelWidth*) wegfällt. Der Algorithmus liefert die Menge der konfliktfreien Bewegungsräume (*freeArea*) für das aktuell betrachtete zu beschriftende Objekt (*currentNode*) zurück.

Existiert mindestens ein konfliktfreier Bewegungsraum, der groß genug ist, das zu platzierende Schriftobjekt vollständig konfliktfrei aufzunehmen (Zeile 18), dann wird das Schriftobjekt in einem der ausreichend großen konfliktfreien Bewegungsräume platziert (Zeile 20 bis 21). Die Positionierung wird in Abschnitt 5.2.1 mit Algorithmus 11 durchgeführt.

Ist das aktuelle Objekt ein zu beschriftendes linienhaftes Objekt, ist das Vorgehen analog. Dazu wird das Verfahren zur Bestimmung der Beschriftungsschwierigkeit Algorithmus 5 aus Abschnitt 4.3.6 für die Bestimmung der konfliktfreien Bereiche adaptiert. Das Verfahren wird auf die Zeilen 1 bis 24 gekürzt, d.h. die Normierung der konfliktfreien Bewegungsräume und die Bewertungsfunktionen in den Zeilen 25 bis 38 fallen weg.

Wie bei den zu beschriftenden punkthaften Objekten gehen auch hier die Beschriftungspositionen der Konfliktpartner mit dem Status *labeled* ein, während bei den Konfliktpartnern mit dem Status *untreated* und *unlabeled* die Beschriftungsräume berücksichtigt werden.

Führt die Berücksichtigung der Beschriftungspositionen der Konfliktpartner mit dem Status *labeled* und der Beschriftungsräume von Konfliktpartnern mit dem Status *untreated* und *unlabeled* zu einem Erfolg, wird mit Algorithmus 13 in Abschnitt 5.2.2 eine konfliktfreie Beschriftungsposition ausgewählt bzw. bestimmt.

- Algorithmus 10 – Zeile 22 bis 28

Falls bei punkthaften Objekten kein ausreichend großer konfliktfreier Bewegungsraum bzw. bei linienhaften Objekten kein ausreichend großer konfliktfreier Bereich existiert, wird die Anzahl der zu berücksichtigenden Konfliktpartner reduziert. In dieser zweiten Stufe werden die Beschriftungspositionen der Konfliktpartner mit dem Status *labeled* und die Beschriftungsräume der Konfliktpartner, die bisher noch nicht behandelten wurden und den Status *untreated* haben, berücksichtigt (Zeile 23 bis 24). Wie zuvor erfolgt eine Unterscheidung zwischen punkt- und linienhaften Objekten und eine entsprechende Wahl des Verfahrens zur Bestimmung der konfliktfreien Bewegungsräume bzw. Bereiche.

Diese Einschränkung der Anzahl der betrachteten Konfliktpartner reduziert das Konfliktpotential des aktuell betrachteten Objekts. Beschriftungsräume von zu beschriftenden Ob-

jekten, für die aus Mangel an einer konfliktfreien Beschriftungsposition keine Beschriftung erfolgte, werden ausgeklammert. Das bedeutet, dass die Heuristik zu beschriftende Objekte, für die keine Beschriftungsposition gefunden werden konnte und im späteren Verlauf deren Berücksichtigung zu weiteren Problemen führt, nicht berücksichtigt. Die Struktur des in den Zeilen 23 bis 28 beschriebenen Vorgehens ist analog zu dem der ersten Stufe in den Zeilen 17 bis 21.

- Algorithmus 10 – Zeile 29 bis 40

Bringt diese Reduktion keinen Erfolg, d.h. führt diese zu keinem konfliktfreien Bewegungsraum bzw. Bereich, dann werden in der dritten Stufe (Zeile 30 bis 36) ausschließlich die bereits beschrifteten Objekte, d.h. die Konfliktpartner mit dem Status *labeled*, berücksichtigt.

Kann keine konfliktfreie Beschriftungsposition gefunden werden, wenn ausschließlich die bisher beschrifteten Objekte berücksichtigt werden, bleibt das Objekt unbeschriftet und wird in die Liste der unbeschrifteten Objekte (*tryItLater*) gespeichert (Zeile 38) und der Status dementsprechend auf *unlabeled* gesetzt (Zeile 39).

Eine spezielle Behandlung der zu beschriftenden Objekte, für die keine konfliktfreie Beschriftungsposition gefunden werden konnte, erfolgt im Anschluss an das Verfahren.

- Algorithmus 10 – Zeile 41 bis 43

Im Anschluss an die Schleife des Hill-Climbing-Verfahrens kann eine Nachbehandlung der Objekte, für die keine konfliktfreie Beschriftungsposition gefunden wurde, erfolgen. Die Behandlung führt zu einer Verlängerung der Laufzeit und kann jederzeit abgebrochen werden. In Abschnitt 5.2.3 wird ein Verfahren für die Nachbehandlung vorgestellt.

- Algorithmus 10 – Zeile 44 bis 46

Die Minimierung von Informationsverlusten und die Berücksichtigung der topologischen Lage der Schrift können unter Inkaufnahme einer höheren Laufzeit abschließend durchgeführt werden. Abschnitt 5.3 diskutiert dieses Thema.

5.2.1 Platzierung der Beschriftung punkthafter Objekte

Der entwickelte Auswahlalgorithmus in Abschnitt 5.2 (Algorithmus 10) ermittelt für zu beschriftende punkthafte Objekte konfliktfreie Bewegungsräume, in denen diese beschriftet werden sollen (Zeilen 20, 27 und 34 bis 36). Aufbauend auf den freien Bewegungsräumen werden möglichst viele noch nicht berücksichtigte Beschriftungsräume von Konfliktpartnern einbezogen (Algorithmus 11), die jedoch nicht alle berücksichtigt werden können (vgl. Algorithmus 10). Das Durchlaufen aller möglichen Kombinationen aus noch nicht berücksichtigten Beschriftungsräumen mit dem Ziel, eine maximale Anzahl zu berücksichtigen, kommt wegen der hohen Laufzeit nicht infrage. Anstelle dessen wird eine Heuristik entwickelt, die in Abhängigkeit einer Kombination aus Status, Priorität, Beschriftungsart und -schwierigkeit, möglichst viele Objekte berücksichtigt. Nach der erneuten Einengung des Beschriftungsplatzes wird in einem weiteren Schritt (Algorithmus 12) die finale Beschriftung in dem verbleibenden Raum durchgeführt (vgl. Abschnitt 4.3.1.1.1). Für die geometrischen Berechnungen können die im letzten Kapitel entwickelten Verfahren genutzt werden; darauf wird hier nicht weiter eingegangen. Das Verfahren zur Platzierung, das in Algorithmus 11 beschrieben ist, ist an einem Beispiel in Abbildung 72 visualisiert, während Algorithmus 12 in Abbildung 74 skizziert ist.

```

1  Input: zu beschriftendes punkthafte Objekt mit Liste der verbleibenden konfliktfreien
        Beschriftungsräume, die zuvor bestimmt wurden und von denen mindestens einer
        garantiert ausreichend groß ist die Beschriftung konfliktfrei aufzunehmen.
        Eine Liste an Konfliktpartnern, deren Beschriftungsräume soweit möglich noch
5  berücksichtigt werden sollen.
Output: eine Beschriftungsposition für das punkthafte Objekt, die innerhalb eines
        der konfliktfreien Beschriftungsräumen liegt.
// object = zu beschriftendes Objekt; freeArea = Liste an verbleibenden konfliktfreien Bewegungsräumen;
// considerObjects = Liste an noch zu berücksichtigenden Beschriftungsräumen
10 wenn noch nicht behandelte Objekte sortiert werden sollen //Benutzerparameter - Erhöhung der Laufzeit
    dann
        sortiere considerObjects absteigend nach Kombination aus Status (unlabeled/untreated),
        Priorität, Beschriftungsart und -schwierigkeit
für jedes Objekt (currentObject) aus (sortierter Liste) considerObjects:
15 wenn Berücksichtigung des Beschriftungsraums von currentObject bei freeArea von object
        ergibt, dass keine der verbleibenden konfliktfreien Bewegungsräume groß
        genug ist den zu object gehörenden Label konfliktfrei aufzunehmen
        dann
            verlasse schleife
20 sonst
            setze freeArea auf die neuen verbleibenden konfliktfreien Beschriftungsräume, die
            groß genug sind den zu object gehörigen Label konfliktfrei aufzunehmen
für jede freeArea (currentFreeArea):
        ermittle Position mit bester Bewertung in currentFreeArea (Algorithmus 12)
25         und speichere ermittelte Position und Bewertung in labelingPos
        der Eintrag mit der besten Bewertung in der Liste labelingPos ist die gesuchte
        Beschriftungsposition

```

Algorithmus 11 Generierung einer konfliktfreien Beschriftungsposition für punkthafte Objekte (Visualisierung des Verfahrens in Abbildung 72). In einem der zuvor ermittelten konfliktfreien Bewegungsräume *freeArea* wird das zu beschriftende punkthafte Objekt *object* beschriftet. Dabei erfolgt eine Berücksichtigung der Beschriftungsräume der Objekte aus *considerObjects* soweit möglich. (Aufruf aus Algorithmus 10 Zeile 20-21, 27-28 und 34-36)

- Algorithmus 11 – Zeile 1 bis 9

Neben dem zu beschriftenden punkthafte Objekt (*object*) erhält der Algorithmus eine Liste mit den konfliktfreien Bewegungsräumen (*freeArea*). Mindestens einer dieser Bewegungsräume ist groß genug, um die zu platzierende Beschriftung konfliktfrei aufzunehmen. Zusätzlich wird eine Liste von Objekten übergeben, deren Beschriftungsräume, soweit möglich, berücksichtigt werden sollen (*considerObjects*). Die Objekte der Liste wurden im Auswahlalgorithmus Algorithmus 10 entweder noch nicht behandelt (*untreated*) oder es existiert keine konfliktfreie Beschriftungsposition für sie (*unlabeled*) (vgl. Algorithmus 10). Schon beschriftete Objekte (*labeled*) sind in *considerObjects* nicht enthalten. Folglich besitzt keines der Objekte der Liste eine ausgewählte Beschriftungsposition, d.h. wenn sie berücksichtigt werden, dann werden ihre Beschriftungsräume betrachtet. Im Folgenden wird versucht möglichst viele der Objekte zu berücksichtigen.

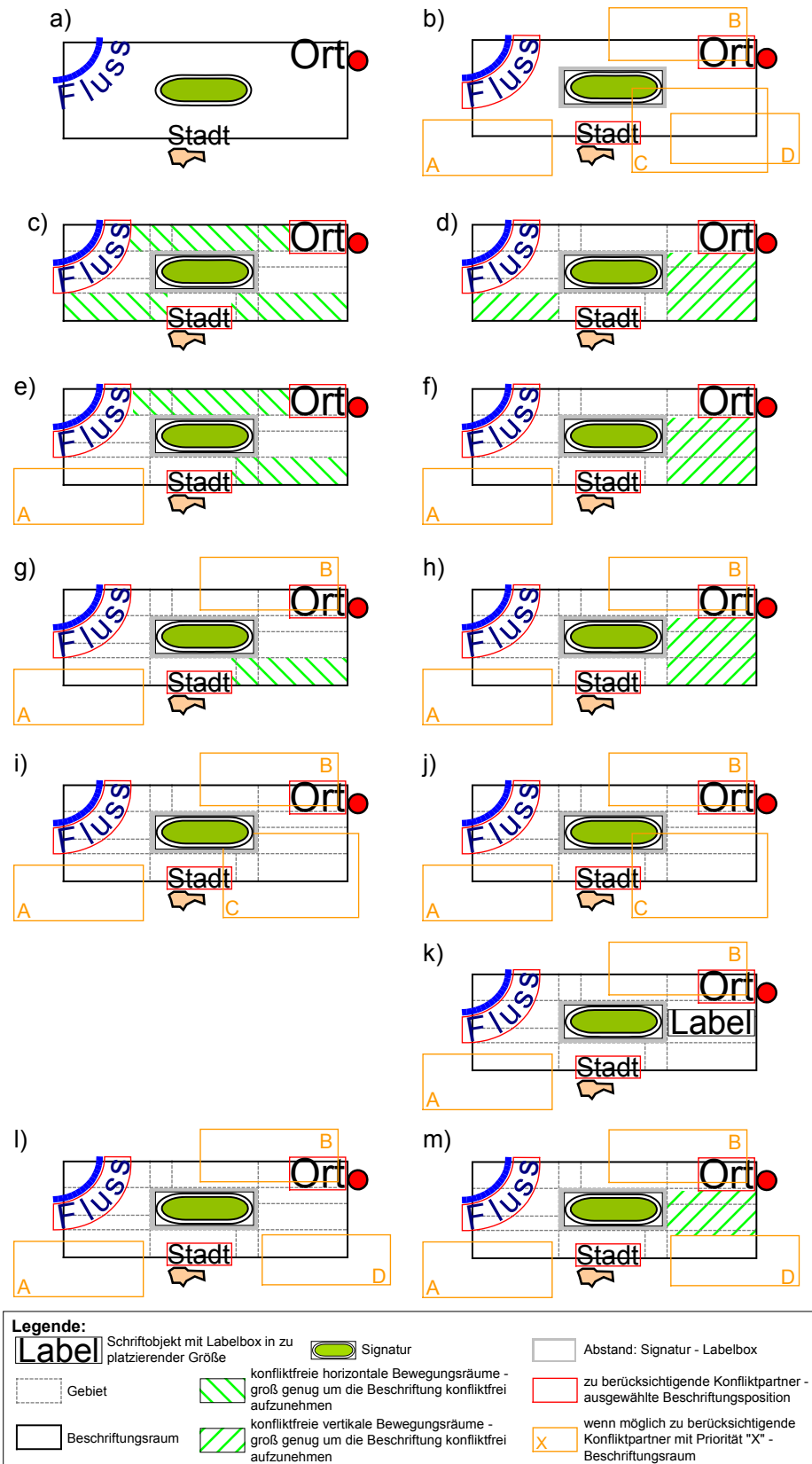


Abbildung 72 Schrittweise Visualisierung von Algorithmus 11: Generierung einer Beschriftungsposition für ein punkthaftes Objekt.

- Algorithmus 11 – Zeile 10 bis 13

Die Sortierung der möglichst zu berücksichtigenden Beschriftungsräume der Objekte aus *considerObjects* wird in Zeile 12 bis 13 in Abhängigkeit einer Gewichtung aus dem Status des Objekts (unlabeled oder untreated), der Priorität, der Beschriftungsart (punkthaft oder linienhaft) und der im Auswahlalgorithmus ermittelten Beschriftungsschwierigkeit gemacht. Die Objekte (*unlabeled*), die nicht erfolgreich beschriftet werden konnten, sollten geringer gewichtet werden, da sie wahrscheinlich wegen ihrem zu großen Konfliktpotential nicht beschriftet werden, genauso wie linienhafte, die wegen ihrer Ausdehnung potentiellen Konflikten ausweichen können, was sich auch in ihrer Beschriftungsschwierigkeit widerspiegelt.

Die Sortierung führt zu einem besseren Gesamtergebnis, jedoch zu Lasten der Gesamtlaufzeit (vgl. Abschnitt 5.2.4). Die Abwägung (Zeile 10) kann auch während der Ausführung des Auswahlalgorithmus geändert werden.

- Algorithmus 11 – Zeile 14 bis 22

In der Schleife in Zeile 14 bis 22 wird versucht, möglichst viele Konfliktpartner, die noch nicht beschriftet wurden, bei der Beschriftung des gerade betrachteten Objekts zu berücksichtigen. Dafür werden die Objekte der Reihenfolge nach aus der (sortierten) Liste *considerObjects* betrachtet und die Änderung der konfliktfreien Bewegungsräumen *freeArea* bei der Berücksichtigung des Beschriftungsraums ermittelt. Führt die Berücksichtigung dazu, dass kein Bewegungsraum verbleibt, der groß genug ist die Beschriftung konfliktfrei aufzunehmen (Zeile 15 bis 17), dann wird dieses Objekt und kein Weiteres mehr berücksichtigt. Die Schleife wird abgebrochen (Zeile 19).

Verbleibt mindestens ein ausreichend großer konfliktfreier Bewegungsraum, dann wird das Objekt bzw. sein Beschriftungsraum berücksichtigt, d.h. er geht in die ausreichend großen konfliktfreien Bewegungsräume *freeArea* ein (Zeile 21 und 22).

Bei einer effizienten Implementierung werden nicht jedes Mal von neuem die konfliktfreien Bewegungsräume berechnet, sondern die Vektoren aus Abschnitt 4.3.3.3 und Abbildung 40 bei jedem neuen Schleifendurchlauf aktualisiert.

- Algorithmus 11 – Zeile 23 bis 27

Die verbliebenen konfliktfreien Bewegungsräume, die mindestens die Größe der zu platzierenden Beschriftung haben, werden durchlaufen (Zeile 23 bis 25). Für jeden wird eine Beschriftungsposition ermittelt, die vollständig in dem betrachteten Bewegungsraum liegt und die beste Bewertung unter den Möglichen ist. Für die Bewertung wird eine gewichtete Kombination aus zwei Kriterien verwendet (Abbildung 73). Die in Abschnitt 2.1.1.2 entwickelte Funktion zur Bestimmung der Beschriftungspositionsbewertung (Gleichung (2.1)) ist das erste Kriterium (Abbildung 73 b) und c)). Als zweites Kriterium geht das Verhältnis des verbleibenden Raums im konfliktfreien Bewegungsraum über- und unterhalb bei vertikalen Bewegungsräumen bzw. links und rechts bei horizontalen Bewegungsräumen ein (Abbildung 73 d) und e)). Ziel ist die Berücksichtigung des freien Raums um Mehrdeutigkeiten bei Beschriftungen zu vermeiden (Algorithmus 12 – Details dazu werden im übernächsten Absatz gegeben) (Abbildung 73 f) und g)). Alternativ können andere Kriterien verwendet oder ergänzt werden, wie z.B. die Berücksichtigung des Informationsverlusts, der durch Verdeckung von Kartenobjekten entsteht.

Die Beschriftungsposition mit der besten Bewertung wird in der Liste *labelingPos* abgelegt. Die Beschriftungsposition mit der besten Bewertung ist das Ergebnis (Zeile 26 bis 27). Die

Speicherung der besten Position kann auch in einer Variablen in der Schleife Zeile 23 bis 25 erfolgen. Dann entfällt die Suche nach dem Maximum in Zeile 26 bis 27.

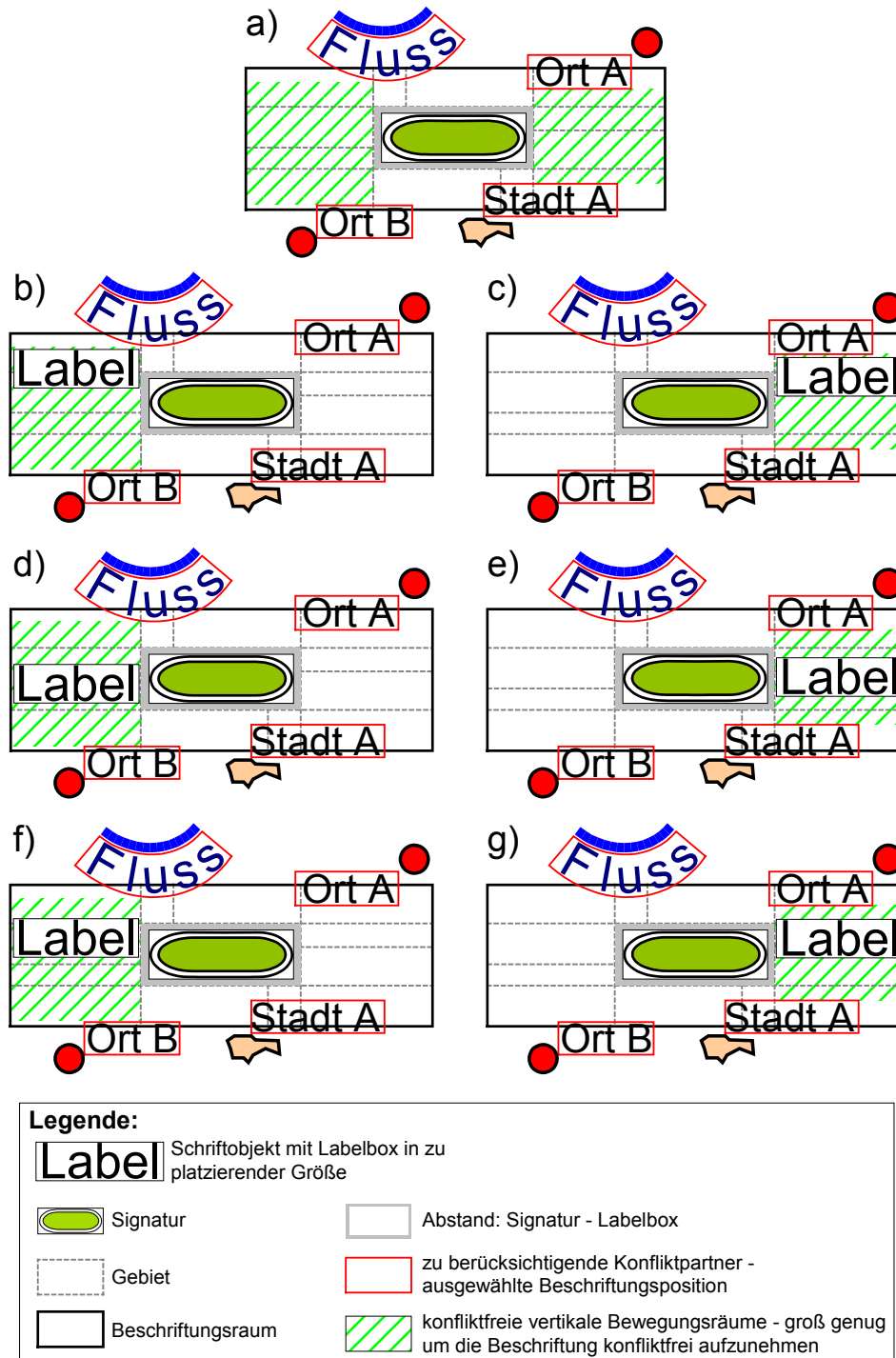


Abbildung 73 Bewertung und Generierung von Beschriftungspositionen mit kombinierten Kriterien: in b) und c) wird die Beschriftungspositionsbewertung Gleichung (2.1) verwendet, wohingegen in d) und e) das Kriterium des Verhältnisses des verbleibenden Raums genutzt wird; f) und g) zeigen eine Kombination der Kriterien – anschaulich rutschen die Beschriftungen in b) und d) bzw. in c) und e) aufeinander zu.

In dem so eingegrenzten Raum wird mit Algorithmus 12 die Beschriftung platziert. Das Verfahren, das in Abbildung 73 skizziert ist, wird im Folgenden beschrieben. Ziel ist, neben

der konfliktfreien Beschriftung, die durch den konfliktfreien Bewegungsraum sichergestellt ist, die Vermeidung von mehrdeutigen Beschriftungen.

```

1  Input: ein konfliktfreier Bewegungsraum eines punkthaften zu beschriftenden Objekts in
    dem die Beschriftung (vollständig) konfliktfrei platziert werden kann.
Output: Beschriftungsposition innerhalb des gegebenen konfliktfreien Beschriftungsraums
    unter Berücksichtigung der Positionsqualität zum zu beschriftenden Objekt und
5   zur Lage innerhalb des konfliktfreien Beschriftungsraums
    // Ermittlung der besten Beschriftungsposition in einem konfliktfreien Bewegungsraum, der groß genug
    // ist eine Beschriftung vollständig konfliktfrei aufzunehmen unter Berücksichtigung der
    // Beschriftungspositionsbewertung und des verbleibenden Raums in dem konfliktfreien Bewegungsraum
wenn vertikaler konfliktfreier Bewegungsraum
10  dann
    // die beste Beschriftungspositionsbewertung ist entweder die höchste Position im Bewegungsraum
    // yMax, die tiefste yMin oder bei einem der fixen Werte
    bestimme yMax und yMin Beschriftungsposition im Bewegungsraum sowie den zugehörigen
    Winkel für die Beschriftungspositionsbewertung
15  bestPos1 ist die Position mit maximum(Beschriftungspositionsbewertung(yMax),
    Beschriftungspositionsbewertung(yMin))
    für jeden fixen Wert der Funktion Beschriftungspositionsbewertung deren Winkel
    innerhalb der Winkel von yMax und yMin liegen (fixValue):
    bestPos1 ist die Position mit maximum(Beschriftungspositionsbewertung(bestPos1),
20  Beschriftungspositionsbewertung(fixValue))
    // die beste Position, wenn nur der konfliktfreie Bewegungsraum betrachtet wird, ist genau in
    // der (vertikalen) Mitte
    bestPos2 = ( yMax - yMin ) / 2 + yMin // in der Mitte zwischen yMax und yMin
    // Kombination der beiden (gewichteten) Kriterien ergibt die endgültige Platzierungsposition
25  bestimme resultPos abhängig von bestPos1 und bestPos2 sowie deren Gewichtungen
sonst // horizontaler konfliktfreier Bewegungsraum
    // die beste Beschriftungspositionsbewertung ist entweder die am weitesten links gelegene Posi-
    // tion im Bewegungsraum xMin, die am weitesten rechts gelegene xMax oder bei einem fixen Wert
    bestimme xMin und xMax Beschriftungsposition im Bewegungsraum sowie den zugehörigen
    Winkel für die Beschriftungspositionsbewertung
30  bestPos1 ist die Position mit maximum(Beschriftungspositionsbewertung(xMin),
    Beschriftungspositionsbewertung(xMax))
    für jeden fixen Wert der Funktion Beschriftungspositionsbewertung deren Winkel
    innerhalb der Winkel von xMin und xMax liegen (fixValue):
35  bestPos1 ist die Position mit maximum(Beschriftungspositionsbewertung(bestPos1),
    Beschriftungspositionsbewertung(fixValue))
    // die beste Position, wenn nur der konfliktfreie Bewegungsraum betrachtet wird ist genau in
    // der (horizontalen) Mitte
    bestPos2 = ( xMax - xMin ) / 2 + xMin // in der Mitte zwischen xMax und xMin
40  // Kombination der beiden (gewichteten) Kriterien ergibt die endgültige Platzierungsposition
    bestimme resultPos abhängig von bestPos1 und bestPos2 sowie deren Gewichtungen

```

Algorithmus 12 Bestimmung der besten Beschriftungsposition in einem konfliktfreien Bewegungsraum, in den die gewichtete Kombination aus der entwickelten Beschriftungspositionsbewertung (Gleichung (2.1)) aus Abschnitt 2.1.1.2 und der besten Position innerhalb des Raumes eingeht. Das Verfahren unterscheidet zwischen vertikalem (Zeile 11 bis 25) und horizontalem Beschriftungsraum (Zeile 27 bis 41). Das Ergebnis ergibt sich aus den gewerteten für die beiden Kriterien abgeleiteten Position (vgl. Abbildung 74). (Aufruf aus Algorithmus 11 Zeile 24)

- Algorithmus 12 – Zeile 1 bis 9

Das Verfahren setzt voraus, dass der konfliktfreie Bewegungsraum ausreichend groß ist, um die Beschriftung konfliktfrei aufzunehmen. Es erfolgt eine Unterscheidung zwischen vertikalen und horizontalen Bewegungsräumen (Zeile 9).

- Algorithmus 12 – Zeile 10 bis 25

Zunächst wird die beste Beschriftungsposition mit der Beschriftungspositionsbewertung (Gleichung (2.1) aus Abschnitt 2.1.1.2) bestimmt (*bestPos1*). Ausgenutzt wird dabei die Monotonie der Funktion sowie deren Sprungstellen, die fixen Werte. Die beste Position ist daher entweder am oberen oder am unteren Rand des Bewegungsraums oder bei einem der

festen Werte der Funktion. Fiktiv wird jeweils eine Beschriftungsposition soweit oben wie möglich und soweit unten wie möglich im konfliktfreien Bewegungsraum angenommen ($yMax$ und $yMin$). Zu diesen werden die Winkel und die Beschriftungspositionsbewertungen bestimmt (Zeile 13 und 14). Die Position mit der besseren Bewertung wird zunächst als beste Position ($bestPos1$) angenommen (Zeile 15 und 16). Liegen innerhalb des betrachteten konfliktfreien Beschriftungsraums fixe Werte der Bewertungsfunktion, werden diese Stellen als weitere fiktive Positionen angenommen (Zeile 17 bis 20) (Abbildung 74 a)). Die Position mit der besten Bewertung ($bestPos1$) ist die Position für das erste Kriterium (Zeile 19 bis 20) (Abbildung 74 c) – schwarze Position).

Bei dem zweiten Kriterium ist die beste Position ($bestPos2$) die, die in der Mitte des konfliktfreien Bewegungsraums liegt (Zeile 21 bis 23) (Abbildung 74 b)).

Die endgültige Beschriftungsposition wird abhängig von den beiden ermittelten Positionen und deren Gewichtung bestimmt (Zeile 24 bis 25). In Algorithmus 12 wird absichtlich kein konkretes Kriterium genannt. Als endgültige Beschriftungsposition könnte z.B. die Position gewählt werden, die genau zwischen $bestPos1$ und $bestPos2$ liegt oder in eindrittel Entfernung zu $bestPos1$ und zweidrittel Entfernung zu $bestPos2$ (Abbildung 74 c))

▪ Algorithmus 12 – Zeile 26 bis 41

Für die Bestimmung der Beschriftungsposition in vertikalen konfliktfreien Bewegungsräumen gilt das Verfahren analog, mit dem Unterschied, dass anstelle der y-Werte die x-Werte betrachtet werden. Auf eine Erläuterung wird wegen der engen Analogie verzichtet.

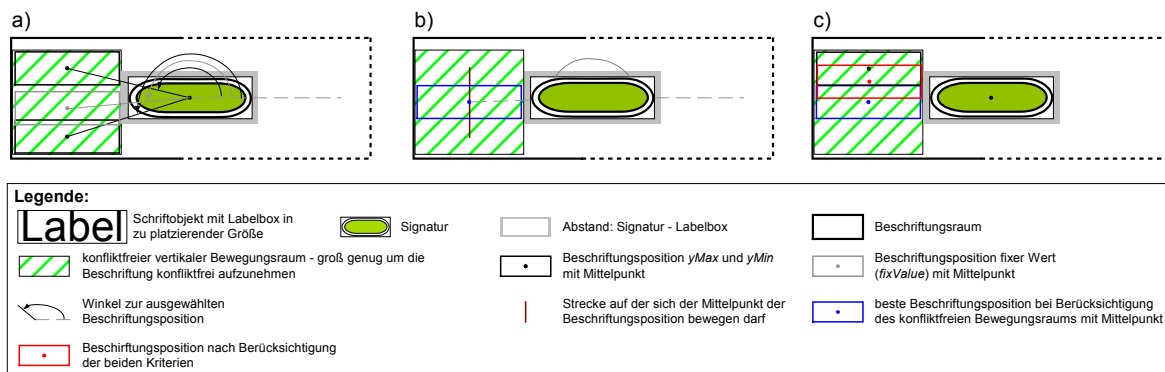


Abbildung 74 Die Abbildung illustriert das Verfahren zur Ermittlung der Beschriftungsposition aus Algorithmus 12 an einem vertikalen konfliktfreien Bewegungsraum.

5.2.2 Platzierung der Beschriftung linienhafter Objekte

Analog zu der Behandlung von zu beschriftenden punkthaftern Objekten werden im Auswahlalgorithmus in Abschnitt 5.2 (Algorithmus 10) für zu beschriftende linienhafte Objekte konfliktfreie Bewegungsräume, in denen diese beschriftet werden sollen, ermittelt (Zeile 20, 27 und 34 bis 36). Die Motivation, Vorgehensweise und Verfahren für die Bestimmung der Beschriftungsposition zu beschriftender linienhafter Objekte sind analog zu denen für punkthafte Objekte in Abschnitt 5.2.1 und Algorithmus 11 und Algorithmus 12. Aus Gründen der Vollständigkeit ist das Verfahren in Algorithmus 13 dargestellt. Die Erläuterung ist analog zu der für punkthafte Objekte in Abschnitt 5.2.1, wobei der Ausdruck „konfliktfreier Bewegungsraum“ durch „konfliktfreier Bereich“ zu ersetzen ist. Das Verfahren zur Bestimmung der besten Beschriftungsposition ist analog zu der für punkthafte Objekte (Algorithmus 12), d.h. eine gewichtete Kombination aus Beschriftungspositionsbewertung aus Abschnitt 2.1.1.3

und der entwickelten Funktionen sowie dem Verhältnis des verbleibenden Raums im konfliktfreien Bereich davor und dahinter.

```

1  Input: zu beschriftendes linienhaftes Objekt mit Liste der verbleibenden konflikt-
    freien Bereiche, die zuvor bestimmt wurden, von denen mindestens einer
    garantiert ausreichend groß ist, die Beschriftung konfliktfrei aufzunehmen.
5  Eine Liste an Konfliktpartnern, deren Beschriftungsräume wenn möglich noch
    berücksichtigt werden sollen.
    Output: eine Beschriftungsposition für das linienhafte Objekt, die innerhalb eines
    der konfliktfreien Bereiche.
    // object = zu beschriftendes Objekt; freeArea = Liste an verbleibenden konfliktfreien Bereiche;
    // considerObjects = Liste an noch zu berücksichtigenden Beschriftungsräumen
10 wenn noch nicht behandelte Objekte sortiert werden sollen //Benutzerparameter-Erhöhung der Laufzeit
    dann
        sortiere considerObjects absteigend nach Kombination aus Status (unlabeled/untreated),
            Priorität, Beschriftungsart und -schwierigkeit
    für jedes Objekt aus (sortierter Liste) considerObjects (currentObject):
15    wenn Berücksichtigung des Beschriftungsraums von currentObject bei freeArea von
        object ergibt, dass keine der verbleibenden konfliktfreien Bereiche groß
            genug ist den zu object gehörenden Label konfliktfrei aufzunehmen
        dann
            verlasse schleife
20    sonst
        setze freeArea auf die neuen verbleibenden konfliktfreien Bereiche, die
            groß genug sind den zu object gehörigen Label konfliktfrei aufzunehmen
    für jede freeArea (currentFreeArea):
        ermittle Position mit bester Bewertung in currentFreeArea
25        und speichere ermittelte Position und Bewertung in labelingPos
        der Eintrag mit der besten Bewertung in der Liste labelingPos ist die gesuchte
            Beschriftungsposition

```

Algorithmus 13 Erzeugung einer konfliktfreien Beschriftungsposition für zu beschriftende linienhafte Objekte (*object*). Aus Algorithmus 10 erfolgt der Aufruf des Verfahrens mit den konfliktfreien Bereichen *freeArea* sowie den Objekten mit den möglichst zu berücksichtigenden Beschriftungsräumen als Parameter. Das Verfahren ist analog zu dem für zu beschriftende punkthafte Objekte Algorithmus 11. Der Unterschied liegt in der Bestimmung der konfliktfreien Bereiche und der Positionsbestimmung in Zeile 24 bis 25. (Aufruf aus Algorithmus 10 Zeile 20-21, 27-28 und 34-36)

5.2.3 Nachbehandlung nicht beschrifteter Objekte

Der Auswahlalgorithmus, der in Abschnitt 5.2 Algorithmus 10 entwickelt wurde, versucht, möglichst viele zu beschriftende Objekte bei der Beschriftung zu berücksichtigen. Falls doch einige Objekte nicht beschriftet werden können, werden diese in der Liste *tryItLater* gesammelt.

Für diese Objekte kann in Abhängigkeit von ihrer Priorität mit dem hier vorgestellten Algorithmus 14 versucht werden, dennoch eine konfliktfreie Beschriftungsposition zu ermitteln. Dazu wird eine laufzeitaufwendige, beschränkte Tiefensuche mit Tiefenschranke (*maxTiefe*) durchgeführt, wobei pro beschriftetem Objekt maximal *maxBreite* Konfliktpartner versucht werden, neu zu beschriften. Der Parameter *tiefe* in Algorithmus 14 zählt dabei die Tiefe der Rekursion und dient somit als Tiefenbeschränkung bzw. Abbruchkriterium. Bei dem ersten Aufruf wird die Tiefe auf Null gesetzt. Für jedes aufgerufene Objekt werden die Konfliktpartner ermittelt und nach Nachbarschaft sortiert (über den Winkel). Abhängig davon, wie viele Konfliktpartner geändert werden dürfen (*maxBreite*), werden dementsprechende Beschriftungssituationen getestet. Führt das Weglassen von Beschriftungen von Konfliktpartnern zu einer möglichen Beschriftung, wird versucht, diese an anderer Stelle neu zu beschriften. Dazu wird das Verfahren auf diese erneut angewendet, d.h. es erfolgt ein rekursiver Aufruf. Die Rekursion bricht ab, wenn entweder die maximale Tiefe erreicht ist oder eine konfliktfreie Beschriftung ohne Neubeschriftung weiterer Objekte erreicht ist. Erst

wenn insgesamt eine Beschriftungslösung gefunden wird, d.h. wenn für das ursprüngliche Objekt und alle geänderten Objekte eine konfliktfreie Beschriftung existiert, wird die Beschriftung „realisiert“. Die Objekte werden tatsächlich an den neu berechneten Positionen beschriftet und die Markierung entfernt.

Falls Algorithmus 10 zu keiner zufriedenstellenden Beschriftung führt, ist bei der Generierung des reaktiven Konfliktgraphen die maximal zulässige Beschriftungsschwierigkeit zu ändern. Dadurch wird der gewünschte Effekt ohne eine Erhöhung der Laufzeit in der Interaktionsphase erzielt. Ist die Anzahl der zu beschriftenden Objekte geringer, so ist die Wahrscheinlichkeit, dass alle konfliktfrei beschriftet werden können, größer.

```

1  Input: ein nachzubehandelndes bisher nicht beschriftetes Objekt sowie zur Beschränkung
      der Tiefensuche in der Nachbehandlung eine maximale Rekursionstiefe (maxTiefe)
      und eine maximale Anzahl an zu gleich betrachteten Konflikten (maxBreite).
Output: eine konfliktfreie Beschriftung des nachzubehandelnden Objekts und notwendige
5  Replatziierung der Schriftpositionen von umliegenden Objekten wenn möglich.
      // Rekursive Funktion zur Nachbehandlung eines nicht beschrifteten Objekts
      // tiefe ist ein Parameter, der beim Aufruf übergeben wird und die Rekursionstiefe zählt
      // maxBreite gibt die maximale Breite und maxTiefe die maximale Tiefe der Suche an
      // replatziert ist eine boolesche Variable, die als Markierung der neuplatzierten Objekte dient um
10  // keinen Knoten mehrfach neu zu beschriften
      für jede inzidente Kante (incidentEdge) bzw. adjazenten Knoten (adjacentNode):
          wenn adjacentNode ist labeled oder replatziert
              dann
                  bestimme Winkel zwischen Mittelpunkt des betrachteten Objekts und Mittelpunkt der
15  Beschriftungsposition von adjacentNode
                  füge Tupel (Winkel, adjacentNode) der Liste conflictPartner hinzu
      sortiere Liste conflictPartner nach Winkel
      wenn tiefe == 0 // initialer erster Aufruf
          dann start = 1
20  sonst start = 0
      rückgabewert = false
      für i = start und i < maxBreite bei jedem Schleifendurchlauf erhöhe i um 1:
          betrachte i benachbarte Konflikte in conflictPartner in denen keiner
25  als replatziert markiert ist
          wenn eine konfliktfreie Beschriftung ohne diese i Konflikte möglich
              dann
                  markiere und platziere Beschriftungsposition in dem erzeugten freien Raum
                  markiere Knoten als replatziert
                  wenn i > 0 und tiefe < maxTiefe
30  dann
                      versuche diese i Objekte wo anders zu beschriften - rekursiver Aufruf mit tiefe+1
          wenn alle (rekursiven) Aufrufe true ergeben oder i = 0
              dann
                  wenn tiefe == 0 // initialer erster Aufruf
35  dann
                      realisiere die Replatziierung bei allen als replatziert markierten Objekten
                      entferne die Markierung replatziert von den untersuchten Objekten
                      rückgabewert = true // es wurde eine erfolgreiche Replatziierung gefunden
                      fertig - beende die Schleife
40  sonst
                      verwerfe Replatziierung d.h. entferne Markierung replatziert
      gib rückgabewert zurück

```

Algorithmus 14 Nachbehandlung von Objekten, die nicht mit Algorithmus 10 beschriftet werden konnten (Algorithmus 10 *tryItLater*). Dazu wird eine beschränkte Tiefensuche mit Tiefenschranke (*maxTiefe*) durchgeführt, wobei pro beschriftetem Objekt maximal *maxBreite* Konfliktpartner versucht werden neu zu beschriften. (Aufruf aus Algorithmus 10 Zeile 42)

5.2.4 Laufzeitanalyse

Der Auswahlalgorithmus (Algorithmus 10) wird zur Laufzeitanalyse in drei Teile zerlegt: Zeile 1 bis 9 (Initialisierung); Zeile 10 bis 40 (Kern) und Zeile 41 bis 46 (Nachbehandlung und Feinplatzierung). Die im Folgende angenommene Anzahl n der zu beschriftenden Objekte ist gleich der Anzahl der Knoten des statischen Konfliktgraphen und ist im allgemeinen kleiner als die Gesamtanzahl der zu beschriftenden Objekte aus Kapitel 4, die Anzahl der Knoten des reaktiven Konfliktgraphen, da der statische Konfliktgraph ein „ausgedünnter“ reaktiver Konfliktgraph ist.

Im Folgenden sei $n \equiv$ Anzahl der zu beschriftenden Objekte; $m \equiv$ maximale Anzahl der Konfliktpartner zum zu beschriftenden Objekt – entspricht dem maximalen Knotengrad des statischen Konfliktgraphen.

- Laufzeitanalyse Auswahlalgorithmus (Algorithmus 10) – Zeile 1 bis 9 – Initialisierung:

Die Ermittlung der zu beschriftenden Objekte, deren Beschriftungsräume nicht vollständig innerhalb des Anfragebereichs liegen (Algorithmus 10 – Zeile 6 bis 7) und der nötigen Verkleinerung ihrer Beschriftungsräume, kann mit $O(n)$ nach oben abgeschätzt werden. Dies kann auch in der Extraktionsphase des statischen aus dem reaktiven Konfliktgraphen durchgeführt werden oder mit der Bestimmung der Beschriftungsschwierigkeit in der nächsten Zeile. Für die hier betrachtete Laufzeitabschätzung führt das zu keiner Änderung. Die Bestimmung der Beschriftungsschwierigkeit für jedes zu beschriftende Objekt verursacht im Worst-Case Kosten von $O(n*m)$. In Anlehnung an die Diskussion von 4.3.11.1 wird angenommen, dass die maximale Anzahl der Liniensegmente und die maximale Anzahl der Gebiete als konstant angesehen werden können. Die Sortierung der zu beschriftenden Objekte, für die Reihenfolge der Beschriftung in Algorithmus 10 – Zeile 9 kann mit $O(n \log n)$ nach oben abgeschätzt werden.

Insgesamt ergibt sich somit eine obere Schranke von $O(n \log n + n * m)$ für die Initialisierung (Algorithmus 10 – Zeile 1 bis 9).

- Laufzeitanalyse Auswahlalgorithmus (Algorithmus 10) – Zeile 10 bis 40 – Kern:

Der Kern des Auswahlalgorithmus besteht aus einer äußeren for-Schleife (Algorithmus 10 – Zeile 10 bis 40), die n -Mal durchlaufen wird ($O(n)$).

Da die maximale Anzahl an Gebieten und Liniensegmenten als konstant angesehen wird und die in der Unterscheidung aufgerufenen Verfahren gleiche Laufzeiten besitzen, reicht bis auf Ausnahmen die Betrachtung des punkthaften Falls (Algorithmus 10 – Zeile 14 bis 40).

Die Bestimmung der konfliktfreien Bewegungsräume kann nach oben mit $O(m)$ für jedes zu beschriftende Objekt abgeschätzt werden (Zeile 17, 23-24 und 30-31). Wenn die Bestimmung der konfliktfreien Bewegungsräume in der verschachtelten if-Anweisung von Algorithmus 10 Zeile 17 bis 39 so implementiert wird, wie dargestellt, werden die Konfliktflächen für die schon platzierten Konfliktpartner bis zu dreimal (Zeile 17, 23-24 und 30-31) und die für die noch nicht behandelten Konfliktpartner bis zu zweimal (Zeile 17 und 23-24) berechnet. Bei der Implementierung ist es daher sinnvoller, die Konfliktpartner stufenweise zu berücksichtigen – erst die Knoten mit dem Status *labeled*, dann *untreated* und schließlich *unlabeled* – und die Zwischenergebnisse zu speichern. Für die Laufzeitanalyse, die Abschätzung des Worst-Cases, ergibt das keine Änderung und dieser bleibt bei $O(m)$.

Die Erkennung, ob ein konfliktfreier Bewegungsraum ausreichend groß ist, um eine Beschriftung konfliktfrei aufzunehmen, (Zeile 18, 25 und 32) kann für punkthafte Objekte mit

konstanter Zeit $O(1)$ und für linienhafte Objekte mit der Anzahl der Konfliktpartner $O(m)$ abgeschätzt werden. Bei punkthaften Objekten ist die Anzahl der konfliktfreien Bewegungsräume durch die Anzahl der Gebiete (vgl. Abschnitt 4.3.3.2 und 4.3.11.1) beschränkt, die als konstant betrachtet wird. Für linienhafte zu beschriftende Objekte kann die Anzahl an konfliktfreien Bereichen durch die (maximale) Anzahl der Konfliktpartner nach oben mit $O(m)$ abgeschätzt werden. Damit können die if-Anweisungen aus Algorithmus 10 – Zeile 18, 25 und 32 für den Worst-Case insgesamt mit $O(m)$ abgeschätzt werden.

Wie im Folgenden hergeleitet, verursacht die Ermittlung der Beschriftungsposition (Zeile 20-21, 27-28 und 34-36) Kosten im Worst-Case von $O(m \log m)$.

Die Kosten innerhalb der for-Schleife können mit $O(m + m \log m) = O(m \log m)$ nach oben abgeschätzt werden. Dadurch ergibt sich insgesamt eine Worst-Case Laufzeit für die gesamte for-Schleife in Algorithmus 10 – Zeile 10 bis 40 von $O(n * m \log m)$.

- Laufzeitanalyse Ermittlung Beschriftungsposition (Algorithmus 11 und Algorithmus 13):

Für das schon im letzten Abschnitt vorweggenommene Ergebnis, die Laufzeit der Ermittlung der Beschriftungsposition, d.h. die Platzierung der Beschriftung, (Algorithmus 10 – Zeile 20-21, 27-28 und 34-36) wird Algorithmus 11 bzw. Algorithmus 13, die der Auswahlalgorithmus (Algorithmus 10) aufruft, untersucht. Dort werden die zu berücksichtigenden Konfliktpartner zunächst sortiert (Algorithmus 11 bzw. Algorithmus 13 – Zeile 12 bis 13). Da es maximal m Konfliktpartner gibt, kann die Laufzeit mit $O(m \log m)$ nach oben abgeschätzt werden. Die Schleife zur Berücksichtigung weiterer Beschriftungsräume aus der zuvor sortierten Liste wird im Worst-Case $O(m)$ -Mal durchlaufen (Algorithmus 11 bzw. Algorithmus 13 – Zeile 14 bis 22). Die if-Anweisung (Algorithmus 11 bzw. Algorithmus 13 – Zeile 15 bis 19) verursacht konstante Kosten, d.h. sie kann mit $O(1)$ nach oben abgeschätzt werden. Für die Bewerksstellung in $O(1)$ müssen den Algorithmen die zuvor in Algorithmus 10 ermittelten konfliktfreien Bewegungsräume bzw. Bereiche übergeben werden, so dass bei jedem Schleifendurchlauf nur das hinzukommende Objekt einbezogen werden muss. Die so berechneten konfliktfreien Bewegungsräume bzw. Bereiche werden in Zeile 21 bis 22 gespeichert, sie können also ebenfalls mit $O(1)$ nach oben abgeschätzt werden.

Für jeden verbleibenden konfliktfreien Bewegungsraum bzw. Bereich wird die beste Beschriftungsposition ermittelt und bewertet. Für punkthafte Objekte geschieht dies mit Algorithmus 13 für linienhafte Objekte analog mit gleicher Laufzeit. Die Kosten sind konstant und können somit nach oben mit $O(1)$ abgeschätzt werden. Da maximal $O(m)$ konfliktfreie Bereiche existieren können, bei punkthaften Objekten ist die Laufzeit abhängig von der Anzahl der Gebiete und somit als konstant anzusehen, kann die for-Schleife (Zeile 23 bis 25) insgesamt mit $O(m)$ nach oben abgeschätzt werden. Die Ermittlung der besten Position (Zeile 26 bis 27) kann ebenfalls mit $O(m)$ nach oben abgeschätzt werden, effizienter ist die Speicherung der besten Beschriftungsposition innerhalb der for-Schleife Zeile 23 bis 25, wodurch keine zusätzlichen Laufzeitkosten anfallen.

Insgesamt kann die Worst-Case-Laufzeit für Algorithmus 11 und Algorithmus 13 mit $O(m \log m)$ bestimmt werden.

- Laufzeitanalyse Auswahlalgorithmus (Algorithmus 10) – Zeile 41 bis 46 – Nachbehandlung und Feinplatzierung:

Für die Nachbehandlung von Objekten, für die bisher keine konfliktfreie Beschriftungsposition gefunden werden konnte (Algorithmus 10 – Zeile 43), wird der in Abschnitt 5.2.3 vorgestellte Algorithmus 14 verwandt. Die Laufzeit des Verfahrens ist abhängig von der

Wahl der Parameter für die maximale Tiefe ($maxTiefe$) und der Anzahl der Konfliktpartner, die auf einmal neu beschriftet werden dürfen ($maxBreite$). Wie in dem zugehörigen Abschnitt beschrieben, verursacht das rekursive Verfahren hohe Laufzeiten. Für jedes noch zu beschriftende Objekt kann die Anzahl der Aufrufe des Verfahrens mit $O(maxTiefe * maxBreite^{maxTiefe})$ nach oben abgeschätzt werden.

Abgesehen von dem rekursiven Aufruf und den Zeilen, die konstante Kosten ($O(1)$) verursachen (Zeile 18-21 und Zeile 42), lässt sich Algorithmus 14 für die Laufzeitanalyse in drei Teile zerlegen: Vorbereitung Zeile 11 bis 16; Sortierung der Konfliktpartner Zeile 17; Positionssuche Zeile 22 bis 41.

Die Vorbereitung (Algorithmus 14 – Zeile 11 bis 16) für die Sortierung der Konfliktpartner kann nach oben mit $O(m)$ und die nachfolgende Sortierung (Algorithmus 14 – Zeile 17) mit $O(m \log m)$ abgeschätzt werden.

Die Kombination aus for-Schleife und benachbarten Konfliktpartnern (Algorithmus 14 – Zeile 22 bis 25) verursacht im Worst-Case $O(maxBreite * m)$ Durchläufe. Die Operationen bzw. Anweisungen innerhalb der Schleife (Algorithmus 14 – Zeile 25 bis 41) sind, abgesehen von dem rekursiven Aufruf, konstant und können mit $O(1)$ nach oben abgeschätzt werden.

Insgesamt ergibt sich somit für Algorithmus 14 für jedes nachzubehandelnde Objekt eine obere Schranke von $O((m \log m + maxBreite * m) * maxTiefe * maxBreite^{maxTiefe})$. Wie erwähnt sollte eine Nachbehandlung nur in einem kleinen Bereich (kleine Werte für $maxTiefe$ und $maxBreite$) durchgeführt werden, wegen des exponentiellen Charakters des Verfahrens und weil eine Beschriftung zusätzlicher Objekte nicht garantiert werden kann.

Die Feinplatzierung jedes zu beschriftenden Objekts, für das eine konfliktfreie Beschriftungsposition ermittelt wurde, wird im nachfolgenden Abschnitt behandelt und kann je Objekt mit $O(m)$ nach oben abgeschätzt werden (m ist die Anzahl der konfliktfreien Bereiche eines linienhaften Objekts). Die Anzahl der konfliktfreien Gebiete punkthafter Objekte ist von der maximalen Anzahl der Gebiete, die als konstant betrachtet wird, abhängig. Somit ergibt sich eine Worst-Case Abschätzung von $O(m * n)$ für die Feinplatzierung, im schlechtesten Fall sind alle n Objekte beschriftet.

- Laufzeitanalyse Auswahlalgorithmus (Algorithmus 10) – Zusammenfassung:

Insgesamt ergibt sich somit eine Laufzeitabschätzung nach oben für die Interaktionsphase – den Auswahlalgorithmus bis auf die Nachbehandlung nicht beschrifteter Objekte (Algorithmus 10 – Zeile 1 bis 40 und 44 bis 46) – von $O(n \log n + n * m \log m)$. Im Worst-Case ist die Anzahl m der Konfliktpartner gleich der Anzahl der zu beschriftenden Objekte n . Somit ergibt sich eine obere Schranke von $O(n^2 \log n)$.

Dies ist eine sehr pessimistische Einschätzung der Anzahl der Konfliktpartner. Wenn man die Anzahl der Konfliktpartner als konstant betrachten könnte, was aber nur durch eine Modifikation der Präprocessingphase realisiert werden kann, könnte die Laufzeit sogar mit $O(n \log n)$ abgeschätzt werden. Die Präprocessingphase und die Wahl und Parametrisierung des Deselektionskriteriums (Ausdünnung der zu beschriftenden Objekte) sind für die Laufzeit der Beschriftung in der Interaktionsphase entscheidend.

Die Nachbehandlung von nicht beschrifteten Objekten ist sehr zeitaufwendig und sollte nur in Ausnahmefällen oder mit einem Zeitabbruchkriterium durchgeführt werden, sie lässt sich pro

zu behandelndem Objekt mit $O((m \log m + \text{maxBreite} * m) * \text{maxTiefe} * \text{maxBreite}^{\text{maxTiefe}})$ nach oben abschätzen.

5.3 Feinplatzierung

Dem Auswahlalgorithmus kann, wie in Abschnitt 5.2 beschrieben, eine Feinplatzierung nachgeschaltet werden. Die Beschriftungspositionen der durch den Auswahlalgorithmus ausgewählten und konfliktfrei beschrifteten Objekte können, wie im Folgenden beschrieben, umplatziert werden, um eine lokal verbesserte Beschriftung zu erzielen. Für die Feinplatzierung werden zusätzliche kartographische Gestaltungsregeln berücksichtigt: die topologisch korrekte Lage der Schrift zum zu beschriftenden Objekt im nächsten Abschnitt und die Berücksichtigung von Informationsverlusten im übernächsten Abschnitt.

5.3.1 Topologisch korrekte Lage von Schrift zum zu beschriftenden Objekt

Die Beschriftung punkthafter Objekte sollte wenn möglich korrekt in Bezug auf Trennlinien erfolgen. In Abschnitt 4.7 wurde dieser Themenkomplex in Abbildung 66 an Beispielen dargestellt und die Integration in den reaktiven Konfliktgraphen behandelt. Die nötigen Informationen für die Berücksichtigung der topologisch korrekten Lage der Schrift wurde für jedes zu beschriftende Objekt während der Erzeugung des reaktiven Konfliktgraphen in Algorithmus 9 gesammelt und lokal, in Abhängigkeit vom Maßstab, in dem die Trennlinie beachtet werden muss, am zu beschriftenden Objekt gespeichert.

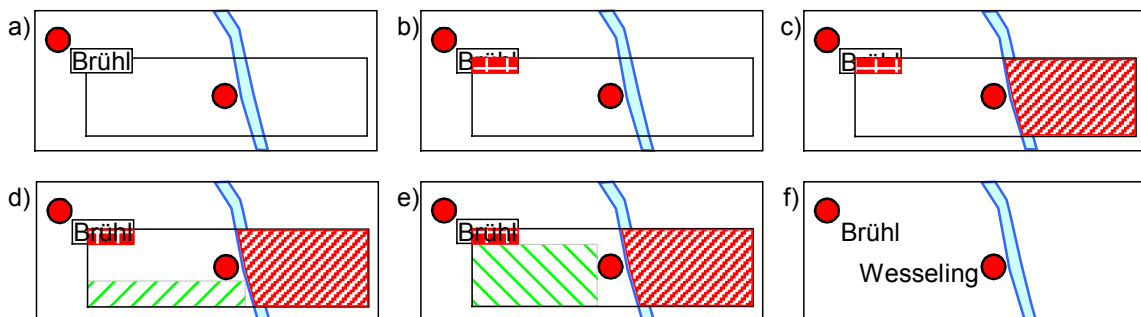


Abbildung 75 Topologisch korrekte Beschriftung von punkthafter Objekten in Bezug auf Trennlinien. Wesseling liegt links vom Rhein und soll (wenn möglich) auf der topologisch korrekten linken Rheinseite beschriftet werden.

In Algorithmus 15 und Abbildung 75 ist das Verfahren zur Berücksichtigung der topologisch korrekten Lage der Schrift während der Interaktionsphase dargestellt.

In der Präprocessingphase in Abschnitt 4.7 in Algorithmus 9 sind die für die topologische Beschriftung zu berücksichtigenden Objekte mit dem Maßstab, ab dem diese relevant werden (*cutScale*), in der Liste *topology* an jedem punkthafter Objekt gespeichert. Diese werden der Reihenfolge nach auf Relevanz getestet und es wird versucht, diese zu berücksichtigen (Algorithmus 15 – Zeile 13 bis 15), wenn dies erforderlich ist. Dazu wird die Fläche des Beschriftungsraums ermittelt, in der keine Beschriftung platziert werden darf, wenn nicht gegen die Topologie verstoßen werden soll (Algorithmus 15 – Zeile 17 bis 18; Abbildung 75 c) bis e) rotgestreifte Fläche). Diese Fläche wird im weiteren als Konfliktfläche berücksichtigt, wenn mindestens eine konfliktfreie Beschriftungsposition bei der Berücksichtigung dieser und der beschrifteten Konfliktpartner und deren ausgewählten Beschriftungspositionen für das betrachtete zu beschriftende Objekt verbleibt (Algorithmus 15 – Zeile 19 bis 21;

Abbildung 75 d) und e) grünschraffierte Fläche). Andernfalls wird die topologische Beschriftung für das betrachtete zu beschriftende Objekte insgesamt verworfen oder alternativ nur für das eine zu berücksichtigende Objekt (Algorithmus 15 – Zeile 25). Wenn der Beschriftungsraum eingeeengt wurde und die ursprüngliche Beschriftungsposition nun einen Konflikt hat (Algorithmus 15 – Zeile 26 bis 27), muss eine neue ermittelt werden. Dazu wird das in Algorithmus 11 – Zeile 23 bis 27 eingesetzte Verfahren verwendet (Algorithmus 15 – Zeile 29 bis 34).

Natürlich kann das Verfahren adaptiert werden, um die topologisch korrekte Beschriftung von Küstenorten zur Küstenlinie zu unterstützen (Abbildung 76). Dabei werden in Algorithmus 15 die Zeilen 17 bis 18 geändert: „berechne Beschriftungsraumfläche auf der Seite von *topologyObject* zu *currentObject*“.

Dieses Verfahren ist dem Auswahlalgorithmus (Algorithmus 10) nachgeschaltet. Wenn anschließend eine Minimierung der Kartenverdeckung, d.h. eine Berücksichtigung der Informationsverluste erfolgen soll, kann das Verfahren mit dem hier vorgestellten kombiniert werden und an die Stelle der Bestimmung einer Beschriftungsposition in Algorithmus 15 – Zeilen 29 bis 34 integriert werden.

```

1  Input: zu beschriftende punkthafte Objekte mit: aktueller Beschriftungsposition,
        Liste der (verbliebenen) konfliktfreien Beschriftungsflächen und Liste
        topology, die die zu berücksichtigenden Trennlinien mit auftretenden
        Konfliktmaßstab enthält.
5  Output: Beschriftungspositionen der punkthafte Objekte, die die topologischen
        Trennlinien berücksichtigen soweit möglich.
        // currentScale ist der Maßstab des gerade zu bearbeitenden statischen Konfliktgraphen.
        // Jedes beschriftete punkthafte Objekt besitzt eine Liste mit den zu berücksichtigenden
10 // topologischen Trennlinien (topology-List) mit zugehörigem Maßstab (cutScale) ab dem
        // sie zu berücksichtigen sind - in der Präprocessingphase erzeugt
für jedes beschriftete punkthafte Objekt (currentObject):
        für jeden Eintrag in topology-Liste von currentObject (topologyObject):
15         wenn topologyObject dargestellt wird
                und zugehöriger cutScale von topologyObject größer als currentScale ist
                dann
                        berechne Beschriftungsraumfläche auf der abgewandten Seite von
                                topologyObject zu currentObject
                        wenn bei Berücksichtigung des ermittelten Bereichs als Konfliktfläche
20                                mindestens ein ausreichend großer konfliktfreier Bewegungsraum
                                        für eine konfliktfreie Beschriftung verbleibt
                                dann
                                        markiere den Bereich als Konfliktfläche
                                sonst
25                                        beende innere Schleife alternativ mache weiter mit nächsten Objekt
wenn mindestens eine Konfliktfläche hinzugekommen ist
        und bisherige Beschriftungsposition von currentObject nun einen Konflikt hat
        dann
                für jede freeArea von currentObject bei Berücksichtigung aller Konflikt-
30                partner und den ermittelten Konfliktflächen (currentFreeArea):
                        ermittle Position mit bester Bewertung in currentFreeArea
                        und speichere ermittelte Position und Bewertung in labelingPos
                        der Eintrag mit der besten Bewertung in der Liste labelingPos ist die neue
                        Beschriftungsposition von currentObject

```

Algorithmus 15 Berücksichtigung der topologisch korrekten Lage von Schriftobjekten in Bezug auf Trennlinien zum zu beschriftenden (punkthafte) Objekt (vgl. Abbildung 75). (Aufruf aus Algorithmus 10 Zeile 46)

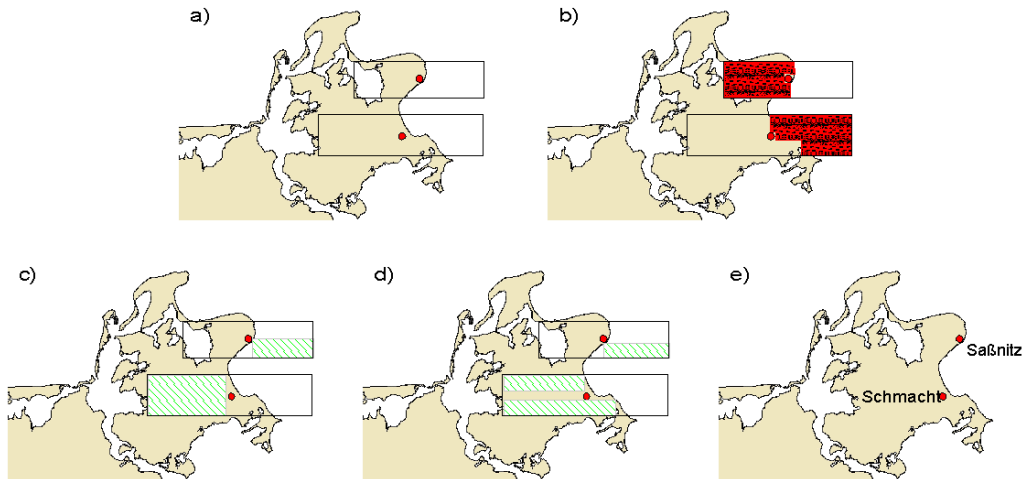


Abbildung 76 Berücksichtigung der topologisch korrekten Beschriftung von Küstenorten – Weiterführung des Beispiels aus Abbildung 66 a). Die Flächen, die für eine topologisch korrekte Beschriftung ausgespart werden müssen sind rot b), grün die vertikalen c) und die horizontalen d) verbleibenden konfliktfreien Bewegungsräume.

Wie schon in Abschnitt 5.2 vorweggenommen ist die Laufzeit abhängig von der Anzahl der beschrifteten Objekte (kleiner der Anzahl der zu beschriftenden Objekte n) – äußere for-Schleife –, der Anzahl der zu berücksichtigenden Trennlinien (t) – innere for-Schleife – sowie der Anzahl der konfliktfreien Beschriftungsräume, die maximal der Anzahl der Gebiete (g) entspricht. Wie in Abschnitt 5.2.4 beschrieben, kann die Anzahl der Gebiete als konstant betrachtet werden. Wenn die Anzahl der zu berücksichtigenden Trennlinien auch als konstant angesehen werden kann bzw. eine Konstante als obere Grenze existiert, was in der Praxis die Regel ist, kann das Verfahren mit $O(n)$ nach oben abgeschätzt werden.

5.3.2 Berücksichtigung von Informationsverlusten

In diesem Abschnitt wird kurz die Thematik des Informationsverlustes, der durch Verdeckungen durch Schriftobjekten entsteht, dargestellt. Dabei wird aufgezeigt, wie diese in den in dieser Arbeit entwickelten Konzepten integriert werden kann. Eine detaillierte Darstellung und Diskussion liegt außerhalb des Fokus dieser Arbeit und kann in [HGM02] und [ELL01] gefunden werden.

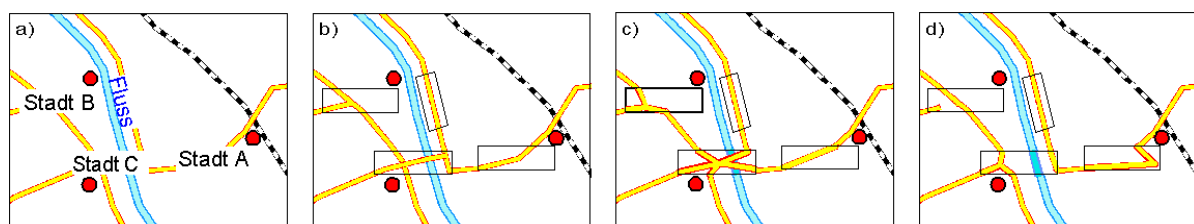


Abbildung 77 Topologischen und geometrische Informationsverluste durch Verdeckung von Kartengrafik.

Neben den zu beschriftenden Objekten und ihren Schriftobjekten gibt es weitere Kartenobjekte, im Folgenden und in der Fachliteratur [HGM02] mit Kartengrafik bezeichnet, die bei der Platzierung der Schriftobjekte „soweit wie möglich“ berücksichtigt werden müssen. Schriftobjekte, die die größte „Bewegungsfreiheit“ in einer Karte haben, können Karteninformationen verdecken, die so dem Kartenleser „entzogen“ werden.

Abbildung 77 a) zeigt einen beschrifteten Kartenausschnitt. Für den Kartenleser bleibt die Kartengrafik verborgen, die unter den Schriftobjekten liegt, was zu einem Informationsverlust

führt. In b) ist die Beschriftung ausgeblendet und die Beschriftungspositionen sind durch Labelboxen dargestellt. Andere Kartengrafikverläufe, die in a) ebenfalls interpretiert werden können, zeigen c) und d). Während der Verlauf der Straße unter der Beschriftung „Fluss“ wahrscheinlich vom Kartenleser richtig interpretiert wird, kommt es bei der Beschriftung von „Stadt A“, „Stadt B“ und „Stadt C“ zu einer Zunahme an Fehlinterpretationen, da wesentliche Eigenschaften der Kartengrafik verdeckt sind. Bei „Stadt B“ stellt sich die Frage ob es eine T-Kreuzung gibt oder eine durchgehende Straße und eine Sackgasse. Hierbei handelt es sich um eine (fehlende) topologische Information. Durch die Verdeckung von „Stadt C“ kann der Kartenleser nicht schließen, ob es eine Verbindung zwischen allen unter der Beschriftung „verschwindenden“ Straßen gibt – der Fluss könnte auch eine „natürliche Barriere“ sein. Wenn es eine Verbindung der Straßen von beiden Seiten des Flusses gibt, sind sie dann durch eine Brücke, einen Tunnel oder eine Fähre verbunden?

Analog zur topologisch korrekten Lage ist die Berücksichtigung von Informationsverlusten nicht zwingend – die „Nicht“-Überlappung von Schriftobjekten ist zwingend – sondern dient der lokalen qualitativen Verbesserung der Beschriftungsposition.

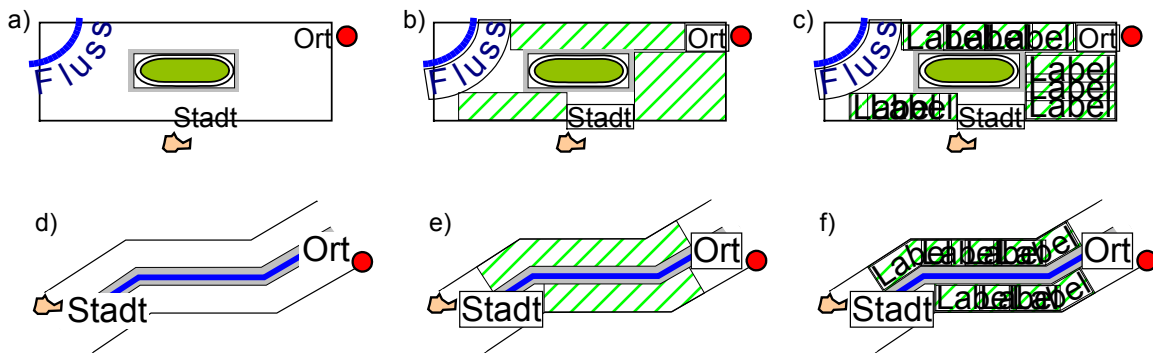


Abbildung 78 Diskretisierung verbleibender konfliktfreier Bewegungsräume punkthafter Objekte (a) bis c) bzw. konfliktfreier Bereiche linienhafter Objekte (d) bis f), die ausreichend groß sind, Beschriftungen konfliktfrei aufzunehmen (grün schraffiert).

Da es im Regelfall keine Beschriftungsposition ohne Verdeckung von Kartengrafik gibt, wird hier der Schritt von den nicht diskreten zu den diskreten Beschriftungspositionen vollzogen. Die Diskretisierung des Beschriftungsraums ist dabei auf die ausreichend großen konfliktfreien Bewegungsräume bzw. -bereiche beschränkt, die nach der Berücksichtigung von Konflikten mit den ausgewählten Beschriftungspositionen der anderen zu beschriftenden Objekten (Abschnitt 5.2 – Algorithmus 10), sowie die Berücksichtigung der Bereiche, die für eine topologisch korrekte Lage der Beschriftung (Abschnitt 5.3.1 – Algorithmus 15) ausgeschlossen sind, verbleiben (Abbildung 78).

In Algorithmus 16 ist ein entsprechendes Verfahren dargestellt. Es ist der letzte Schritt der Feinplatzierung in der Interaktionsphase. Dafür wird, wenn nicht am beschrifteten Objekt gespeichert, die momentane Beschriftungsqualität der beschrifteten Objekte bestimmt (Algorithmus 16 – Zeile 10). Zusammen mit gewichteter Priorität und Beschriftungsart werden die beschrifteten Objekte sortiert (Algorithmus 16 – Zeile 11 bis 12) und in dieser Reihenfolge abgearbeitet (Algorithmus 16 – Zeile 13 bis 23). Bei jedem Objekt (*currentObject*) wird dazu der verbleibende konfliktfreie Bewegungsraum bzw. Bereich, der groß genug ist, die Beschriftung konfliktfrei aufzunehmen, diskretisiert, wie in Abbildung 78 dargestellt (Algorithmus 16 – Zeile 14 bis 15). Konfliktflächen sind die Flächen, in denen Beschriftungen anderer zu beschriftender Objekte stehen oder die für eine topologisch korrekte Lage der Beschriftung in Algorithmus 15 (Abschnitt 5.3.1) ausgeschlossen wurden.

Für die Letztgenannten ist der Ausschluss nicht zwingend. Die Diskretisierung kann z.B. durch eine maximale (prozentuale) Überlappungsfläche zwischen Beschriftungspositionen erfolgen. Für jede der diskreten Beschriftungspositionen (Algorithmus 16 – Zeile 16) werden die Objekte bestimmt, die diese verdeckt (Algorithmus 16 – Zeile 17 bis 19). Dabei können die Datenstrukturen, die in der Präprocessingphase aufgebaut wurden (Abschnitt 4.6 – Algorithmus 8), genutzt werden. Neben der Bewertung des Informationsverlusts wird mit den Bewertungen der in Abschnitt 2.1 entwickelten Funktionen jede Beschriftungsposition bewertet. Dies kann durch eine Modifikation der Verfahren aus Abschnitt 5.2.1 und 5.2.2 unterstützt werden. Das Objekt wird an der Beschriftungsposition mit der besten Bewertung – aus der Kombination der beiden Bewertungen – beschriftet (Algorithmus 16 – Zeile 23).

```

1  Input: beschriftete Objekte mit weiteren freien Beschriftungspositionen, Priorität,
      und den in Konflikt stehenden Kartenobjekten mit Konfliktmaßstab.
      Maßstab zu dem die Beschriftung erfolgen soll.
      Output: Beschriftungspositionen, die keinen oder minimalen Informationsverlust
5     verursachen soweit möglich.
      // currentScale ist der Maßstab des gerade zu bearbeitenden statischen Konfliktgraphen.
      // Jedes zu beschriftende Objekt besitzt eine Liste an Kartenobjekten (mapConflicts-List)
      // deren Verdeckung zu Informationsverlusten führt mit zugehörigem Maßstab (cutScale) ab
      // dem sie zu berücksichtigen sind - in der Präprocessingphase erzeugt
10  bestimme zu jedem beschrifteten Objekt die momentane Beschriftungsqualität
      sortiere die beschrifteten Objekte absteigend nach Kombination aus Priorität,
      Beschriftungsart und -qualität
      für jedes dieser Objekte (currentObject) - durchlaufe sortierte Liste:
      diskretisiere die für die konfliktfreie Beschriftung ausreichend großen
15     konfliktfreien Beschriftungsräume bzw. Bereiche (discretePositionList)
      für jede diskrete Beschriftungsposition (discretePosition):
      bestimme die Objekte (conflictObjects) aus mapConflicts, die dargestellt werden
      und deren zugehöriger Schnittmaßstab cutScale größer currentScale ist
      und discretePosition schneidet
20     ermittle für jede discretePosition Informationsverlust bei Berücksichtigung der
      ermittelten Objekte conflictObjects
      bewerte discretePosition (abgesehen von Informationsverluste)
      wähle die beste der bewerteten diskreten Beschriftungspositionen

```

Algorithmus 16 Minimierung des Informationsverlustes durch lokale Auswahl der Beschriftungspositionen. (Aufruf aus Algorithmus 10 Zeile 46)

Für die Bestimmung des Informationsverlusts (Algorithmus 16 – Zeile 20 bis 21) – die Bewertung ist unabhängig davon ob es sich um eine Beschriftungsposition von punkthaften oder linienhaften Objekten handelt – können Verfahren aus [ECM96], [PET96] oder [ELL01] genutzt werden, die im Folgenden kurz beschrieben werden.

Bei [ECM96] und [PET96] erfolgt die Bewertung des Informationsverlusts bei Verdeckungen linienhafter Objekte über die Länge des verdeckten Bereichs oder über die Distanz zwischen den Schnittpunkten von Linienzug und Beschriftungsposition („Ein-“ und „Aus“-Tritt) und der Priorität des verdeckten Objekts. Flächenhafte Geometrien werden wie geschlossene linienhafte Objekte behandelt. Die Bewertung des Informationsverlustes verdeckter punkthafter Objekte ist einfacher und erfolgt über deren Priorität.

Die Bestimmung des Informationsverlusts mit dem Verfahren aus [ELL01] basiert neben der zuvor vorgestellten Bewertung zusätzlich auf der Verlustbewertung von topologischen Informationen. Als topologische Informationen werden in diesem Kontext z.B. Kreuzungen angesehen, die – wenn sie verdeckt werden – zu Fehlinterpretationen des Kartenlesers führen, da wegen der Verdeckung nicht erkennbar ist, ob Objekte zusammenhängen oder nicht.

Die Bewertung des Informationsverlusts wird hier nicht weiter behandelt und als bekannt vorausgesetzt. Details können [ECM96], [PET96] und [ELL01] entnommen werden.

Die Integration der Bewertung erfolgt in der Bewertung der Beschriftungsposition in Algorithmus 16 – Zeile 20 bis 23. Dabei wird die Beschriftungsposition aus einer kombinierten und gewichteten Bewertung des Informationsverlustes (Algorithmus 16 – Zeile 20 bis 21) und der Bewertung der Beschriftungsposition aus Abschnitt 2.1 zusammengesetzt. Die Bewertung kann mit der Idee der Verfahren aus Abschnitt 5.2.1 und 5.2.2, möglichst in der Mitte von konfliktfreien Räumen zu beschriften, ergänzt werden.

Zur Abschätzung der Laufzeit des Verfahrens wird Algorithmus 16 in zwei Teile zerlegt. Der erste Teil des Verfahrens (Algorithmus 16 – Zeile 1 bis 12) kann mit $O(n \log n)$ nach oben abgeschätzt werden. Dabei wird angenommen, dass die Anzahl für die im Auswahlalgorithmus beschrifteten Objekte gleich der Anzahl der zu beschriftenden Objekte (n) ist. Das ermöglicht auch eine einheitliche Betrachtung der Laufzeit in Abschnitt 5.2.4.

Die Schleife Algorithmus 16 – Zeile 13 bis 23 kann mit $O(n * d * c)$ als obere Schranke abgeschätzt werden. Dabei ist d die Anzahl der diskreten Positionen (*discretePositionList*) und c die Anzahl der Konfliktobjekte (*mapConflicts*). Während die Anzahl der Beschriftungspositionen pro zu beschriftendem Objekt mit einer Konstanten nach oben abgeschätzt werden kann – ein geeigneter Algorithmus kann diese Anzahl garantieren –, kann die Anzahl der schon während der Präprocessingphase ermittelten Konfliktobjekte nur durch die Anzahl aller Kartenobjekte (abgesehen von den Schriftobjekten) abgeschätzt werden. Diese ist mindestens so groß wie die Anzahl der zu beschriftenden Objekte. Dadurch ist die Laufzeit höher als die quadratische Anzahl der zu beschriftenden Objekte.

Wie auch die Berücksichtigung der topologisch korrekten Lage ist dieses Verfahren zur Feinplatzierung nur optional, wenn ausreichend Zeit zur Verfügung steht.

5.4 Zusammenführung von unterschiedlichen Beschriftungen des gleichen Kartenausschnitts

Im Unterschied zu klassischen kartographischen Papierkarten können Nutzer Bildschirmkarten interaktiv ihren Erfordernissen durch Ein- und Ausblenden von Layern anpassen (Abbildung 79). Jedes Ein- und Ausblenden eines Beschriftungslayers zieht eine (vollständige) Neubeschriftung nach sich. Bei einem naiven Ansatz müsste mit dem beschriebenen Verfahren jeweils ein neuer Konfliktgraph generiert werden, da nur die Beschriftungskonflikte innerhalb eines Layers – jeder Layer besitzt einen separaten statischen Konfliktgraphen –, nicht jedoch die Konfliktgraphen zwischen den unterschiedlichen Layern berücksichtigt werden. Diese Neugenerierung ist während der Interaktionsphase in Echtzeit nicht möglich. Eine Vorausberechnung in der Präprocessingphase ist wegen der Vielzahl der möglichen Kombinationen nicht sinnvoll.

Daher wird ein Verfahren vorgestellt, das die separaten Beschriftungen eines Kartenausschnitts bezüglich eines Maßstabs zu einer gemeinsamen zusammenführt. Die Zusammenführung „maßstabsunabhängiger“ Beschriftungen, d.h. reaktiver Konfliktgraphen, ist nicht sinnvoll, da die Datenmenge im Gegensatz zum Kartenausschnitt bzw. statischen Konfliktgraphen viel größer ist und die Ausdünnung der zu beschriftenden Objekte nicht berücksichtigt wird. In diesem Fall ist eine Neugenerierung mit den Verfahren, das in Kapitel 4 vorgestellt wurde, sinnvoll (Abschnitt 4.3.11 – Algorithmus 7).

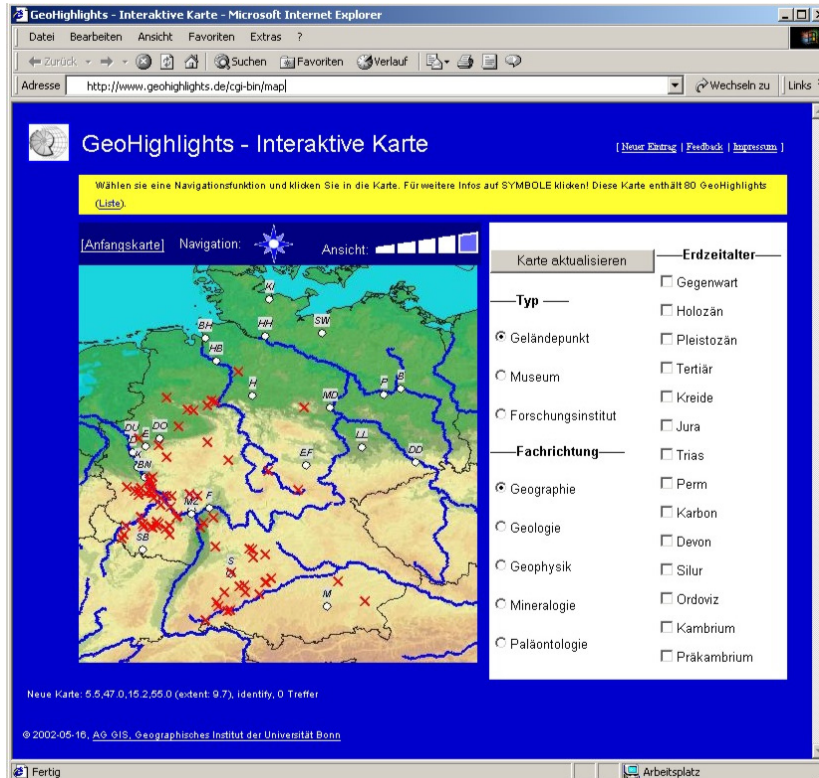


Abbildung 79 Unterschiedliche Beschriftungen desgleichen Kartenausschnitts, die als Layer kombiniert werden können. In dem Beispiel können vierzehn Beschriftungslayer der Erdzeitalter beliebig kombiniert werden. Es gibt $2^{14} = 16384$ Kombinationen. Der naive Ansatz erfordert ebenso viele Konfliktgraphen.

Formal, d.h. mathematischer betrachtet auf Graphenebene, kann die Vereinigung zweier Beschriftungen eines Kartenausschnitts zu einem spezifischen Maßstab, d.h. zweier statischer Konfliktgraphen, wie folgt beschrieben werden: Wenn A und B die existierenden statischen Konfliktgraphen mit zugehörigen Kanten sind, dann ist die Menge der Knoten und Kanten des neuen vereinigten Konfliktgraphen C wie folgt definiert:

$$\text{Knoten}(C) = \text{Knoten}(A) \cup \text{Knoten}(B) \quad (5.1)$$

$$\text{Kanten}(C) = \text{Kanten}(A) \cup \text{Kanten}(B) \cup \text{Kanten}(A, B) \quad (5.2)$$

Wie zu erkennen ist, erfolgt keine Ausdünnung der zu beschriftenden Objekte (Knoten der beiden statischen Konfliktgraphen – Gleichung (5.1)). Neben den bereits ermittelten Konflikten in den statischen Konfliktgraphen A und B ($\text{Kanten}(A)$ und $\text{Kanten}(B)$) kommen die potentiellen Konflikte zwischen den beiden statischen Konfliktgraphen ($\text{Kanten}(A, B)$) hinzu (Gleichung (5.2)).

Die Anzahl der zu beschriftenden Objekte und damit die potentiellen Konflikte nehmen bei gleich bleibender Kartengröße zu. Das führt zu einer Erhöhung der Laufzeit des Auswahlalgorithmus. Die Anzahl der zu beschriftenden Objekte, für die keine konfliktfreie Beschriftungsposition gefunden werden kann, nimmt folglich auch zu. Das sind Gründe, warum das Werkzeug für den „maßstabslosen“ reaktiven Konfliktgraph un geeignet ist und für einen Kartenausschnitt zu einem spezifischen Maßstab wegen der Erhöhung der Flexibilität und damit des Einsatzbereichs toleriert werden kann. Durch iterative Anwendung der Vereinigung zweier statischer Konfliktgraphen kann das Verfahren auf die Vereinigung beliebig vieler verallgemeinert werden.

Wie dargestellt, setzt sich der vereinigte Konfliktgraph aus den Knoten der beiden statischen Konfliktgraphen – den zu beschriftenden Objekten – (Gleichung (5.1)) und den Kanten der beiden statischen Konfliktgraphen – den potentiellen Konflikten innerhalb der statischen Konfliktgraphen – (Gleichung (5.2)) zusammen. Zusätzlich kommen die Konflikte zwischen den zu beschriftenden Objekten der unterschiedlichen statischen Konfliktgraphen ($Kanten(A,B)$ – Gleichung (5.2)) hinzu. Dazu können bekannte Verfahren für „Spatial Joins“ verwendet werden. Eines dieser Verfahren ist in Algorithmus 17 dargestellt. Es ist aus [RSV02] entnommen. Das Verfahren ist eine Kombination aus Plane-Sweep und Nested-Loop Techniken.

```

1  Input: Knoten der beiden zusammenzuführenden statischen Konfliktgraphen in
      zwei Arrays nodesA und nodesB. Die Knoten sind mit ihrer zum Maßstab
      gehörigen Boundingbox attribuiert.
      Output: Menge der Kanten, die durch die Zusammenführung der beiden statischen
5  Konfliktgraphen hinzukommen (additionalConflicts).
      // Vereinigung der statischen Konfliktgraphen A und B. Die zu beschriftenden Objekte
      // sind die Knoten der Graphen.
      // nodesA und nodesB sind Arrays mit den zu beschriftenden Objekten bzw. Knoten der
      // statischen Konfliktgraphen A und B
10  // indexA und indexB sind Zeiger auf die aktuellen Felder der Arrays
      // additionalConflicts ist die Menge der Tupel (a,b) a ist Knoten im statischen Konflikt-
      // graphen A und b ist Knoten im statischen Konfliktgraphen B und es existiert eine Über-
      // lappung zwischen den beiden zu beschriftenden Objekten a und b
      additionalConflicts =  $\emptyset$ 
15  sortiere die Knoten von A nach den x-Koordinaten ihrer Boundingboxen (nodesA)
      indexA = 1
      sortiere die Knoten von B nach den x-Koordinaten ihrer Boundingboxen (nodesB)
      indexB = 1
      solange (indexA <= |nodesA|) und (indexB <= |nodesB|)
20  wenn nodesA[indexA].boundingbox.x < nodesB[indexB].boundingbox.x
      dann
          a = nodesA[indexA]
          s = posB
          b = nodesB[s]
25  solange b.boundingbox.x <= (a.boundingbox.x + a.boundingbox.width)
      wenn a.boundingbox  $\cap$  b.boundingbox  $\neq \emptyset$ 
          dann
              additionalConflicts = additionalConflicts  $\cup$  {(a, b)}
              s++
30  b = nodesB[s]
      indexA++
      sonst
          tausche nodesA und nodesB sowie indexA und indexB

```

Algorithmus 17 Bestimmung der Überlappungen zwischen zwei statischen Konfliktgraphen A und B ($Kanten(A,B)$ – Gleichung (5.2)). Dazu wird das Verfahren zur Ermittlung von Überlappungen zwischen zwei Rechteckmengen aus [RSV02] verwendet.

Die zu beschriftenden Objekte des statischen Konfliktgraphen A werden in dem Array *nodesA* und die von B in *nodesB* abgelegt, sortiert und Zeiger auf ihnen initialisiert (Algorithmus 17 – Zeile 15 bis 18). Die solange-Schleife (Algorithmus 17 – Zeile 19 bis 33) wird sooft durchlaufen, bis einer der Zeiger hinter dem letzten Element steht. Im Weiteren erfolgt eine Betrachtung der x-Koordinaten und das ausgewählte Element von *nodesA* schaut voraus auf Konflikte mit Konflikten aus *nodesB*. Durch den wenn-Teil (Algorithmus 17 – Zeile 20) und die solange-Schleife (Algorithmus 17 – Zeile 25 bis 31) wird das gerade betrachtete Objekt *a* aus *nodesA* mit allen Objekte *b* aus *nodesB* verglichen, deren Boundingboxanfang vor dem Boundingboxende von *a* liegt (Algorithmus 17 – Zeile 25). Ggf. werden die Paare als Tupel in die Ergebnisliste übernommen (Algorithmus 17 – Zeile 26 bis 28). Einen Test, dass sich die Geometrien tatsächlich überlappen – nicht nur die Boundingboxen –, muss abschließend

erfolgen. Wenn die Objekte durchlaufen sind, werden die Listen und Zeiger getauscht (Algorithmus 17 – Zeile 33), so dass die *nodesB* abgearbeitet werden können – nur *indexA* wird erhöht (Algorithmus 17 – Zeile 31).

Die obere Schranke kann mit $O(n^2)$ abgeschätzt werden [RSV02]. Der Worst-Case tritt dann ein, wenn alle Boundingboxen im gleichen x -Intervall liegen.

Kapitel 6

Komponenten des Prototypen

Im Rahmen dieser Arbeit wurde ein komponentenbasiertes System geschaffen, in dem wesentliche Konzepte, die in dieser Arbeit entwickelt wurden, implementiert und erfolgreich evaluiert wurden. Da die automatische Schriftplatzierung eng mit der kartographisch angemessenen Visualisierung von Geoobjekten verknüpft ist, wurde eine zweistufige Strategie bei der Entwicklung des Systems gewählt.

Ziel war es zunächst, eine Plattform für die Umsetzung und Evaluation von Beschriftungskonzepten zu schaffen. Dieser Kern des komponentenbasierten Systems konnte erfolgreich umgesetzt werden. Er besteht aus drei Hauptkomponenten, der Schnittstelle zum Datenimport, der Struktur zur Speicherung der Geo- bzw. Kartenobjekte sowie deren Visualisierung.

In einer zweiten Stufe wurden entwickelte Beschriftungskonzepte als Komponenten umgesetzt, in das System integriert und evaluiert. Die Umsetzung aller entwickelten und beschriebenen Strukturen und Verfahren konnte auf Grund des Umfangs nicht realisiert werden und sind auf punkthafte zu beschriftende Objekte beschränkt.

6.1 Überblick

Das komponentenbasierte System ist ein offenes, objektorientiertes und interoperables System. Die Wahl einer objektorientierten Programmiersprache bot sich daher an. Da Bildschirmkarten in Echtzeit insbesondere durch das Internet an Bedeutung gewinnen und genutzt werden, sollte die gewählte Programmiersprache unabhängig vom Betriebssystem sein und die Einbindung in Internetapplikationen ermöglichen. Java unterstützt diese Anforderungen und wurde daher als Entwicklungsplattform gewählt.

Abbildung 80 zeigt die Struktur des komponentenbasierten Systems als statisches UML-Klassenmodell. Auf Grund des geringen Platzes ist die Darstellung auf die wichtigsten Klassen und Assoziationen beschränkt. Die dargestellten Klassen und Assoziationen sind implementiert, jedoch nicht alle Operationen. In den folgenden Abschnitten werden die realisierten Komponenten vorgestellt.

Die Abbildung 81 und Abbildung 83 bis Abbildung 85 zeigen Screenshots des entwickelten Systems. Die Oberfläche setzt sich aus mehreren Fenstern zusammen, in diesem Beispiel aus dem Kartenfenster („Map“), dem Fenster für Objektinformationen („Object Info“) und dem Übersichtsfenster („Overview“).

Der Kartenausschnitt kann im Kartenfenster durch die Scroll-Balken frei gewählt werden. Die Koordinate, an der der Mauszeiger steht, wird genauso angezeigt wie die benötigte Rechenzeit für die letzte Änderung des Kartenausschnitts und -maßstabs – in der unteren Statuszeile

des Kartenfensters. Durch die Zoom-In- und Zoom-Out-Schaltflächen sowie dem Slider (Schiebereglar) im Kartenfenster und durch direkte Eingabe des Maßstabs kann der Nutzer den Maßstab bestimmen. Die Angabe des Maßstabs erfolgt sowohl in Blattkoordinaten („data Scale“) als auch in Bildschirmkoordinaten („screen Scale“). Selektierte Objekte – durch An-klicken der Objekte mit der rechten Maus-Taste – werden hellblau markiert und zugehörige Informationen im „Object Info“-Fenster angezeigt. Das Übersichtsfenster („Overview“) zeigt alle Daten und den augenblicklichen Kartenausschnitt als Rechteck.

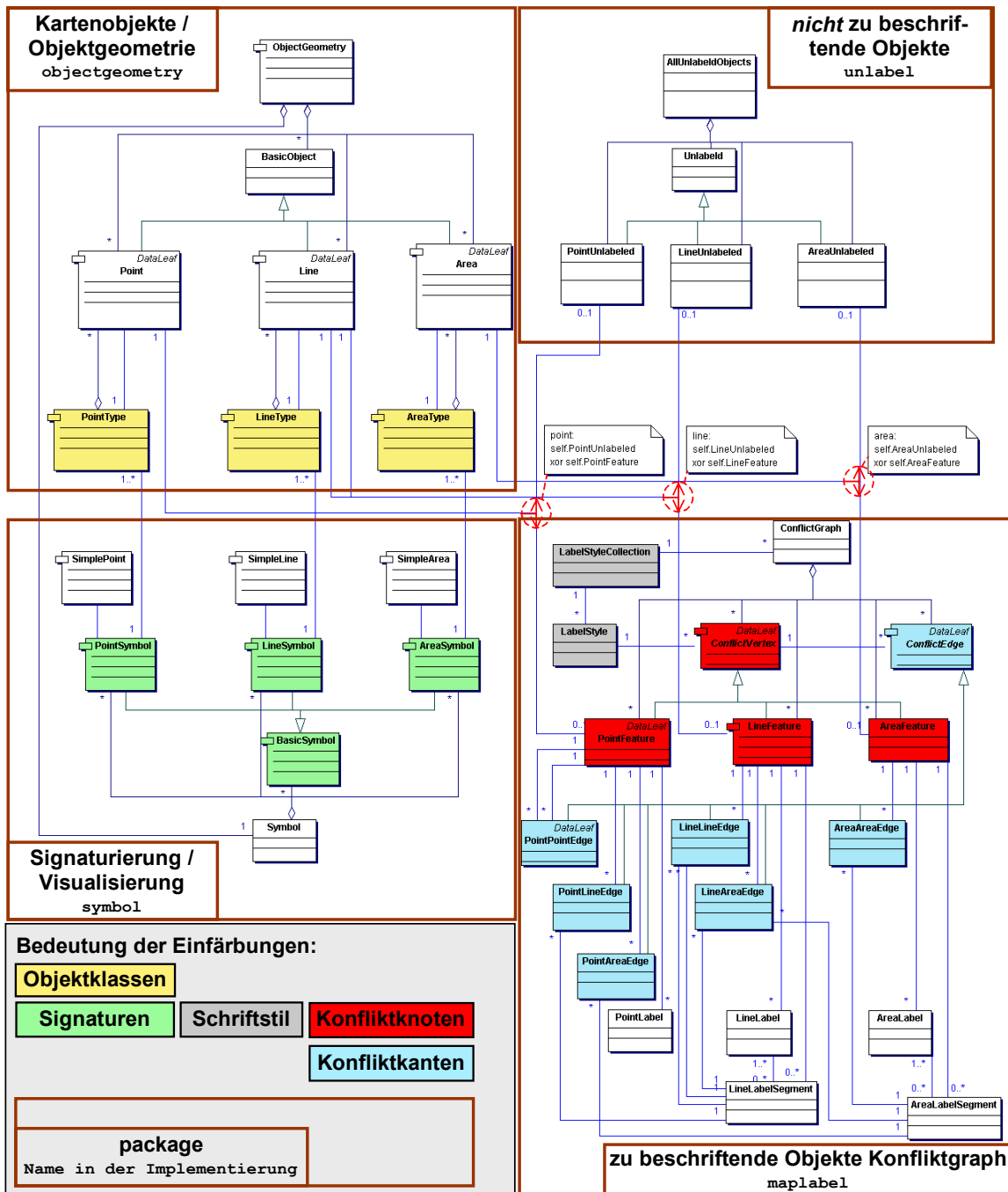


Abbildung 80 Statisches UML-Klassenmodell des komponentenbasierten Systems.

Über das Hauptmenu und die Buttons können folgende Aktionen ausgelöst werden, die teilweise in eigenen Abschnitten detaillierter beschrieben werden:

- Datenimport
- Import von georeferenzierten Bildern und einblenden als Hintergrund
- Generierung des reaktiven Konfliktgraphen mit einem der beiden frei parametrisierbaren Kriterien zur Bestimmung der Beschriftungsschwierigkeit:
 - Knotengrad – Anzahl der Konflikte
 - freie Beschriftungsfläche
 - (beides zurzeit auf punkthafte Objekte beschränkt)
- Ausgabe der Knoten und Kanten des reaktiven Konfliktgraphen mit allen Informationen in Dateien
- Visualisierung des reaktiven und statischen Konfliktgraphen (und Selektierbarkeit dieser)
- Ein- und Ausblenden der im statischen Konfliktgraph nicht enthaltenen Objekte
- Visualisierung der Beschriftungsräume des statischen Konfliktgraphen (zurzeit auf punkthafte Objekte beschränkt)
- Ein- und Ausblenden von Schrift für punkthafte Objekte:
 - einfache Platzierung oben rechts in der Referenzschriftgröße
 - Platzierung oben rechts in der Schriftgröße passend zum Maßstab
 - konfliktfreie Beschriftung
 - optional kann die Schrift freigestellt werden
- Visualisierung der größten konfliktfreien Bewegungsräume – wenn das zu beschriftende punkthafte Objekt keinen Konflikt im aktuellen Maßstab hat, wird die Fläche der Beschriftungsposition oben rechts hervorgehoben
- Hervorhebung der zu beschriftenden Objekte des statischen Konfliktgraphen, für die keine konfliktfreie Beschriftungsposition gefunden wurde und die daher nicht beschriftet werden
- Berücksichtigung der Signaturen von zu beschriftenden Objekten bei Ermittlung von Beschriftungspositionen bzw. konfliktfreien Bewegungsräumen
- Protokollierung aller Aktionen in einer log-Datei

Die Laufzeit des Prototypen und seiner Funktionen ist abhängig von der Anzahl der Geoobjekte, der davon zu beschriftenden Objekte, der verwendeten Hardware und der virtuellen Java Maschine. Theoretische Betrachtungen wurden dazu in den beiden konzeptionellen Kapiteln an den entsprechenden Stellen detailliert angestellt. Eine allgemeine Aussage über die Laufzeit ist daher schwierig zu treffen und wird deshalb exemplarisch an einem Beispiel gezeigt.

Als Testdaten dienen dabei die Daten der Städte Nordrhein-Westfalens, die der „ArcDeutschland'500“ der Firma ESRI [ESR93] entnommen sind. Als zu beschriftende Objekte wurden die 1235 Städte verwendet mit dem Attribut Stadtbevölkerung zur Bestimmung der Objektpriorität und dem Stadtnamen als zu platzierendes Schriftobjekt. Die Generierung des reaktiven Konfliktgraphen für das Beispiel benötigt für das Kriterium Konfliktgrad 26 Sekunden und für das Kriterium freie Beschriftungsfläche 43 Sekunden. Tabelle 4 gibt einen Überblick über die Laufzeiten für die Funktionen Scrollen und Zoomen. Zu den Laufzeiten muss angemerkt werden, dass es sich um einen Prototypen handelt, bei dem die Effizienz und damit die Laufzeit vieler Methoden erheblich verbessert werden können und die für die Auswertung und zum Nachvollziehen wichtigen Daten ausführlich in

einer log-Datei protokolliert werden, die in diesem Beispiel nach wenigen Operationen auf einige 100MB ansteigt und es somit zu einer weiteren Laufzeiteinbuße kommt.

Für die Größe der Datenmenge konnten somit für einen Prototypen diesen Umfangs zufriedenstellende Laufzeiten erzielt werden.

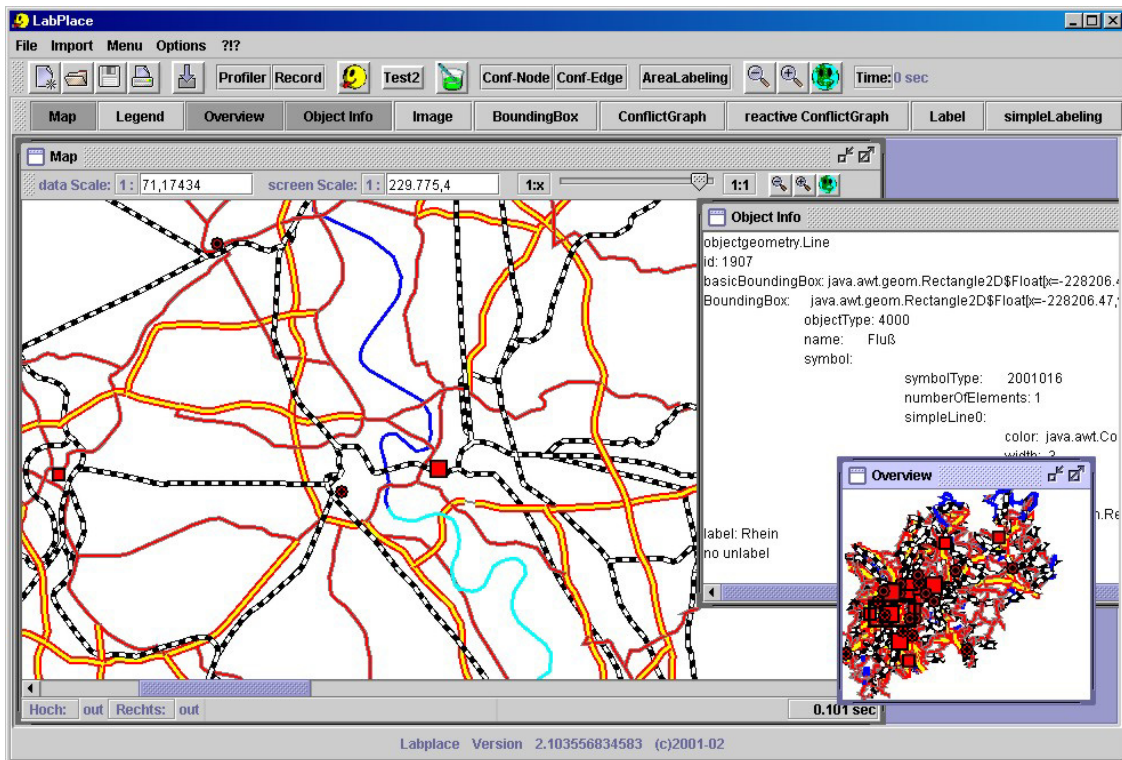


Abbildung 81 Screenshot des komponentenbasierten Prototypen mit Kartenfenster („Map“), Fenster für Objektinformationen („Object Info“) und Übersichtsfenster („Overview“).

	punkthafte Objekte (alle)	punkthafte Objekte (alle) mit Bounding-box	punkthafte Objekte des statischen Konfliktgraphen	punkthafte Objekte des statischen Konfliktgraphen mit Boundingbox (Abbildung 83)	konfliktfrei beschriftete punkthafte Objekte	konfliktfrei beschriftete punkthafte Objekte und Berücksichtigung von Signaturen (Abbildung 84)	Darstellung von Abbildung 81
Scrollen	0,13	0,16	0,04	0,06	0,7	0,9	0,7
Zoomen	0,12	0,16	0,03	0,08	0,8	1,0	0,7

Tabelle 4 Laufzeiten des Prototypen in Sekunden für die 1235 Städte Nordrhein-Westfalens, ausgeführt auf einem Rechner mit Pentium III 1 GHz. Die Laufzeit ist abhängig vom statischen Konfliktgraphen, d.h. vom Maßstab, Ausschnitt und der Anzahl der zu beschriftenden Objekte. Die Daten sind aus [ESR93] entnommen.

6.2 Offene Schnittstellenkomponente

Da das entwickelte System nicht auf ein bisher existierendes System aufsetzt und offen sein soll, ist eine offene interoperable Schnittstelle unerlässlich. Dazu wurde eine den Anforderungen entsprechende Schnittstelle zum Import von Daten im Interchange File Format von

ESRI¹⁸ implementiert. Dadurch ist ein unkomplizierter Datenimport von Objekten mit beliebigen zweidimensionalen Geometrien und zugehörigen Attributen aus marktführenden GIS möglich.

6.3 Klassifizierung und Signaturzuordnung

Topographische Daten, wie sie mit der Schnittstellenkomponente importiert werden, sind keine kartographischen Daten und beinhalten Geometrien, aber keine (direkten) Informationen über ihre Grafik, d.h. Informationen wie sie dargestellt werden sollen. Diese Informationen sind nicht nur für die Visualisierung wichtig, sondern insbesondere auch für die automatische Schriftplatzierung, was in den beiden vorhergehenden Kapiteln dargestellt wurde.

Das entwickelte und implementierte Verfahren zur automatischen Signaturierung basiert auf einem dreistufigen Konzept: Klassifikation, Definition von Signaturen und Zuordnung von Klassen zu Signaturen.

Zunächst werden die Geoobjekte anhand ihrer Attribute automatisch klassifiziert. Die zur Einteilung in Klassen nötigen Informationen werden in zu den Geodaten gehörigen „match“-Dateien bereitgestellt.

Signaturen werden getrennt für die Signaturierung punkt-, linien- und flächenhafter Objekte in Dateien beschrieben („point.symbol.type“, „line.symbol.type“ und „area.symbol.type“). Die Beschreibung erfolgt mittels primitiver, übereinander gelagerter Geometrien attribuiert mit Größen, Farben, Linientypen etc.

Schließlich erfolgt in einem letzten Schritt die Zuordnung von Signaturen zu Klassen. Diese Informationen stehen in einer weiteren Datei „objectType-symbolType.match“.

Diese, auf den ersten Blick umständlich aussehende Vorgehensweise, hat den Vorteil einer hohen Flexibilität und einer kartographisch angemessenen Behandlung.

6.4 Visualisierungskomponente

Die Visualisierung der Geoobjekte erfolgt mit den ihnen zugeordneten Signaturen, wie im letzten Abschnitt beschrieben. Ein Beispiel ist in Abbildung 81 dargestellt. Die Darstellung von Schriftobjekten wird separat in Abschnitt 6.7 besprochen.

Zusätzlich wurde für die Verifikation und Veranschaulichung der entwickelten Datenstruktur des Konfliktgraphen die Visualisierung des reaktiven und des statischen Konfliktgraphen realisiert. Das Einblenden der zu den Objekten gehörigen Boundingboxen in Abhängigkeit des dargestellten Maßstabs ist eine weitere Funktionalität.

Informationen zu Kartenobjekten und Elementen der Konfliktgraphen können über deren Selektion mit der rechten Maustaste eingeblendet werden. Dies ist in Abbildung 81 dargestellt.

Zusätzlich können Rastergrafiken im jpg- und gif-Format als Kartenhintergrund angezeigt werden. Für ihre korrekte Einpassung wird zu jeder Grafik bzw. Datei ihr Ursprung sowie die nötige Verzerrung in Breite und Höhe angegeben. Die Einblendung von Rastergrafiken als Hintergrund ist eine sehr nützliche Funktion für Prototypen, da damit entwickelte Verfahren bzw. deren Ergebnisse gut verifiziert und verglichen werden können.

¹⁸ Environmental Systems Research Institute, Sitz in Redlands, USA. (<http://www.esri.com>)

6.5 Effiziente geometrische Zugriffsstrukturen – R-Baum

Auf dem Bildschirm wird in der Regel eine Teilmenge aller Kartenobjekte dargestellt. Damit die im Bildschirmausschnitt liegenden Objekte effizient während der Interaktionsphase ermittelt werden können – bei der Ableitung des statischen aus dem reaktiven Konfliktgraphen – wird eine effiziente geometrische Datenstruktur zur Speicherung und Verwaltung von dreidimensionalen Objekten benötigt (vgl. Abschnitt 5.1).

In diesem Kontext wurde der 3D-R-Baum als Komponente in Java implementiert. Durch die Verwendung einer offenen Schnittstelle ist seine Nutzung nicht auf den Prototypen beschränkt.

6.6 Vorausberechnung von Konflikten – Konfliktgraphen

Die in den Abschnitten 4.3.2.1, 4.3.5.1 und 4.3.7.1 vorgestellten Verfahren zur Bestimmung der Schnittmaßstäbe zwischen punkthaften, linienhaften und punkt- und linienhaften zu beschriftenden Objekten wurden als Komponenten prototypisch implementiert. Abbildung 82 zeigt Screenshots der Komponenten zur Bestimmung des Schnittmaßstabs zwischen linienhaften und zwischen punkt- und linienhaften zu beschriftenden Objekten. Der linke Screenshot in Abbildung 82 stammt von der Komponente zur Bestimmung des Schnittmaßstabs zwischen zwei linienhaften zu beschriftenden Objekten (vgl. Abschnitt 4.3.5.1). Ein Screenshot der Komponente zur Ermittlung des Schnittmaßstabs zwischen einem punkt- und einem linienhaften zu beschriftenden Objekt ist rechts in Abbildung 82 dargestellt (vgl. Abschnitt 4.3.7.1). Die Beschriftungsräume sind blau, die zu beschriftenden Objekte sind rot dargestellt. Zusätzlich sind bei dem punkthaften Objekt die Diagonalen (Sektorengrenzen) des Beschriftungsraums grau dargestellt. Grün ist der Schnittpunkt der Verlängerung des linienhaften Objekts mit der Diagonalen markiert.

Die Kriterien zur Bestimmung der Beschriftungsschwierigkeit, Knotengrad (Abschnitt 4.3.3.1) und freie Beschriftungsfläche (Abschnitt 4.3.3.3) wurden exemplarisch für punkthafte Objekte zur Generierung des reaktiven Konfliktgraphen umgesetzt. Wie in Abschnitt 4.3.3.4 vermutet, ergibt sich ein unterschiedlicher reaktiver Konfliktgraph für die beiden Kriterien.

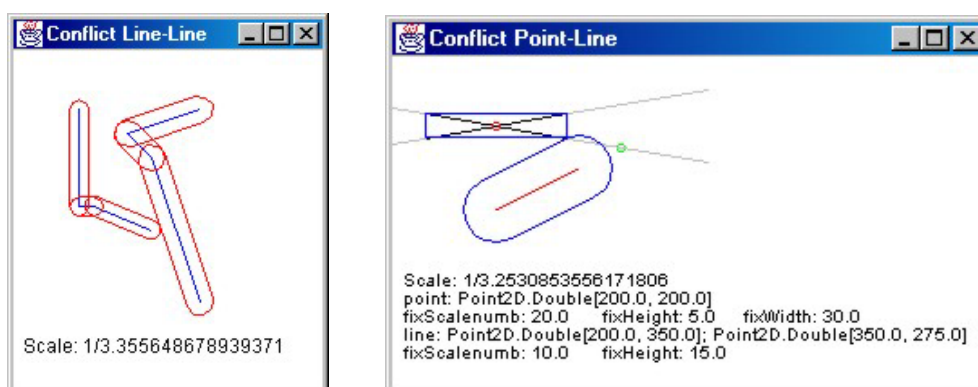


Abbildung 82 Screenshots von Komponenten zur Bestimmung der Schnittmaßstäbe. Der Darstellungsmaßstab ist gleich dem Schnittmaßstab, wodurch es zu der Berührung der beiden Konfliktpartner kommt.

Zum besseren Vergleich sind in Abbildung 83 die statischen Konfliktgraphen für die beiden Kriterien zum gleichen Maßstab, gleichen Ausschnitt, gleicher Datenbasis, etc. dargestellt, d.h. der einzige Unterschied ist die Wahl des Kriteriums zur Generierung des reaktiven Konfliktgraphen. Für beide Kriterien wurden die Parameter so gewählt, dass die Anzahl der Knoten und Kanten der dargestellten statischen Konfliktgraphen nahezu gleich sind. Für den

Knotengrad wurden maximal fünf Konflikte zu Objekten mit höherer Priorität zugelassen und für die freie Beschriftungsfläche wurde die Bewertung $labelScore_{max} = 0,7$ verwendet, d.h. es muss mindestens ein konfliktfreier Bewegungsraum von 70% der zu platzierenden Schriftfläche zu Objekten mit höherer Priorität verbleiben.

Das führt dazu, dass in dem statischen Konfliktgraphen in Abbildung 83 links für das Kriterium Knotengrad die Anzahl der Konfliktgraphknoten 51 und für die Konfliktgraphkanten 83 ist. Für Abbildung 83 rechts mit dem Kriterium freie Beschriftungsfläche ergeben sich 55 Konfliktgraphknoten und 79 Konfliktgraphkanten. Die Anzahl der Knoten und Kanten ist also nahezu gleich. Die statischen Konfliktgraphen unterscheiden sich somit nur in den ausgewählten Knoten, d.h. den Objekten, die versucht werden sollen, zu dem dargestellten Maßstab und Ausschnitt zu beschriften.

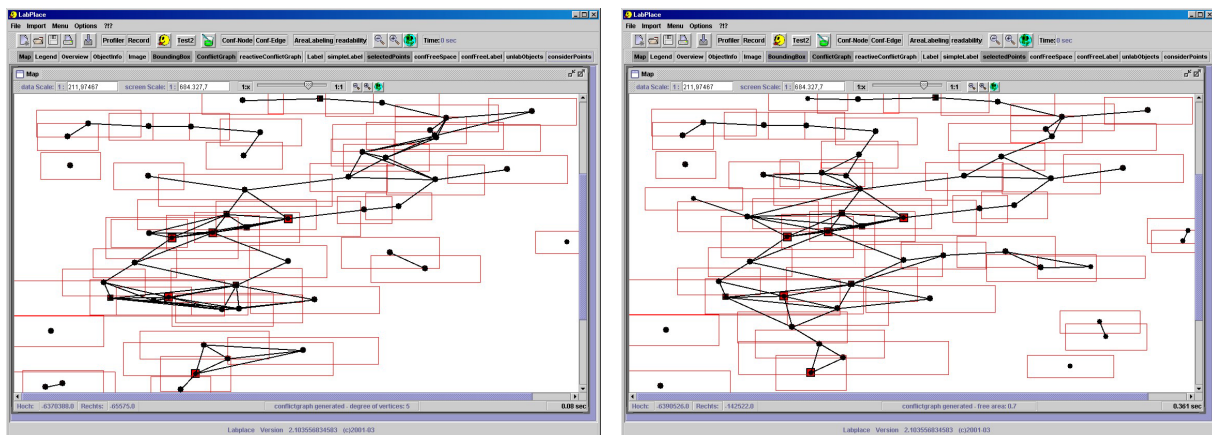


Abbildung 83 Auswirkung auf den statischen Konfliktgraphen durch die Wahl des Kriteriums zur Bestimmung der Beschriftungsschwierigkeit: links das Kriterium „Knotengrad“ und rechts das Kriterium „freie Beschriftungsfläche“, wobei bei dem Kriterium „Knotengrad“ weniger zu beschriftenden Objekte selektiert sind (51 zu 55) und gleichzeitig mehr Konflikte auftreten (83 zu 79) als mit dem Kriterium „freie Beschriftungsfläche“.

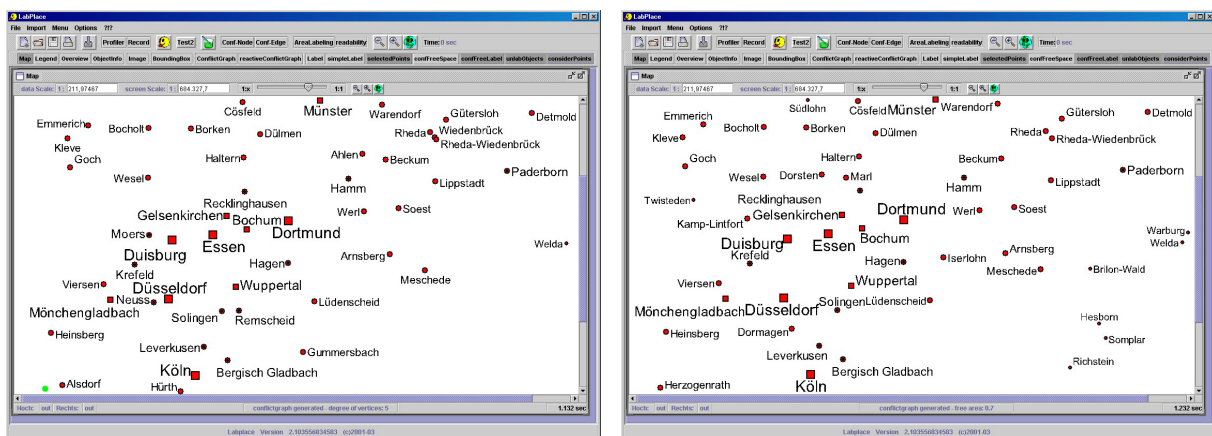


Abbildung 84 Beschriftung der Kartenausschnitte auf Basis der in Abbildung 83 dargestellten statischen Konfliktgraphen: links das Kriterium „Knotengrad“ und rechts das Kriterium „freie Beschriftungsfläche“.

Die beschrifteten Bildschirmkarten auf Basis der in Abbildung 83 gezeigten statischen Konfliktgraphen ist in Abbildung 84 dargestellt. Die in Abschnitt 4.3.3.4 beschriebenen Unterschiede, d.h. die gleichmäßigere Verteilung der zu beschriftenden Objekte bei dem Kriterium „freie Beschriftungsfläche“ (Abbildung 84 rechts) als bei dem Kriterium „Konfliktgrad“ (Abbildung 84 links), konnten somit erfolgreich nachgewiesen werden.

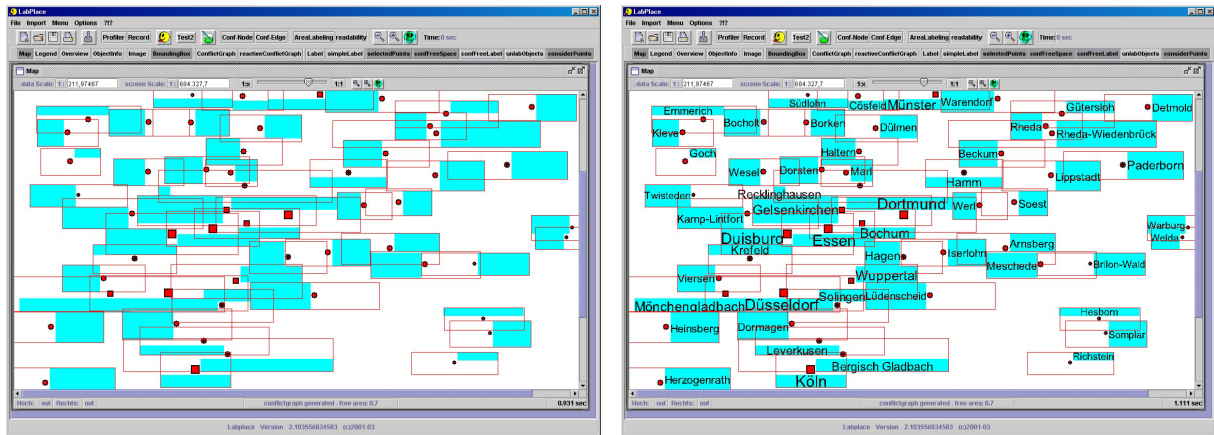


Abbildung 85 Ermittelte größte verbleibende konfliktfreie Bewegungsräume für jedes zu beschriftende Objekt links und mit Beschriftung rechts. Ist der Beschriftungsraum konfliktfrei, erfolgt die Platzierung oben rechts.

6.7 Vermeidung von Konflikten – Beschriftung

Der Auswahlalgorithmus aus Abschnitt 5.2 wurde in einer modifizierten Version implementiert. Er ist auf die Beschriftung punkthafter Objekte beschränkt, die Bewertung der Beschriftungsposition und die Feinplatzierung wurden abgewandelt realisiert. Die konfliktfreien Bereiche werden, wie in Algorithmus 10 beschrieben, bestimmt, d.h. für das gerade betrachtete zu beschriftende Objekt werden zunächst die Beschriftungspositionen der schon beschrifteten Objekte berücksichtigt. Dann erst werden soweit wie möglich die Beschriftungsräume der noch nicht behandelten Objekte und der Objekte, für die keine konfliktfreie Beschriftungsposition ermittelt werden konnte, einbezogen. Die Beschriftung erfolgt anschließend im größten verbleibenden konfliktfreien Bewegungsraum, der ausreichend groß ist, das Schriftobjekt vollständig konfliktfrei aufzunehmen (Abbildung 85, linker Screenshot), andernfalls wird das Objekt nicht beschriftet. Die Platzierung des Schriftobjekts erfolgt in der Mitte des konfliktfreien Bewegungsraums (Abbildung 85 rechter Screenshot), wie in Abbildung 74 b) dargestellt.

Die Signaturen der zu beschriftenden Objekte können bei dem Beschriftungsprozess optional berücksichtigt werden. Dazu werden die Signaturen wie die Beschriftungspositionen von bereits beschrifteten Objekten in das Verfahren einbezogen. Entsprechend verbessert sich die Beschriftungsqualität (Abbildung 84 rechts).

Wie in den Abbildungen zu erkennen, ist die grafische Darstellung der Schriftobjekte auf dem Bildschirm unbefriedigend, was auf java-technische Gründe zurückzuführen ist. Die Schrift wird als Shape, d.h. als Umriß, behandelt, damit sie entsprechend der in Abschnitt 2.1.2 entwickelten Skalierungsfunktion passend zum Maßstab dargestellt werden kann. Durch die Skalierung des Shapes treten erwartungsgemäß Verluste ein, die jedoch durch die rendering engine der Graphics2D-Klasse verstärkt werden. Die *rendering engine* kann als Bindeglied zwischen den (Java) Grafikprimitiven, wie Shapes und Textobjekten, und dem Ausgabegerät, in der Regel dem Bildschirm, angesehen werden [KNU99]. Sowohl für die Transformation von Text in einen Shape, als auch für die Datenstruktur des Shapes, dessen Skalierung und Darstellung werden Standardfunktionen von Java bzw. der Graphics2D-Klasse mit der rendering engine verwendet. Sie ist somit für die Qualitätsverschlechterung (mit-)verantwortlich.

Eine Verbesserung könnte erzielt werden, wenn mit der in Abschnitt 2.1.2 entwickelten Skalierungsfunktion die zum Maßstab passende Schriftgröße bestimmt wird und diese zur

Erzeugung des passend dimensionierten Schriftobjekts verwendet wird. Eigene Tests haben ergeben, dass diese Verfahrensweise zu einer Erhöhung der Laufzeit führt, da die Skalierung und Ausgabe eines Shapes eine geringere Laufzeit hat als die Ausgabe eines Grafiktextes. Auch bei dieser Vorgehensweise, die die beste Qualität liefern sollte, wären Verbesserungen der rendering engine wünschenswert.

Vorbereitend für die Beschriftung linienhafter Objekte wurde ein Verfahren zur Ausrichtung von Schrift an Linienzügen entwickelt, das nicht Bestandteil von Java ist. Ein Beispiel ist in Abbildung 86 dargestellt. Dabei wird sowohl die Beschriftung von Liniensegmenten (Abbildung 86 rechts: Mississippi) als auch von Kurvensegmenten (Abbildung 86 links: Siebengebirge, Hunsrück und Rhein) unterstützt.

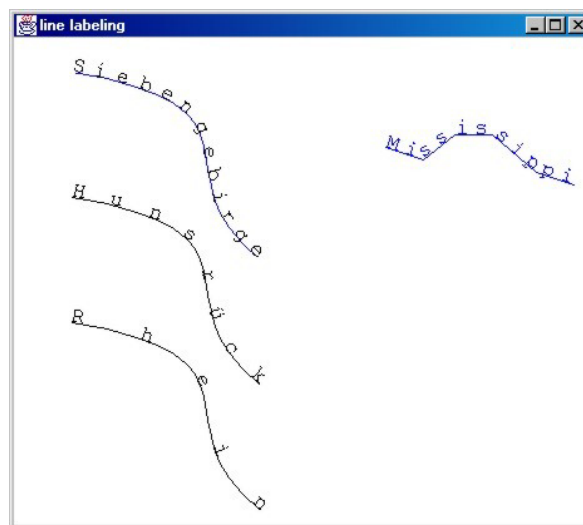


Abbildung 86 Screenshot der entwickelten Komponente zur Beschriftung von Linienzügen mit zugehörigen Standlinien.

Kapitel 7

Resümee und Ausblick

In dieser Arbeit werden Konzepte und Datenstrukturen zur Beschriftung von Bildschirmkarten in Echtzeit entwickelt. Dabei wurden spezifische Eigenschaften und Funktionen von Bildschirmkarten integriert, die sich wesentlich von denen der traditionellen Papierkarte unterscheiden. Gleichzeitig berücksichtigt das Konzept explizit kartographische Regeln, um eine gute Lesbarkeit der Karten zu erzielen. Ein solches Konzept zur Beschriftung von Bildschirmkarten gab es bisher nicht.

Diese Arbeit ist in Kooperation mit dem Lehrstuhl für Kartographie entstanden. Sie nutzt die Ergebnisse von Matthias Ellsiepen [ELL01]. Während bei jener Arbeit kartographische Aspekte der Gestaltung im Vordergrund stehen, geht es hier um die Lösung der dabei auftretenden Probleme unter Nutzung der Methoden der Informatik. Die Aspekte der Kartographie und Informatik, die zum Verständnis dieser Arbeit nötig sind, wurden in Kapitel 2 behandelt, ebenso wie eine kurze Skizzierung der Komplexität der Schriftplatzierung aus Sicht der Informatik.

Existierende Beschriftungsalgorithmen und -systeme wurden in Kapitel 3 diskutiert und eingeordnet. Einige sind für die vollständige Beschriftung von Karten entworfen, andere auf die Beschriftung von punkthaften Objekten beschränkt. Kein bisher bekanntes Verfahren berücksichtigt die typischen Anforderungen und Funktionalitäten, die an Bildschirmkarten und deren Beschriftung gestellt werden. Die für Bildschirmkarten wesentlichen Funktionalitäten sind Zoomen, Scrollen, Zusammenführen und Beschriften heterogener Daten und Kopplung von Beschriften und Weglassen von Beschriftungen unter Berücksichtigung kartographischer Gestaltungsregeln. Die Forderung, dass die Beschriftung in Echtzeit erfolgen soll, verschärft das zu lösende Problem.

Die Entwicklung eines neuartigen Beschriftungskonzepts für Bildschirmkarten erfolgt in den Kapiteln 4 und 5. Dabei wird ausgenutzt, dass Kartenobjekte und die zu beschriftenden Objekte schon vor einer Interaktion des Nutzers bekannt sind. In einer ersten Phase, der so genannten Präprocessingphase, die Kapitel 4 behandelt, können im Vorhinein Informationen gesammelt und berechnet werden.

Die Auswahl von zu beschriftenden Objekten ist im Kontext von Bildschirmkarten und freier Maßstabswahl wichtig, da Kartenmaßstab und Kartenschrift nicht proportional voneinander abhängig sind. Eine bei einem großen Maßstab überlappungsfreie Beschriftung kann bei einem kleinen Maßstab zu Überlappungen führen, so dass eine überlappungsfreie Beschriftung nicht mehr möglich ist. Deshalb wird eine maßstabsabhängige Vorselektion der zu beschriftenden Objekte auf eine hinreichend kleine Obermenge beschriftbarer Objekte vorgenommen. Die Reduktion der Anzahl zu beschriftender Objekte auf ein dem Maßstab angemessenes Maß führt zu einer Verbesserung der Beschriftungsqualität und zu einer Verkürzung der Beschrif-

tungszeit. Durch die maßstabsabhängige Auswahl der zu beschriftenden Objekten wird ein Beitrag zur kartographischen Generalisierung geleistet. In Kapitel 4 werden daher ausführlich drei unterschiedliche Selektionskriterien entwickelt, analysiert und evaluiert. In diese fließen neben dem Darstellungsmaßstab auch die Bedeutung der zu beschriftenden Objekte, ihr Konfliktpotential und ihre räumliche Lage ein. Diese Vorselektion der zu beschriftenden Objekte wurde bei automatischen Beschriftungsverfahren bisher nicht vorgenommen.

Die Anfrage des Nutzers nach einer Bildschirmkarte erfolgt in einer zweiten Phase, der Interaktionsphase, die in Kapitel 5 beschrieben ist. Dabei spezifiziert der Nutzer, welchen Kartenausschnitt zu welchem Maßstab er visualisiert bzw. beschriftet haben möchte. Um diese Anfrage in Echtzeit zu beantworten wird auf das Wissen der ersten Phase zurückgegriffen und darauf aufbauend die geforderte individuelle Beschriftung erzeugt.

Entscheidend für die Laufzeit der zweiten Phase ist eine effiziente Datenstruktur, der reaktive Konfliktgraph, der für ein großes Maßstabsintervall gilt und in der ersten Phase generiert wird. Er unterstützt dabei die Funktionen Zoomen und Scrollen. Kartographisches Wissen in Form von formalisierten Beschriftungsregeln, wie überlappungsfreies Beschriften, bevorzugte Beschriftungspositionen, Vermeidung von Informationsverlusten durch Verdeckungen und die beschriebene Vorselektion an zu beschriftenden Objekten, fließen in den reaktiven Konfliktgraphen ein und werden explizit in der zweiten Phase berücksichtigt.

Im Unterschied zu den meisten automatischen Beschriftungskonzepten ist in dem entwickelten Konzept die Beschriftung nicht auf vorgegebene diskrete Beschriftungspositionen beschränkt, sondern auf einen Beschriftungsraum, wie es den traditionellen kartographischen Beschriftungsmethoden entspricht. Diese Modellierung wird auch als „sliding labels“ bezeichnet.

Die entwickelte Datenstruktur ist aus technischer Sicht „maßstabsunabhängig“, jedoch im Hinblick auf kartographische Generalisierungsprozesse auf Maßstabsintervalle eingeschränkt. In Abschnitt 4.5 wird diese Thematik diskutiert und es werden dazu passende Lösungen vorgestellt.

Durch die Zweiteilung des Beschriftungsprozesses und deren Vorverarbeitung in der ersten Phase ist das Verfahren bei Kombinationen von thematischen (Beschriftungs-) Layern und Änderungen von Kartenobjekten zunächst unflexibel. Durch das in Abschnitt 5.4 entwickelte Verfahren können Beschriftungen sich überschneidender Kartenausschnitte während der Interaktionsphase homogen zu einer neuen Beschriftung zusammengeführt werden.

Neben der Darstellung der entwickelten Verfahren in Pseudocode erfolgt eine analytische Betrachtung der Laufzeiten. Darüber hinaus wurde in Kapitel 6 ein Prototyp entwickelt in dem die wesentlichen Konzepte umgesetzt sind und an dem man das Verhalten in der Praxis studieren kann. Der Prototyp zeigt insbesondere, dass eine Beschriftung in Echtzeit praktisch möglich ist. Er fokussiert auf die Beschriftung punkthafter Objekte. Die Schnittstellenkomponente ermöglicht den Datenimport von kommerziellen GIS, wie ArcGIS der Firma ESRI. Die eingelesenen Objekte werden automatisch klassifiziert und signaturiert. Dies ermöglicht eine kartographisch gerechte Darstellung von punkt-, linien- und flächenhaften Kartenobjekten mit der entwickelten Visualisierungskomponente. Die Implementierung des Prototypen erfolgte in Java und hat einen Umfang von ca. 18.000 Programmzeilen.

Weiterer Forschungsbedarf besteht bei der Beschriftung von flächenhaften Objekten und ihrer Integration in eine Gesamtbeschriftung, wobei auf Ergebnisse dieser Arbeit aufgebaut werden kann. Für punkthafte und linienhafte Objekte wurden in dieser Arbeit eigene Beschriftungsräume entwickelt. Der Beschriftungsraum für flächenhafte Objekte wurde auf den von linienhaften reduziert, da es ansonsten zu einer drastischen Zunahme der potentiellen Überlappungen kommt [4.3.9]. Eine Beschriftung flächenhafter Objekte wird so ermöglicht, jedoch

ist die Lage der Schrift eingeschränkt. Auch bei der Variation des Abstands zwischen den Buchstaben und der Zulassung von Kreuzungen mit anderen Schriftobjekten zwischen den Buchstaben sind weitere Untersuchungen nötig. In diesem Kontext stellt sich die Frage der Reihenfolge bzw. der Priorität der Beschriftungen der unterschiedlichen Geometrietypen (punkt-, linien- und flächenhaft). Imhof schlägt dazu vor: „[Platziere] zuerst die großen, sich über ein größeres Gebiet hinziehenden Namen und passe dann die kleinen Namen ein.“ [IMH62] Wie diese Maxime in das hier entwickelte Verfahren integriert werden kann, ist Gegenstand zukünftiger Forschung.

Diese Arbeit behandelt das Problem der Auswahl und Änderung von Beschriftungen, die durch die Funktion Zoomen bei Bildschirmkarten hervorgerufen werden. Dies ist verwandt zum Problem der kartographischen Generalisierung. Zoomen löst neben einer neuen Beschriftung auch Generalisierungsvorgänge aus. Dabei entstehen durch die Verknappung des Raums graphische Zwänge, die zum Weglassen oder zur Verdrängung von Kartenobjekten führen können. In der Kartenbeschriftung entspricht dies dem Weglassen von Beschriftungen, für das diese Arbeit Kriterien bereitstellt (Abschnitt 4.3.3.4). Zur Verdrängung in der kartographischen Generalisierung gibt es Analogien zur Auswahl einer der möglichen Beschriftungspositionen. Anschaulich kann jedem Kartenobjekt ein maximaler „Bewegungsraum“ zugeordnet werden, der analog zu dem in dieser Arbeit entwickelten Konzept des Beschriftungsraums ist. Im Unterschied zu den Ansätzen der kartographischen und auch der Modellgeneralisierung fordern Bildschirmkarten kontinuierliche Maßstabsübergänge und nicht diskrete.

Es ist zu untersuchen, ob eine Weiterentwicklung des hier vorgestellten Konzepts des reaktiven Konfliktgraphen nicht auch bei dieser wesentlich allgemeineren Fragestellung verwandt werden kann. Die Eigenschaft der maßstabsunabhängigen Repräsentation von Konflikten könnte sowohl zum Weglassen als auch zur Verdrängung von Kartenobjekten genutzt werden.

Glossar

Bereich, konfliktfreier – linienhaftes Objekt (Abschnitt 4.3.6, S. 79):

Der *konfliktfreie Bereich* (*conflictFree*) eines linienhaften Objekts sichert zu, dass die zu platzierende Labelbox keinen Konflikt bezüglich der Höhe hat. Die Labelbox kann breiter als der konfliktfreie Bereich sein.

Beschriftungsposition (Abschnitt 5.2, S. 116):

Die *Beschriftungsposition* ist eine ausgewählte Beschriftungsstelle innerhalb des Beschriftungsraums, an der das zu beschriftende Objekt beschriftet ist.

Beschriftungsraum (Abschnitt 4.3.1.1, S. 34):

In Abhängigkeit der Beschriftungsklasse wird der *Beschriftungsraum* definiert. Dieser hat die folgenden Eigenschaften:

- er ist die konvexe Hülle aller möglichen Beschriftungspositionen, und
- er ist ein einfach berechenbares geometrisches Objekt

(punkthafte Objekte Abschnitt 4.3.1.1, S. 33; linienhafte Objekte Abschnitt 4.3.4.1, S. 73; flächenhafte Objekte Abschnitt 4.3.9, S. 92; Siedlungsflächen Abschnitt 4.3.10, S. 95)

Beschriftungsschwierigkeit (Abschnitt 4.3.3, S. 42):

Die *Beschriftungsschwierigkeit* beschreibt, wie schwer es ist, ein Objekt zu einem vorgegebenen Maßstab und unter Berücksichtigung einer Menge von Konfliktpartnern zu beschriften. Dies kann entweder über eine Bewertung der Konflikte oder über die verbleibenden Beschriftungsmöglichkeiten erfolgen.

(punkthafte Objekte Abschnitt 4.3.3, S. 42; linienhafte Objekte Abschnitt 4.3.6, S. 79)

Bewegungsraum – punkthafte Objekt (Abschnitt 4.3.3.3.1, S. 55):

Der *Bewegungsraum* eines punkthaften zu beschriftenden Objekts ist der Bereich der zulässigen Beschriftungspositionen ohne Berücksichtigung von Konflikten, in dem die Labelbox innerhalb des Beschriftungsraums an seiner Innenseite „entlanggleiten“ kann. Es gibt zwei horizontale Bewegungsräume unter der oberen Grenze bzw. über der unteren Grenze des Beschriftungsraums und zwei vertikale links der rechten bzw. rechts der linken Grenze des Beschriftungsraums.

Bewegungsraum, konfliktfreier (Abschnitt 4.3.3.3.2, S. 63):

Der *konfliktfreie Bewegungsraum* eines zu beschriftenden punkthaften Objekts ist der Bereich des Bewegungsraums, der keinen Konflikt besitzt.

Im Unterschied zu einer konfliktfreien Beschriftungsposition garantiert ein konfliktfreier Bewegungsraum nicht die vollständige konfliktfreie Platzierung der Labelbox in diesem. Der horizontale konfliktfreie Bewegungsraum sichert zu, dass die zu platzierende Labelbox keinen Konflikt bezüglich der Höhe mit dem Raum hat. Die Breite des horizontalen Beschriftungsraums kann kleiner als die der Labelbox sein. Analog sichert der vertikale konfliktfreie Beschriftungsraum die konfliktfreie Platzierung der Labelbox bezüglich der Breite zu. Über die Höhe wird keine Zusicherung getroffen.

Bewegungsraum, Bewertung – *value* – einfacher Fall (Abschnitt 4.3.3.3.1, S. 58):

Für jeden der vier Bewegungsräume eines punkthaften Objekts wird in Abhängigkeit der Bewegungsrichtung, seiner Lage und seinen Konflikten der Wert *value* bestimmt. Durch eine Normierung (Division durch die Labelboxhöhe bzw. -breite) können auch vertikale und horizontale Bewegungsräume miteinander verglichen werden.

Bewegungsraum, Bewertung – *value* – verallgemeinerter Fall (Abschnitt 4.3.3.3.2, S. 66):

Für jeden Bewegungsraum werden die konfliktfreien Bewegungsräume normiert durch $value_{1,m}$ für den oberen horizontalen, $value_{2,n}$ für den linken vertikalen, $value_{3,o}$ für den unteren horizontalen und $value_{4,p}$ für den rechten vertikalen Bewegungsraum bewertet. Durch eine Normierung (Division durch die Labelboxhöhe bzw. -breite) können alle Werte, unabhängig ob sie einen vertikalen oder horizontalen konfliktfreien Bewegungsraum repräsentieren, miteinander verglichen werden.

Deselektionsmaßstab einer Konfliktgraphkante – `lowerBound` (Abschnitt 4.3.2.2, S. 41):

Der Deselektionsmaßstab der Konfliktgraphkante ist der Maßstab, ab und unter dem der Konflikt nicht mehr auftritt, da einer der beiden Konfliktpartner (Konfliktgraphknoten) nicht mehr beschriftet wird. Er ergibt sich aus dem größeren der beiden Deselektionsmaßstäbe der inzidenten Konfliktgraphknoten.

(punkthaft-punkthaft-Konflikt Abschnitt 4.3.2.2, S. 40; linienhaft-linienhaft-Konflikt Abschnitt 4.3.5.2, S. 79; punkthaft-linienhaft-Konflikt Abschnitt 4.3.7.2, S. 91)

Deselektionsmaßstab eines Konfliktgraphknoten (Abschnitt 4.3.3, S. 42):

Der Deselektionsmaßstab eines Konfliktgraphknoten ist der Maßstab, ab und unterhalb dessen ein zu beschriftendes Objekt nicht mehr beschriftet wird.

Gebiete – punkthafte Objekte (Abschnitt 4.3.3.3.2, S. 61):

Der Beschriftungsraum eines punkthaften Objekts unterteilt sich in *Gebiete*, die maximal die Größe der Labelbox besitzen. In den Ecken des Beschriftungsraums liegen Gebiete, welche die Größe der Labelbox haben. Die verbleibenden horizontalen Flächen und die vertikalen Flächen zwischen diesen Gebieten werden mit Gebieten, welche die Größe der Labelbox haben, aufgefüllt. Verbleibende Lücken werden, wenn vorhanden mit Gebieten aufgefüllt, die kleiner als die Labelbox sind, sodass die Fläche zwischen Signatur und der Grenze des Beschriftungsraums in Gebiete eingeteilt ist.

Konflikt und Informationsverlust (Abschnitte 4.6, 5.2 und 5.3.2, S. 105, 116 und 134):

Ein *Konflikt* ist eine Überlappung zwischen zwei Konfliktpartnern. Ein *potentieller Konflikt* ist eine Überlappung von Beschriftungsräumen, d.h. von möglichen Beschriftungspositionen. Ein (*echter*) *Konflikt* ist die Überlappung von Beschriftungspositionen, also die tatsächliche Überlappung von Schrift. Während des Beschriftungsprozesses (Abschnitt 5.2, S. 116) kann es zu einer Überlappung von einer Beschriftungsposition und einem Beschriftungsraum kommen, der als spezieller potentieller Konflikt betrachtet wird. Die Überlappung eines Kartenobjekts, das keine Beschriftungsposition und kein Beschriftungsraum ist, durch einen Beschriftungsraum oder einer Beschriftungsposition, wird als *Verdeckung* bezeichnet. Dabei können *Informationsverluste* eintreten (Abschnitte 4.6 und 5.3.2, S. 105 und 134).

(Beschriftungskonflikt zwischen zwei punkthaften Objekten Abschnitt 4.3.2, S. 38; zwischen zwei linienhaften Objekten Abschnitt 4.3.5, S. 77; zwischen punkthaften und linienhaften Objekten Abschnitt 4.3.7, S. 82)

Konfliktgebirge (Abschnitt 4.3.3.2, S. 48):

Das *Konfliktgebirge* ist der in dem dreidimensionalen Raum eingebettete statische Konfliktgraph. Die Lage der Konfliktgraphknoten ergibt sich aus den Koordinaten der Geoobjekte und die Höhe aus seinen Attributen Priorität und Beschriftungsschwierigkeit. Die Einbettung einer Konfliktgraphkante ergibt sich durch Interpolation aus Lage, Höhe und Beschriftungsraum der Konfliktgraphknoten.

Konfliktgraph (Abschnitt 4.2, S. 32):

Der *Konfliktgraph* ist ein Graph, der aus Knoten und ungerichteten und schlingenfreien Kanten besteht. Die Konfliktgraphknoten repräsentieren die zu beschriftenden Objekte und die Konfliktgraphkanten die potentiellen Konflikte. Die Knoten sind mit dem Deselektionsmaßstab attribuiert und die Kanten mit den Intervallgrenzen *lowerBound* und *upperBound*. Die *lowerBound* ergibt sich aus dem Maximum der Deselektionsmaßstäbe der inzidenten Knoten und die *upperBound* aus dem Schnittmaßstab der Knoten.

Dieser maßstabsunabhängige Konfliktgraph wird reaktiver Konfliktgraph genannt im Unterschied zu dem statischen Konfliktgraphen, der zu einem spezifischen Maßstab gilt.

Konfliktrechteck (Abschnitt 4.3.3.3.1, S. 54):

Das *Konfliktrechteck* ist die Schnittfläche zweier Beschriftungsräume.

Konfliktsektor (Abschnitt 4.3.3.2, S. 49):

Der *Konfliktsektor* eines Konfliktknotens bezüglich einer Konfliktkante ist der Schnitt des kleinsten Sektors des Knotens, der die Überlappungsfläche der Beschriftungsräume der beiden zur Kante inzidenten Knoten enthält, mit dem Beschriftungsraum des Knotens.

Labelbox (Abschnitt 2.1.1.1, S. 6):

Die Approximation der Referenzbeschriftung durch ein Rechteck ergibt die *Referenzlabelbox* (Abschnitt 2.1.1.1, S. 6). Die *Labelbox* für einen beliebigen Maßstab ergibt sich aus der Skalierung der Referenzlabelbox mit dem Referenzmaßstab mit der Skalierungsfunktion (Abschnitt 4.3.1.1/S.33).

Objekt, zu beschriftendes (Abschnitt 2.1.1, S. 5):

Das *zu beschriftende Objekt* wird mit einem Schriftzug, der aus seinen Attributen abgeleitet werden kann, beschriftet. In Abhängigkeit von seiner Signaturausdehnung wird zwischen punkt-, linien- und flächenhaften Objekten differenziert. Innerhalb dieser Klassen kann es zu Beschriftungs Sonderfällen kommen. Mit dem Konfliktgraph enthält jedes zu beschriftende Objekt einen Deselektionsmaßstab.

(punkthafte Objekte Abschnitt 4.3.1, S. 33; linienhafte Objekte Abschnitt 4.3.4, S. 72; flächenhafte Objekte Abschnitt 4.3.9, S. 92; Grundlagen Abschnitt 2.1.1, S. 5)

Quadrant – punkthafte Objekte (Abschnitt 4.3.3.3.1, S. 53):

Der Beschriftungsraum eines punkthaften Objekts ist, unter der Annahme, dass die Signatur keine Ausdehnung hat und es keinen Abstand zwischen Signatur und Beschriftungsposition gibt, in vier *Quadranten* zerlegbar. Eine Ecke jedes Quadranten liegt im

Mittelpunkt des Beschriftungsraums und die diagonale Ecke auf einer Ecke des Beschriftungsraums. Die Breite jedes Quadranten entspricht der Breite der zu platzierenden Labelbox nach Definition des Beschriftungsraums und die Höhe des Quadranten der Höhe der Labelbox.

Referenzbeschriftung, Referenzmaßstab und Skalierungsfunktion (Abschnitt 4.1.3, S.31):

Beschriftungen sind „künstliche“ Objekte in Karten und können daher nicht direkt aus diesen abgeleitet werden. Für die zu beschriftenden Objekte wird daher in Abhängigkeit ihrer Attribute zu einem spezifischen Maßstab anhand einer Musterbeschriftung die Schriftart, -höhe, -farbe, Sperrung etc. festgelegt. Dies bildet die *Referenzbeschriftung* zum *Referenzmaßstab* (Abschnitt 4.1.3, S. 31). Damit Mindestgrößen nicht unterschritten werden, erfolgt die Skalierung der Referenzen nicht proportional sondern exponentiell zum Maßstab mit der *Skalierungsfunktion* (Abschnitt 2.1.2, S. 9).

Schnittmaßstab (Abschnitt 4.3.2.1, S. 39):

Der *Schnittmaßstab* ist der Maßstab, ab und unterhalb dem es zu einem Konflikt zwischen den Beschriftungsräumen zweier zu beschriftender Objekte kommt.

(zwischen zwei punkthaften Objekten Abschnitt 4.3.2.1, S. 39; zwischen zwei linienhaften Objekten Abschnitt 4.3.5.1, S. 78; zwischen punkthaften und linienhaften Objekten Abschnitt 4.3.7.1, S. 83)

Sektor (Abschnitt 4.3.3.2, S. 49):

Der *Sektor* eines Knotens ist die Fläche zwischen zwei Halbgeraden, die in dem Knoten beginnen.

Winkel, größter freier (Abschnitt 4.3.3.2, S. 49):

Der *größte freie Winkel* eines Konfliktgraphknotens ist der Winkel des größten Sektors, der disjunkt zu allen Konfliktsektoren ist.

\bar{x} - und \bar{y} -Vektor (Abschnitt 4.3.3.3.1, S. 54):

Die \bar{x} - und des \bar{y} -Vektoren beschreiben (Teile) konfliktfreier Bewegungsräume von punkthaften Beschriftungsräumen.

$\bar{x}_1 \square \bar{x}_2$ und $\bar{y}_1 \square \bar{y}_2$ (Abschnitt 4.3.3.3.1, S. 56):

Vereinigung der Flächen von entweder zwei \bar{x} - oder zwei \bar{y} - Vektoren mit gleichem Ursprung. Die Höhe der $\bar{x}_1 \square \bar{x}_2$ -Fläche ist gleich der Höhe der Labelbox, nach Definition des \bar{x} -Vektors, und die Breite gleich der Summe der x-Koordinaten der beiden Vektoren. Die Breite der $\bar{y}_1 \square \bar{y}_2$ -Fläche ist gleich der Breite der Labelbox, nach Definition des \bar{y} -Vektors, und die Höhe ist gleich der Summe der y-Koordinaten der beiden Vektoren. Ist die resultierende Fläche leer, dann wird das durch das Symbol „ \emptyset “ repräsentiert.

Literaturverzeichnis

- [AHF83] Ahn, J., Freeman, H., *A Program for Automatic Name Placement*, in Proc. Auto-Carto 6, S. 444-453, 1983.
- [AHF84a] Ahn, J., Freeman, H., *AUTONAP - An Expert System for Automatic Map Name Placement*, In Proc. Int. Symp. Spatial Data Handling (SDH'84), S. 544-569, 1984.
- [AHF84b] Ahn, J., Freeman, H., *A Program for Automatic Name Placement*, Cartographica, 21(2-3):101-109, 1984.
- [AKS97] Agarwal, P. K., van Kreveld, M., Suri, S., *Label placement by maximum independent set in Rectangles*, in Proceedings of the 9th Canadian Conference on Computational Geometry (CCCG'97), S. 233-238, 1997.
- [ARV91] Arvo, J., *Graphics Gems II*, Academic Press, San Diego, 1991.
- [BKK96] Berchtold, S., Keim, D. und Kriegel, H.-P., *The X-tree: An Index Structure for High-Dimensional Data*, Proc. of the 22nd VLDB Conference, Institute for Computer Science, Universität München, 1996.
- [BKO97] Berg, M. de, Kreveld, M., Overmars, M., Schwarzkopf, O., *Computational Geometry – Algorithms and Applications*, Springer Verlag, 1997.
- [BKS90] Beckmann, N., Kriegel, H.-P., Schneider, R. und Seeger, B., *The R*-tree: An Efficient and Robust Access Method for Points and Rectangles*, Proc. of ACM SIGMOD Conference on Management of Data, S. 322-331, Institut für Praktische Informatik, Universität Bremen, 1990.
- [CAR87] Carstensen, L. W., *A comparison of simple mathematical approaches to the placement of spot symbols*, Cartographica, Bd. 24, Nr. 23, 1987, S. 46-63.
- [CHR95] Christensen, J., *Managing Designs Complexity: Using Stochastic Optimization in the Production of Computer Graphics*, Ph.D.-Thesis, Center for Research in Computing Technology, Harvard University, Technical Report tr-10-95, Cambridge, Massachusetts, USA, Juni 1995.
- [CLR90] Cormen, T., Leiserson, C. E., Rivest, R. L., *Introduction to Algorithms*, McGraw-Hill, MIT-Press, 1990.
- [CLS01] Claus, V., Schwill, A., *Duden „Informatik“*, Dudenverlag, 2001.
- [CMS92] Christensen, J., Marks, J., Shieber, S., *Labeling Point Features on Maps and Diagrams*, Technical Report tr-25-92, Harvard University, Cambridge, Massachusetts, USA, 1992.
- [CMS95] Christensen, J., Marks, J., Shieber, S., *An Empirical Study of Algorithms for Point-Feature Label Placement*, ACM Transactions on Graphics, Juli 1995.

- [COJ90] Cook, A. C., Jones, C. B., *A Prolog Rule-Based System for Cartographic Name Placement*, Computer Graphics Forum 9, 1990, S. 109-126.
- [COO88] Cook, A. C., *Automated cartographic name placement using rule-based systems*, Ph.D.-Thesis, Department of Mathematics and Computer Science, Polytechnic of Wales, Pontypridd, Mid Glamorgan, UK, September 1988.
- [DIE76] *Diercke Weltatlas*, 184. Auflage, Georg Westermann Verlag, Braunschweig, 1976.
- [DIE96] *Diercke Weltatlas – Ausgabe 2*, 4. Auflage, Westermann Schulbuchverlag, Braunschweig, 1996.
- [DKS99] van Dijk, S., van Kreveld, M., Strijk, T., Wolff, A., *Towards an evaluation of quality for label placement methods*, in Proc. 19th Int. Cartographic Conf. (ICA'99), S. 905-913, Ottawa, 14-21 August 1999.
- [DOE01] Dörschlag, D., *Automatische Platzierung von Objekten in Flächen einer Karte*, Diplomarbeit am Institut für Kartographie und Geoinformation, Universität Bonn, 2001.
- [DPP03] Dörschlag, D., Petzold, I., Plümer, L., *Placing objects automatically in areas of maps*, Proceedings of the Durban International Cartographic Conference ICC03, International Cartographic Association (ICA), Durban, August 2003.
- [ECM96] Edmondson, S., Christensen, J., Marks, J., Shieber, S., *A General Cartographic Labeling Algorithm*, Manuscript, Harvard University, Cambridge, Massachusetts, 1996.
- [EGH90] Egenhofer, M. J., Herring, J. R., *A mathematical framework for the definition of topological relationships*, Proceedings of the 4th International Symposium on Spatial Data Handling, Zürich, Schweiz, International Geographical Union, 1990.
- [ELL01] Ellsiepen, M., *Formalisierung kartographischen Wissens zur Schriftplatzierung in topographischen Karten*, Dissertation, Schriftenreihe des Instituts für Kartographie und Geoinformation der Rheinischen Friedrich-Wilhelms-Universität Bonn, 2001.
- [ESR93] ESRI, *ArcDeutschland'500*, 1993.
- [ESR98] ESRI, *Maplex – Automatic Cartographic Name Placement Software*, ESRI White Paper, <http://www.esri.com/library/whitepapers/pdfs/maplexwp.pdf>, August, 1998.
- [FOW91] Formann, M., Wagner, F., *A packing problem with applications to lettering of maps*, Proceedings of the 7th Annual Symposium on Computational Geometry, North Conway, New Hampshire, Juli 1991, S. 281-288.
- [FRE10] Fretwurst, A., *Die Kartenschrift*, Konrad Wittwer Verlag, Stuttgart, 1910.

- [FRE91] Freeman, H., *Computer name placement*, In: Maguire, D. J., Rhind, D. W. Geographical information systems: Principles and applications, Longman Scientific and Technical, 1991.
- [GLA90] Glassner, A., *Graphics Gems*, Academic Press, San Diego, 1990.
- [GUT84] Guttman, A., R-trees: *A Dynamic Index Structure for Spatial Searching*, Proc. of ACM SIGMOD Conference on Management of Data, S. 47-57, University of California, Berkeley, 1984.
- [HGM02] Hake, G., Grünreich, D., Meng, L., *Kartographie*, deGruyter, 2002.
- [HEB98] Heber, M., *Vorausberechnung reaktiver Datenstrukturen zur schnellen Beschriftung von Landkarten*, Diplomarbeit am Institut für Informatik, Universität Bonn, 1998.
- [HIR80] Hirsch, S. A., *Algorithms for Automatic Name Placement of Point Data*, MSc-Thesis, Department of Geography, State University of New York, Buffalo, 1980.
- [HIR82] Hirsch, S. A., *An Algorithms for Automatic Name Placement around point Data*, The American Cartographer, 9(1):5-17, 1982.
- [IMH62] Imhof, E., *Die Anordnung der Namen in der Karte*, Internationales Jahrbuch für Kartographie II, 1962.
- [JON89] Jones, C. B., *Cartographic Name Placement with Prolog*, IEEE Computer Graphics & Applications, Bd. 9, Nr. 5, September 1989, S. 36-47.
- [JON90] Jones, C. B., *Conflict resolution in cartographic name placement*, Computer Aided Design, Bd. 22, Nr. 3, April 1990, S. 173-183.
- [KAH01] Kahn, T., *Beschriftung von Linien und Flächen in Bildschirmkarten*, Diplomarbeit am Institut für Kartographie und Geoinformation, Universität Bonn, 2001.
- [KGV83] Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P., *Optimization by Simulated Annealing*, Science, Bd. 220, Nr. 4598, Mai 1983, S. 671-680.
- [KNI98] Knipping, L., *Beschriftung von Linienzügen*, Diplomarbeit im Fachbereich Mathematik und Informatik und Informatik, Freie Universität Berlin, 1998.
- [KNU99] Knudsen, J., *Java 2D Graphics*, O'Reilly, 1999.
- [KRE94] Kresse, W., *Plazierung von Schrift in Karten*, Dissertation, Heft 23 der Schriftenreihe des Instituts für Kartographie und Topographie der Rheinischen Friedrich-Wilhelms-Universität Bonn, 1994.
- [KSW98] van Kreveld, M., Strijk, T., Wolff, A., *Point Set Labeling with Sliding Labels*, In Proc. 14th Annu. ACM Sympos. Comput. Geom. (SoCG'98), S. 337-346, Minneapolis, 7-10 Juni 1998.
- [KSW99] van Kreveld, M., Strijk, T., Wolff, A., *Point Labeling with Sliding Labels*, Computational Geometry, Theory and Applications, 13:21-47, 1999.

- [LJF95] Lin, K.-I., Jagadish, H. V. und Faloutsos, C., *The TV-tree - an index structure for high-dimensional data*, Proc. of the VLDB Conference, S. 517-542, Department of Computer Science und Institute for Systems Research, University of Maryland, 1995.
- [LUT97] Lutterbach, D., *Auswirkung der Bildschirm-Visualisierung auf die kartographische Darstellung der raumbezogenen Planung*, Dissertation, Heft 24 der Schriftenreihe des Instituts für Kartographie und Topographie der Rheinischen Friedrich-Wilhelms-Universität Bonn, 1997.
- [MAL98] Malic, B., *Physiologische und technische Aspekte kartographischer Bildschirm-visualisierung*, Dissertation, Heft 25 der Schriftenreihe des Instituts für Kartographie und Topographie der Rheinischen Friedrich-Wilhelms-Universität Bonn, 1998.
- [MAN89] Manber, U., *Introduction to Algorithms – A Creative Approach*, Addison-Wesley, 1989.
- [MAS93] Marks, J., Shieber, S., *The Computational Complexity of Cartographic Label Placement*, Technical Report tr-05-91, Harvard Univeristy, Cambridge, Massachusetts, 1991/1993.
- [MIC01] Microsoft, *AutoRoute 2002*, Redmond, 2001.
- [MUS93] *Musterblatt für die Topographische Karte 1:25000*, herausgegeben vom LVA NRW, Bad Godesberg, 1993.
- [NAT95] National Geographic Society, *The World – Political*, Washington, März 1995.
- [NEU01] Neudeck, S., *Zur Gestaltung topografischer Karten für die Bildschirm-visualisierung*, Dissertation, Schriftenreihe Studiengang Geodäsie und Geo-information Universität der Bundeswehr München, 2001.
- [NIL80] Nilsson, N.J., *Principles of Artificial Intelligence*, Springer Verlag, 1980.
- [OES98] Oestereich, B., *Objektorientierte Softwareentwicklung – Analyse und Design mit der Unified Modeling Language*, R. Oldenbourg Verlag, München, 1998.
- [OTW93] Ottmann, T., Widmayer, P., *Algorithmen und Datenstrukturen*, BI-Wissenschaftsverlag, Reihe Informatik, Bd. 70, 1993.
- [PEP97] Petzold, I., Plümer, L., *Plazierung der Beschriftung in dynamisch erzeugten Bildschirmkarten*, Nachrichten aus dem Vermessungswesen, Reihe I, Heft Nr. 117, Verlag des Instituts für Angewandte Geodäsie, Frankfurt a. M., 1997.
- [PET96] Petzold, I., *Textplatzierung in dynamisch erzeugten Karten*, Diplomarbeit am Institut für Informatik, Universität Bonn, 1996.
- [PGP03] Petzold, I., Gröger, G., Plümer, L., *Fast Screen Map Labeling – Data-Structures and Algorithms*, Proceedings of the Durban International Cartographic Conference ICC03, International Cartographic Association (ICA), Durban, August 2003.

- [PIT64] Pillwitzer, W., Töpfer, F., *Das Auswahlgesetz, ein Mittel zur kartographischen Generalisierung*, Kartographische Nachrichten, 14. Jahrgang, Heft 4, C. Bertelsmann Verlag, Gütersloh, 1964.
- [PPH99] Petzold, I., Plümer, L., Heber, M., *Label Placement for dynamically generated screen maps*, Proceedings of the Ottawa International Cartographic Conference ICC99, International Cartographic Association (ICA), Canada, 1999.
- [PRS85] Preparata, F. P., Shamos, M. I., *Computational Geometry: An Introduction*, Springer Verlag, 1986.
- [QWX00] Qin, Z., Wolff, A., Xu, A., Zhu, B., *New algorithms for two-label point labelling*, in Proc. 8th Annu. Europ. Symp. on Algorithms (ESA'00), volume 1879 of Lecture Notes in Computer Science, Springer Verlag, S. 368-379, Saarbrücken, 5.-8. September 2000.
- [RJB99] Rumbaugh, J., Jacobson, I., Booch, G., *The Unified Modeling Language – Reference Manual*, Addison-Wesley, 1999.
- [RMN98] Rand McNally, *Road Atlas*, Skokie, Illinois, 1998.
- [ROL85] Roussopoulos, N. und Leifker, D., *Direct Spatial Search on Pictorial Databases Using Packed R-trees*, Proc. of ACM SIGMOD Conference on Management of Data, S. 17-31, Department of Computer Science, University of Maryland, 1985.
- [RSV02] Rigaux, P., Scholl, M., Voisard, A., *Spatial Databases - With Application to GIS*, Academic Press, Morgan Kaufmann Publisher, 2002.
- [SAM90] Samet, H., *The Design and Analysis of Spatial Data Structures*, Addison-Wesley, 1990.
- [SCH86] Schrijver, A., *Theory of Linear and Integer Programming*, Wiley, New York, 1986.
- [SCH95] Schmidt, V. A., *Reine Forschung, praktische Resultate*, Die Zeit, Nr. 18, 28. April 1995, S. 45-46.
- [SRF87] Sellis, T., Roussopoulos N. und Faloutsos, C., *The R+-tree: A Dynamic Index for Multi-dimensional Objects*, Proc. of the 13th VLDB Conference, Department of Computer Science, University of Maryland, 1987.
- [STK99a] Strijk, T., van Kreveld, M., *Labeling a Rectilinear Map More Efficiently*, Information Processing, Letters, 69(1):25-30, 1999.
- [STK99b] Strijk, T., van Kreveld, M., *Practical extensions of point labeling in the slider model*, In Proc. 7th ACM Symposium on Advances in Geographic Information Systems, S. 47-52, Kansas City, 5-6 November 1999.
- [STR01] Strijk, T., *Geometric Algorithms for Cartographic Label Placement*, PhD thesis, Utrecht University, Department of Computer Science, Januar 2001.
- [STW01] Strijk, T., Wolff, A., *Labeling points with circles*, International Journal of Computational Geometry and Applications, 11(2), S.181-195, April 2001.

- [TWX00] Thon, M., Wolff, A., Xu, Y., *Ein neuer Algorithmus zur Beschriftung von Punkten mit je zwei Kreisen*, In Gesellschaft für Informatik e.V. (Hrsg.), Tagungsband der Informatiktag 00, Neues Kloster, Bad Schussenried, 27-28 Oktober 2000.
- [WAG94] Wagner, F., *Approximate map labeling is in $\Omega(n \log n)$* , Information Processing Letters, 52(3):161-165, 1994.
- [WAW95] Wagner, F., Wolff, A., *Map Labeling Heuristics: Provably Good and Practically Usefull*, Proceedings of the 11th Annual Symposium on Computational Geometry, ACM Press, Vancouver, B.C., Canada, 1995, S. 109-118.
- [WAW97] Wagner, F., Wolff, A., *A Practical Map Labeling Algorithm*, Computational Geometry, Theory and Applications, 7:387-404, 1997.
- [WES96] Westermann, *Handkarte Deutschland – physikalisch und politisch*, Westermann Schulbuchverlag, Braunschweig, 1996.
- [WOL99] Wolff, A., *Automated Label Placement in Theory and Practice*, PhD thesis, Fachbereich Mathematik und Informatik, Freie Universität Berlin, Mai 1999.
- [WOR95] Worboys, M. F., *GIS: A Computing Perspective*, Taylor & Francis, UK, 1995.
- [WTX02] Wolff, A., Thon, M., Xu, Y., *A simple factor-2/3 approximation algorithm for two-circle point labelling*, International Journal of Computational Geometry and Applications, 12(4):269-281, 2002.
- [YOE72] Yoeli, P., *The logic of automated map lettering*, The Cartographic Journal, Bd. 9, Nr. 2, 1972, S. 99-108.
- [ZOR86] Zoraster, S., *Integer programming applied to the map label placement problem*, Cartographica, Bd. 23, Nr. 3, 1986, S. 16-27.

Index

- #
- $\bar{x}_1 \square \bar{x}_2$ **56**
- A**
- adjazent 13
- Anfrage 111, 112
- Anfragebereich 114
- Anfragemastab 114
- Auflösung 32
- Aufspieß-Problem 114
- Auswahlalgorithmus 30, 70, 71, 116, 127, 132, 148
- B**
- Beschriftungskonflikt 33, 41, 96
- Beschriftungslayer 137
- Beschriftungsposition 71, 116, 121, 148
konfliktfrei 56, 60, 118
- Beschriftungspositionsbewertung 7, 125
- Beschriftungsraum 27, 29, **34**, 106, 109, 116, 118, 143
flächenhaft 8, 95
konfliktfrei 71
linienhaft 8, 73, **75**
punkthaft 6, **34**, 53, 61
- Beschriftungsschwierigkeit 42, 52, 60, 63, 72, 96, 98, 109, 143, 146
Bewertung 69
kombinierte Kriterien 72
linienhaft 79
punkthaft 42, 67, 91
- Bewegungsraum **55**, 143
Bewertung **58**, 67
konfliktfrei **63**, 148
- Bewegungsrichtung 62, 64
- Bewertung **58**
Beschriftungsschwierigkeit 69
konfliktfreier Bewegungsraum 66, 67
- Bewertungsfunktion 58
- Bewertungskriterium 60
- Bildschirmkarte 5, 27
- Bildschirmkoordinaten 31
- Blattkoordinaten 31
- Boundingbox 34
- C**
- calculateCutScale₁* 84
- calculateCutScale₂* 88
- calculateCutScale₃* 90
- Cluster 76
- conflictFree* 81
- cutScale* 88
- D**
- Datenstrukturen 13
- deselect* 41, 44, 96, 101, 104
- Deselektionskriterium 33, 43, 58, 69, 96
- Deselektionsmastab 33, 44, 96, 109
Konfliktgraphkante **41**
Konfliktgraphknoten **42**
linienhaft 79
punkthaft 42
punkthaft-linienhaft 91
- E**
- Echtzeit 27
- Einbettung
geometrisch 49
- F**
- Feinplatzierung 116, 132
- freeArea* 121

freie Beschriftungsfläche 52, 69, 71, 79,
109, 143, 146

G

Gebiet **61**, 71

Generalisierung 104

Geometriotypwechsel 104

Globalisierungseffekt 72, 76

Gradient Descent Algorithmus 23

Graph 13

Greedy Algorithmus 23

H

Heuristik 13, 21, 70, 116

I

Information

geometrisch 48

Informationsverlust 105, 134

Interaktionsphase 27, 101, 111

Intervall-Baum 14

inzident 13

K

Kartenausschnitt 111, 112, 137

Kartenbeschriftung 5

Karteninteraktion 27

Klassifikation 145

Knotengrad 43, 48, 70, 109, 143, 146

Kombination 137

Konfliktbereich 80

Konflikte

geometrische Lage 70

Konfliktfläche 71

Konfliktgebirge **48**

Höhe 49

Konfliktgraph 27, 29, **32**, 72, 92, 116

Ableitung 112

eingebettet 48

Erzeugung 96

Generierung 143

Kante **32**, 44

Knoten **32**

maßstabsunabhängig 33

reaktiver 29, **33**, 96, 103, 109, 111, 112

Speicherung 101

statischer 29, **33**, 111, 112, 137

vereinigt 138

Konfliktmaßstab 44

Konfliktrechteck 52, **54**

Konfliktsektor **49**, 71, 109

L

Labelbox 6

labeled 116

labelScore 58, 67, 82, 99, 147

Linienglättung 104

lokale Minima 24

lowerBound 33, 41, 97, 100, 104

M

mapConflicts 107

Maßstab **103**

Maßstabsintervall 103, 109

Maßstabsunabhängigkeit 29, 33, 41, 103

Medial Axis 94

N

Neubeschriftung 111

nonLabelObject 107

NP-vollständig 13

O

Objekt 5

flächenhaft 92

linienhaft 7, **74**

punkthaft 6

O-Notation 12

P

Phasen 27

Präprocessingphase 27, 105, 111, 112, 136

Priorität 43, 44, 97

Puffer 74, 84

Q

Quadrant **53**, 55, 62, 71

R

R-Baum 16, 102, 112, 146

Realweltkoordinaten 31

Realweltobjekte 31

Referenzbeschriftung 32

Referenzmaßstab 32

Referenzsignatur 32

S

Schlinge 13

Schlüssel 101, 112

Schnittmaßstab **39**, 44, 98, 146

linienhaft 78

punkthaft 39

punkthaft-linienhaft 83

Schnittpunkt 88

Schriftobjekt 43

Schriftüberlappung 109, 116

Scrollen 2, 27, 111

Segment-Baum 13

Segmentierung 72, **76**

Sektor **49**, 83

Selektionskriterium 69

Siedlungsfläche 95

Signatur 5, 104

Signaturierung 145

Simulated Annealing 24

Skalierungsfunktion 9

Skelett 94

Standlinie 94

stripe 81

System

komponentenbasiert 141

T

Textplatzierungsverfahren 21

Tiefensuche 127

topology 108, 132

Trennlinie 29, 107, 110, 132

tryItLater 117, 127

U

unlabeled 116

untreated 116

upperBound 33, 44, 96, 99, 104

V

value **58**, 66

Vektor 63

x-, y-Vektor 53, **54**

Vektorraum

Vereinigung 55, **56**

Verdeckung 105, 136

Verdrängung 104

Visualisierung 141, 143, 145

Vorselektion 116

W

Winkel

größter freier **49**, 52, 71, 109

Worst-Case 12

Z

Zoomen 2, 27, 111