Bottom-up Object Segmentation for Visual Recognition

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

vorgelegt von
João Luís da Silva Carreira
aus
Coimbra, Portugal

Bonn 2012

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

1. Gutachter:	Prof. Dr. Cristian Sminchisescu
2. Gutachter:	Prof. Dr. Reinhard Klein
Tag der Promotion:	19.12.2012
Erscheinungsjahr: 2013	

Abstract

Automatic recognition and segmentation of objects in images is a central open problem in computer vision. Most previous approaches have pursued either sliding-window object detection or dense classification of overlapping local image patches.

Differently, the framework introduced in this thesis attempts to identify the spatial extent of objects prior to recognition, using bottom-up computational processes and mid-level selection cues. After a set of plausible object hypotheses is identified, a sequential recognition process is executed, based on continuous estimates of the spatial overlap between the image segment hypotheses and each putative class.

The object hypotheses are represented as figure-ground segmentations, and are extracted automatically, without prior knowledge of the properties of individual object classes, by solving a sequence of constrained parametric min-cut problems (CPMC) on a regular image grid. It is show that CPMC significantly outperforms the state of the art for low-level segmentation in the PASCAL VOC 2009 and 2010 datasets.

Results beyond the current state of the art for image classification, object detection and semantic segmentation are also demonstrated in a number of challenging datasets including Caltech-101, ETHZ-Shape as well as PASCAL VOC 2009-11. These results suggest that a greater emphasis on grouping and image organization may be valuable for making progress in high-level tasks such as object recognition and scene understanding.

Acknowledgements

Many people have contributed to make this thesis a reality, by providing me with their guidance, friendship, shelter, love, money, code and data.

I would like to start by thanking my advisor, Prof. Cristian Sminchisescu, for teaching me a great deal about science, research and computer vision. He has always been extremely generous with his time, ideas and resources. We worked together closely and this thesis would not have been possible without his help and his passion for tackling hard problems. I would also want to thank the members of my doctoral committee, Profs. Reinhard Klein, Andreas Weber and Martin Rumpf for their time, feedback and interest in my work. I must also acknowledge the important contributions of my other co-authors Dr. Fuxin Li and Dr. Adrian Ion. Their singular abilities, expertises and creativity were fundamental to this thesis and it was great fun to work and travel with both.

I truly enjoyed the time spent doing research during my four years in Germany. The remaining time was equally pleasant. For that I have to thank my friends in Bonn, starting of course with Catalin (who can always count on me), and also Orestis, Habiba, Stefan, Irene, Simplice, Dong, Martin and Galina, Branimir, Alessandro, Yadong, Qiuxia and Sascha Zhu. I also would like to acknowledge people in Coimbra from whom I learned much before leaving to Germany. I would like to thank in particular Prof. Paulo Peixoto, but also Profs. Jorge Batista, Joao Barreto and Hélder Araújo, as well as my ex-office mates Luiz Mirisola and Rui Caseiro.

During my thesis work, I was funded initially by a PhD scholarship from the Portuguese Science Foundation (FCT), reference SFRH/BD/24295/2005, and later by an early stage researcher contract under a Marie Curie Excellence Grant (MCEXT-025481) of the European Commission to Prof. Cristian Sminchisescu. I owe taxpayers a great deal and I hope to give back to them in some way in the future. My preferred way would be to help creating machines that can do the work that people dislike, or are unable to do themselves. If this turns out to be too difficult, I am confident I can contribute to at least build machines that can find nearby buses and cows.

I would like to dedicate this thesis to my family, in particular to my parents, brothers, Sandra and Afonso (who will solve computer vision).

Contents

Chapter 1.

Introduction

The goal of research in computer vision is to develop systems that can automatically construct representations of scenes from their images. The desired representation should be very rich. It should not only include spatial 3D information, but also 'labels' attached to things in the scene. These labels allow the perceiving system to relate the contents of the scene to things it has observed in the past and that got labeled in a similar way. For example one car instance could get labels such as 'car', 'Porsche car', 'my car'. The general process of labeling the elements of an image is called visual recognition. Replicating visual recognition in machines would lead to great economic and technological growth, as it would be a key enabler of robotics and would allow for dangerous and uninteresting activities to be automated.

Achieving successful visual recognition has, however, proven extremely hard. Consider objects: the recognition of each object present in a scene requires evaluating whether its appearance in the image is compatible with some object model. Since the specificities of the 3d scene are generally unknown, recognition requires a search over both available object models and over which image pixels belong to them. Both search spaces are gigantic: there are innumerous objects that we may want to recognize and there are innumerous subsets of image pixels which can a priori correspond to the projection of an object in the scene. This creates a need for efficient machinery to explore both search spaces.

The main problem studied in this thesis is how to efficiently explore the space of subsets of image pixels (segments) for recognition tasks. We pursue a bottom-up segmentation paradigm: the computation of a small number of segment proposals *precedes* the search over matching label models. We propose to sample a large set of segments using a new mechanism called Constrained Parametric Min-Cuts (CPMC), then rank the segments using mid-level regularities learned from annotated imagery and select the most promising. We show that this technique is extremely powerful and indeed outperforms the state-of-the-art on various benchmarks.

It is widely believed that there is often not enough evidence in an image for perfectly sampling all object regions. Our goal is more modest: to sample segments that cover objects accurately enough for successful recognition. To evaluate how well this goal is achieved we introduce a new recognition approach which labels and selects individual segments sequentially. Notably, multiple recognition tasks such as semantic segmentation, object detection and image classification, that are usually attacked using different techniques, can be solved in a unified way by our method. Besides their conceptual appeal, our techniques obtain results that are competitive or superior to those of the more specialized techniques.

For obtaining very precise segmentations robustly it may be necessary to develop additional processes. Top-down feedback could signal local errors in parts of otherwise good matches between object models and segments. Such signals can lead to a reinterpretation of low and mid-level cues in the image and to refined segments that better align with object

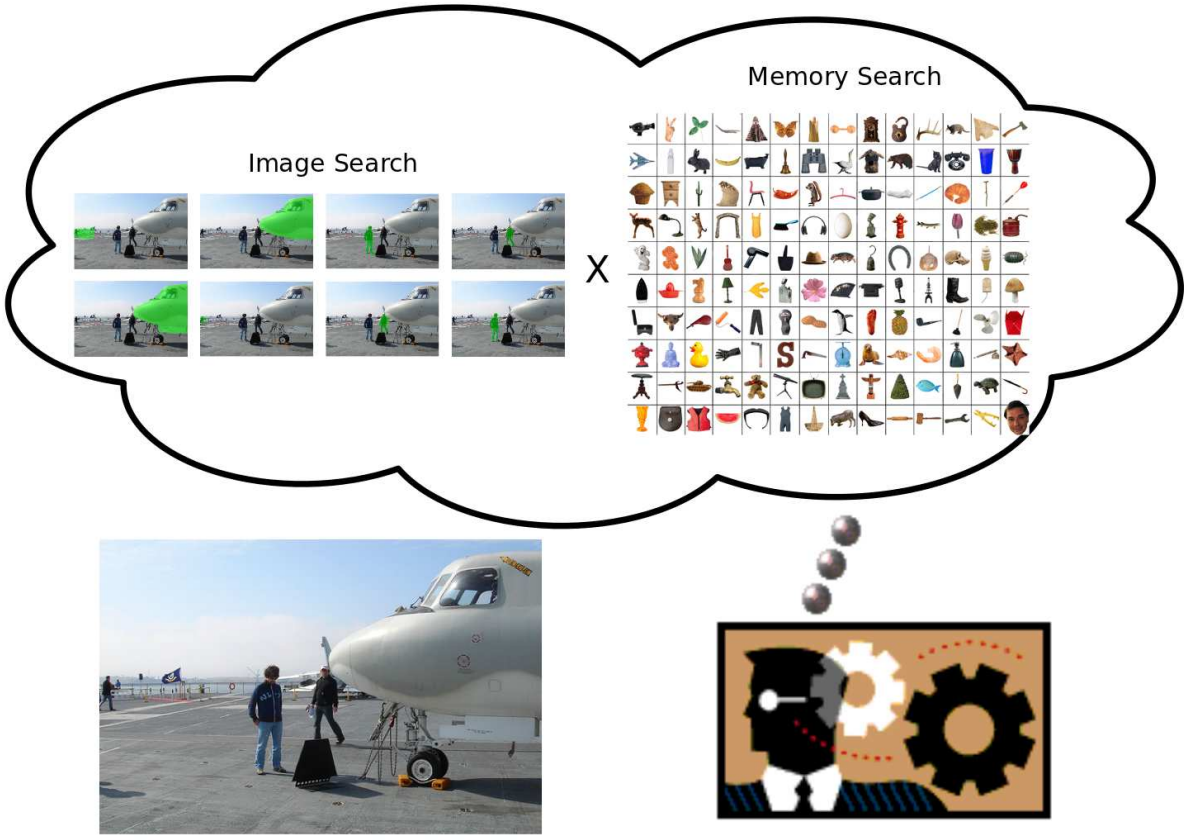


Figure 1.1.: Visual recognition involves searching for matches between patterns in the image and in memory. The precise location of these patterns in the image is unknown a priori and this is a major difficulty for recognition: the space of all possible closed boundaries in an image is immense. This thesis studies mechanisms for efficient exploration of the image, in the absence of prior knowledge about the scene. We propose a new method that moves the image search problem from the space of possible boundaries to the much reduced space of *plausible* boundaries. It does this by exploiting effectively low and mid-level regularities learned from ground-truth region annotations. This thesis also proposes a sequential recognition mechanism that employs such free-form regions.

models. The scope of this thesis is, however, confined to an investigation of the bottom-up stage.

The next section surveys the history of ideas in computer vision that relate segmentation and visual recognition. Then sec. 1.2 lists the main contributions of this thesis and sec. 1.3 details how the thesis is structured.

1.1. Historical Background

It is useful to place the ideas pursued in this thesis in their historical context. The fact that there is still much uncertainty about how the human brain achieves visual recognition, together with the limited performance that the best artificial systems achieve, contribute to keeping the debate about the role of segmentation in visual recognition, if any, alive up to this day. A central question is whether object segmentation can feasibly precede recognition, and,

if so, how much can be expected from it.

Some of the first proposals related to bottom-up processes for object segmentation were advanced by the Gestalt psychology school [157], which was founded in the beginning of the 20th century. They proposed simple bottom-up rules, called grouping principles, which permeated the perception of certain combinations of image elements as wholes. Features such as similarity and proximity were seen as favorable for grouping.

The work of the pioneers [121] of computer vision had a Gestalt flavor. They assumed bottom-up segmentation was feasible. Between the mid fifties and the end of the sixties, many of the vision systems were heavily customized to specific application domains applications such as optical character recognition and remote sensing [121]. Most shared a similar architecture: regions of interest in the image were identified, features were extracted on each and input to a classic pattern recognition technique such as nearest neighbor or a linear classifier. An example of a sophisticated technique developed at the end of this period, was the 'Edinburgh system' of Barrow and Popplestone [7]. Their system grew regions around locations on a regular grid and assumed that some of them would align with object parts. It searched then for the best matching between model parts and the set of computed regions.

Such approaches proved unsatisfactory, and a new movement was born that favored the use of task-specific knowledge to help segmentation [51, 138], or what is now known as top-down segmentation. Freuder [51] criticized Barrow and Popplestone's system, citing an example application he was studying: localizing hammers. With the 'Edinburgh system' the head and the handle of the hammer would have to be located bottom-up, independently. He argued: *'If the head were to be confused with the background in a scene, the match with the hammer model simply would not succeed. The presence of the handle did not direct the system in a search for the head.'*. He then asked a question that illustrates well top-down segmentation ideas, *'(...) must we simply accept and work with the results of the passive segments, or can other problems motivate a return to modify the results, or to consult with the primitive input data for these segments?'*.

The influential David Marr criticized object segmentation in general and suggested avoiding it altogether. He proposed to instead pursue bottom-up processes that produced descriptions of the surfaces in the scene - the 2 1/2D representation [105]. He believed surface properties could be retrieved independently of which semantic properties they held or to which objects they belonged to. Object segmentation would be available after 3D reconstruction and recognition. He criticized bottom-up object segmentation on two main grounds:

1. Not being clear as a problem.
2. The lack of enough information at local scales, to detect the regions.

Regarding the first problem, Marr famously wrote in his book: *'What, for example, is an object, and what makes it so special that it should be recoverable as a region in an image? Is a nose an object? Is a head one? Is it still one if it is attached to a body? What about a man on horseback?'* [105].

He also explained the second problem: *'People soon found the structure of images to be so complicated that it was usually quite impossible to recover the desired region by using only grouping criteria based on local similarity or other purely visual cues that act on the image intensities (...)'*. He stated that *'regions that have semantic importance do not always have any particular visual distinction.'* Marr also seemed to argue that top-down influences were necessary to achieve a segmentation: *'What was wrong with the idea of segmentation? The most obvious flaw seemed to be that 'objects' and 'desirable regions' were almost never visually primitive constructions and hence*

could not be recovered from the primal sketch or other similar early representations without additional specialized knowledge.' [105].

Research in bottom-up segmentation continued nevertheless, namely regarding its relationship with recognition. David Lowe developed the SCERPO system [100], which could identify a set of 3D objects, and determine their pose. SCERPO fitted and grouped sets of line segments, then ranked them by order of 'significance' and used the top-ranked sets for recognition. Although in this case the focus was not on regions but on sets of line segments, similar principles applied. Lowe believed in Gestalt ideas: *'(...) a major function of perceptual organization is to distinguish non-accidental groupings from the background of groupings that arise through accident of viewpoint or random positioning.'* [100]. He argued against Marr's case for bottom-up 3D preceding recognition, giving the example of line drawing interpretation. In line drawings there are often very few cues for Marr's methods (shape-from-shading, stereo) to exploit, and people are able to recognize objects in them [100], regardless.

Eric Grimson studied the combinatorics of several recognition approaches, and observed that successful bottom-up segmentation would greatly reduce the complexity of recognition [62]. The ideas pursued in this thesis are close to the views expressed in his book [62]: *'Can we determine subsets of the data likely to have come from a single object, using only characteristics of the data? The key word here is likely. We do not expect such methods to uniquely identify the best subset this would amount to solving the recognition problem. Rather, we want such methods to provide candidate data subsets, that are likely to have come from a single object. Our expectation is that we may have to search several of these subsets before finding a correct interpretation , but so long as the number of subsets to be searched remains small, we will still reduce the overall effort of the search.'*

Shimon Ullman's words, expressed in his book [145], reflect our ideas best. Considering the problems raised by Marr about the objective of segmentation not being clear and low-level cues being ambiguous, he frames segmentation in a well defined manner as *'a process that attempts to extract image structures that correspond to significant portions of stored object representations.'* He adds: *'If one object forms a part of another, the segmentation process should be capable of pulling out significant portions of either one.'* [145]. Similarly to Grimson he states that bottom-up segmentation should not be expected to obtain perfect object boundaries, but at least sufficiently good alignment with stored object representations.

The new millennium brought a new wave of sophisticated bottom-up segmentation techniques, including Mean Shift [30] and Normalized Cuts [131]. These methods can segment homogeneous regions accurately and energized research on recognition approaches that progressively combined segments, while guided by high-level knowledge about object shape [49, 70, 108]. This idea fell out of favor in the meanwhile, maybe because homogeneous image regions are neither very repeatable with respect to object semantics, nor very discriminative. Two other popular tendencies arose: class-specific top-down segmentation [15] and pixel labeling [65]. The first was a simplification of the problem explored in the seventies. Instead of using scene-specific knowledge to aid segmentation, it was now assumed that the object category and rough bounding box were known. Success was obtained in real images but restricted scenarios, such as side-views of left-facing horses [15], and such ideas have yet to be demonstrated in more unconstrained imagery [38, 39]. Pixel labeling approaches recognize locally at all locations in the image. They are very flexible and produce impressive results in datasets with restricted intra and inter-class variation [134, 132]. It is however doubtful that such approaches, unaided, will be effective at the desired scale: predicting hundreds of thousands of labels. There does not seem to be enough information in pre-specified neighborhoods

around a pixel to perform such fine discriminations reliably.

Despite its appeal, bottom-up segmentation has stubbornly resisted reliable solutions. For example in the nineties, there was excitement about recognition techniques that explicitly incorporated geometric invariance [166, 147]. This movement faded in part because the computation of geometric invariants required bottom-up grouping of a large number of features (*e.g.* five lines) [110].

Many approaches avoided segmentation altogether. One idea was to resort to minimal feature sets that could be explored exhaustively. These sets were used to index object models, for example in the *alignment* method [69]. One other approach that avoids segmentation and is still popular today is the sliding window [113]: rectangular regions are sampled exhaustively and in an image-independent way in both space and scale, so that each object is bounded tightly by at least one rectangle. The usage of the sliding window has become so widespread that localizing objects with it is now synonym with 'object detection' [41]. However, bounding boxes do not constitute a sufficiently accurate form of object localization for many tasks and exhaustive search sets tight constraints on the kind of processing that can be executed at each location, precluding many interesting but more burdensome ideas. There is now some evidence that progress in recognition using this paradigm has been slowing down, with minor variations of the same ideas dominating, and performance seems to have plateaued at around 40 % in the Pascal VOC challenge of 2011 [40]. It may be the right time to look again at alternatives, such as bottom-up segmentation.

1.2. List of Contributions

This thesis makes two principal contributions:

1. Effective mid-level region sampling. A main obstacle to bottom-up visual processing approaches has been the lack of effective algorithms for computing a small set of segment proposals that align well with the boundaries of structures and objects in an image. Previous techniques targeted the full-image segmentation problem, which imposes non-overlap consistency constraints between segments very early. We propose instead to generate segments independently from each other (they may overlap), using a novel algorithm named Constrained Parametric Min-Cuts (CPMC), which considerably improves upon the state-of-the-art on several challenging datasets. The algorithm consists of two steps. In the first step, efficient parametric max-flow techniques are used to sample a large pool of segments under alternative sets of putative constraints. The resulting segments are afterwards ranked using a learned function based on mid-level cues, which prefers regions that exhibit object-like regularities over those with implausible real-world statistics. The retained top-ranked segments provide a focused space of object hypotheses for recognition. This technique is the topic of chapter 2.

2. Ranking-based learning for segment selection and sequential recognition. Most popular detection and classification approaches rely on binary classifiers to select among a set of possible regions and their labels. These approaches encourage models that give high score to the ground-truth 'right answer' and low scores indiscriminately for all 'wrong answers'. Results that are 'partially correct' are ignored during learning. This is problematic when the possible outputs are regions obtained bottom-up, for two reasons. First, perfect ground-truth regions are unlikely to be obtained bottom-up, hence the ground-truth regions are not representative of the regions that are used during testing. Second, there may be multiple

Figure 1.2.: Illustration of issues involved in segment ranking. After bottom-up segmentation (here using CPMC), recognition can be posed as selection among multiple sampled segments and a set of labels. While there are usually multiple segments covering each object, segments that align perfectly with objects may not always, if ever, be sampled. What seems important is to select the segment that best covers each object. Secondly, the ranking is affected by occlusion. In the images above, the segment covering the upper body of the girl is undesirable since there is a better one covering her full body. In the other image the segment covering the upper-body of the man is the most desirable. These properties justify our ranking formulation to learning: segments are regressed on the predicted overlap they have with ground truth objects. This formulation encourages finer segment selection than standard learning approaches based on binary classification and handles better the part-whole issues. The segment covering the girl's upper-body is not a negative example in our formulation, it is a 'positive example' which is learned to be ranked proportionally lower than the segment covering the full body.

regions that only partially align with an object and we want that those that align better get higher scores. These issues are illustrated in fig. 1.2. We attack such problems using ranking techniques, modeled as regression on segment alignment measures. We show that such ranking techniques outperform learning approaches based on binary classification for segment selection and recognition tasks. We also show how to use the rankers to sequentially parse images in semantic segmentation tasks. Finally, when selecting segments in a mid-level, label-independent fashion, we show also that diversifying the ranking improves the pools of retained segments significantly. This contribution is presented in both chapters 2 and 3.

1.3. Thesis Outline

The first half of the thesis studies mid-level segmentation, the second studies visual recognition. We review specific prior work in the respective chapters.

Chapter 2 presents one of the main contributions of this thesis: the framework for mid-level region sampling which we call *Constrained Parametric Min-Cuts* (CPMC). The chapter begins

with the description of a constrained binary energy function over pixels. We then refer the different types of constraints applied and discuss how inference is efficiently performed using parametric max-flow. Afterwards, we introduce a set of features inspired by Gestalt principles [157] and an approach to learn object regularities from these features, using a ranking formulation. The learned ranking function allows many redundant elements to be discarded and maximum marginal relevance diversification improves performance further. The chapter concludes with results on extensive experiments performed on various datasets and with multiple different performance benchmarks.

After our mid-level segmentation machinery has been introduced the thesis shifts to the topic of recognition – how to select the most appropriate conjunction of regions and their labels given an image. Chapter 3 introduces our sequential ranking approach to recognition. We revisit the ranking problem from chapter 2 and discuss large-scale least-squares methodologies suitable for learning to rank regions in large-scale problems with millions of training examples. We then propose a segment similarity function that penalizes undersegmentations more than the standard similarity function. Afterwards we explain the combination of region-based recognition features we employ for recognition, together with the machinery for support vector regression with non-linear kernels. Finally, we discuss a procedure that sequentially combines high-scoring labeled segments and forms a robust labeling of the image. We present results on image classification, object detection and semantic segmentation, as well as empirical studies on the best number of sampled segments to retain for recognition and the advantage of using regression over classification for segment selection and recognition.

Conclusions and suggested future directions are discussed in chapter 4. The technical material in the thesis is supported by appendix A, a tutorial about parametric max-flow, the combinatorial optimization technique used in the proposed CPMC algorithm. We review the combinatorial algorithm, the family of energy functions it minimizes and its background in computer vision.

1.3.1. Publications

The main material in this thesis has been published in journals and conference proceedings. We now list the relevant publications.

Chapter 2:

- *Constrained Parametric Min-Cuts for Automatic Object Segmentation*, João Carreira and Cristian Sminchisescu, in proceedings of IEEE Conference on Computer Vision and Pattern Recognition 2010.

- *CPMC:Automatic Object Segmentation Using Constrained Parametric Min-Cuts*, João Carreira and Cristian Sminchisescu, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012.

Chapter 3:

- *Object Recognition as Ranking Holistic Figure-Ground Hypotheses*, by Fuxin Li, João Carreira and Cristian Sminchisescu (the first two authors contributed equally), in proceedings of IEEE Conference on Computer Vision and Pattern Recognition 2010.

- *Object Recognition by Sequential Figure-Ground Ranking*, by João Carreira, Fuxin Li and Cristian Sminchisescu (the first two authors contributed equally), International Journal of Computer Vision, 2012.

The following co-authored papers are also closely related to this thesis, but not included:

- *Image Segmentation by Discounted Cumulative Ranking on Maximal Cliques*, by João Carreira, Adrian Ion and Cristian Sminchisescu, in technical report 06-2010 (arXiv:1009.4823).
- *Image Segmentation by Figure-Ground Composition into Maximal Cliques*, by Adrian Ion, João Carreira and Cristian Sminchisescu, in proceedings of International Conference on Computer Vision 2011.
- *Probabilistic Joint Image Segmentation and Labeling*, by Adrian Ion, João Carreira and Cristian Sminchisescu, Advances in Neural Information Processing Systems 2011.

Chapter 2.

Constrained Parametric Min-Cuts for Automatic Object Segmentation

Abstract

We present a novel framework to generate and rank plausible hypotheses for the spatial extent of objects in images using bottom-up computational processes and mid-level selection cues. The object hypotheses are represented as figure-ground segmentations, and are extracted automatically, without prior knowledge of the properties of individual object classes, by solving a sequence of constrained parametric min-cut problems (CPMC) on a regular image grid. In a subsequent step, we learn to rank the corresponding segments by training a continuous model to predict how likely they are to exhibit real world regularities (expressed as putative overlap with ground truth) based on their mid-level region properties, then diversify the estimated overlap score using maximum marginal relevance measures. We show that this algorithm significantly outperforms the state of the art for low-level segmentation in the VOC 2009 and 2010 datasets.

This chapter corresponds to paper *CPMC: Automatic Object Segmentation Using Constrained Parametric Min-Cuts*, João Carreira and Cristian Sminchisescu, PAMI 2012, which is an extension of *Constrained Parametric Min-Cuts for Automatic Object Segmentation*, João Carreira and Cristian Sminchisescu, presented at CVPR 2010.

2.1. Introduction

Reliably identifying the spatial extent of objects in images is important for high-level vision tasks like object recognition. A region that covers an object fully provides a characteristic spatial scale for feature extraction, isolates the object from potentially confusing background signal and allows for information to be propagated from parts of the object to the whole (a region covering a human fully makes it possible to propagate the person identity from the easier to identify face area to the rest of the body).

Given an image, the space of all possible regions, or segments that can be obtained, is exponentially large. However, in our perceived visual world not all image regions are equally likely to arise from the projection of a three-dimensional object. Objects are usually compact and this results in their projection in the image being connected; it is also common for strong contrast edges to mark objects boundaries. Such properties reduce the number of plausible object regions greatly, but may not be sufficient to unambiguously identify the optimal spatial support for each of the objects in an image.

In this chapter, we follow a two step strategy by combining a figure-ground, multiple hypothesis bottom-up approach to segmentation with subsequent verification and ranking based on mid-level region properties. Key to an effective solution is the capability to leverage

the statistics of real-world objects in the selection process. One possibility would be to learn the parameters of the segmentation algorithm directly, by training a machine learning model using large amounts of human annotated data. However, the local scope of dependencies and the intrinsically combinatorial nature of image segmentation diminishes the effectiveness of learning in such 'pixel spaces' as many interesting features such as the convexity and the smoothness of a region boundary are difficult to capture locally. On the other hand, once sufficient image support is available, learning to distinguish 'good' segments that represent plausible projections of real-world surfaces, from accidental image partitions becomes in principle feasible. This motivates our novel decomposition of the problem into two stages. In the first stage, we explore the space of regions that can inferred from local measurements, using cues such as good alignment with image edges. The process of enumerating regions with plausible alignment with the image contours is performed using exact combinatorial methods based on parametric max-flow. Then, in the restricted space of generated regions, we use a learned combination of advanced mid-level features to induce a more accurate global ranking of those regions in terms of their probability to exhibit 'object-like' regularities.

A key question, and one of our contributions, is how should image partitions be generated. Should region hypotheses be allowed to overlap with each other? Should one aim at multi-region image segmentations early? We argue that segmentation is already a sufficiently challenging problem without such constraints. It may be better to enforce global inter-region spatial consistency at a later stage of processing, by higher-level routines with more precise spatial scope for this calculation. We argue that attempts to enforce complex multi-region consistency constraints early may disallow the speculative behavior necessary for sampling regions effectively, given the inherently ambiguous nature of the low-level cues one typically operates on initially. Hence, differently from most of the existing approaches to segmentation, we derive methods to generate *several independent figure-ground partitions*, rather than a battery of splits of each image into multiple, non-overlapping regions.

Our proposed framework is depicted in fig. 2.1. We first solve a large number of independent binary min-cut problems on an image grid, at multiple scales. These are designed as energy functions efficiently solvable with parametric min-cut/max-flow techniques. The resulting pool of segments is minimally filtered to remove trivial solutions and ranked using a regressor trained to predict to what extent the segments exhibit the regularities typical of real-world objects, based on their low and mid-level region properties. Because ranking tends to place redundant variations of a same segment in similar ranks, we diversify the resulting segment ranking using Maximal Marginal Relevance measures, with the top ranked segments retained.

The quality of the list of object hypotheses returned by our algorithm is evaluated empirically by measuring how accurate they are with respect to pixel-level ground truth human annotations, in object recognition datasets. We also record performance as a function of the number of segments. Results are reported on several publicly available benchmarks: MSRC [133], the Weizmann Segmentation Database [124] and both VOC2009 and VOC2010 [38, 39] where the proposed method is shown to significantly outperform the state of the art, while at the same time using significantly fewer segments.

Several visual analysis methods may benefit from outputs like the ones provided by our algorithm. Object detectors usually scan a large number of bounding boxes in sliding window schemes [46, 149] without considering the plausibility of pixel grouping within each. Semantic segmentation algorithms [58, 159, 86, 56] incorporate the outputs of these object de-

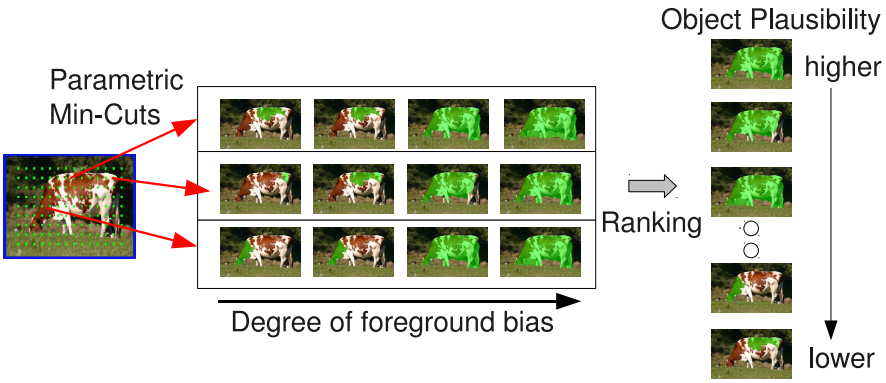


Figure 2.1.: Our object segmentation framework. Segments are extracted around regularly placed foreground seeds, with various background seeds corresponding to image boundary edges, for all levels of foreground bias, which has the effect of producing segments at different locations and spatial scales. The resulting set of segments is ranked according to their plausibility of being good object hypotheses, based on mid-level properties. Ranking involves first removing duplicates, then diversifying the segment overlap scores using maximum marginal relevance measures.

tectors, and may need to mediate the transition between the rectangular regions produced by the detector and the desired free-form regions that align with object boundaries. Unsupervised object discovery [123] also requires good class-independent object proposals. While the presentation focuses on the problem of object segmentation, the proposed method is general and can rank lists of segments that exhibit the statistics of non-object, 'stuff' regions such as grass or sky, as long as appropriate ground truth training data is provided.

An implementation of the proposed algorithm is made publicly available via our website [25].

Chapter Organization: Section §2.2 reviews the related literature, §2.3 introduces the methodology used to generate an initial pool of segments for an image and §2.4 presents the segment ranking procedure. Section §2.5 presents experimental results and shows comparisons with the state of the art. An extension of the basic algorithm to include bounding box constraints and the corresponding results are described in §2.5.3. We conclude and discuss ideas for future work in §2.6.

2.2. Related Work

One of the first image segmentation approaches, published more than 40 years ago by Muerle and Allen [109], aimed to compute 'object' regions. Small patches having similar gray-level statistics were iteratively merged, starting at a seed patch. Region growing stopped when none of the neighboring candidate patches was sufficiently similar to the current region. The process was repeated until all pixels were assigned. This method took advantage of the fundamental grouping heuristic that neighboring pixels with different color are more likely to belong to different objects. However it produced very local solutions and was not able to deal with textured regions, and even less, take advantage of more sophisticated object statistics. Later, more accurate techniques emerged—good surveys can be found in [64, 164, 3]. However, most methods still pursued a single optimal segmentation of an image into a set

of non-overlapping regions that covered it fully (a multi-region image partitioning). But a sufficiently good partitioning is not easy to obtain given the ambiguity of low and mid-level cues. Moreover, there were no quantitative benchmarks to gauge progress and most papers only described the merits of the output segmentations qualitatively, usually based on results obtained on a few images.

As a consequence, in the nineties, part of the recognition community lost confidence that a reliable segmentation procedure would be found and began investigating solutions that avoided bottom-up segmentation entirely [110]. This trend led to the current prevalence of bounding box detectors operating on sliding windows [46, 154]. These detectors rely on a dense evaluation of classifiers in overlapping rectangular image regions, with consistency usually enforced a posteriori by non-maxima suppression operations. Sliding window methods are effective in localizing certain objects like faces or motorbikes, but do not obviously generalize to more complex structures and cannot be easily adapted for general 3d scene understanding: *e.g*. information predicted on rectangular image regions is not sufficient for tasks such as vision-based manipulation of a cup by a robot, where it is critical to precisely identify the cup handle in order to grasp it.

Such considerations made a revival of segmentation inevitable. The trend has gained momentum during the past ten years, propelled by the creation of annotated benchmarks [106, 38] and new segmentation performance metrics [38, 146]. A second important factor was the adoption of machine learning techniques to optimize performance on benchmarks. A third factor was relaxing the constraint of working with a single partitioning. A popular approach emerged by computing several independent segmentations, possibly using different algorithms. This idea was pursued by Hoiem *et al*. [68] for geometric labeling problems. Russel *et al*. [123] computed normalized cuts for different number of segments and image sizes in the context of unsupervised object discovery. By generating tens to hundreds of thousands of segments per image, Malisiewicz and Efros [102] produced very good quality regions for the MSRC dataset, by merging pairs and triplets of segments obtained using the Mean Shift [30], Normalized Cuts [131] and Felzenszwalb-Huttenlocher's (FH) [45] algorithms. Stein *et al*. [137] solved Normalized Cut problems for different number of segments, on a special affinity matrix derived from soft binary mattes, whereas Rabinovich *et al*. [117] shortlisted segmentations that reoccured, hence were potentially more stable.

The computation of multiple segmentations can also be organized hierarchically. Shi and Malik [131] recursively solve relaxations of a Normalized Cut cost based on graphs constructed over pixel nodes. Sharon *et al*. [130] proposed algebraic multigrid techniques to efficiently solve normalized cuts problems at multiple levels of granularity, where graphs with increasingly more complex features were used at coarser levels. Arbeláez *et al*. [4] derive a segment hierarchy by iteratively merging superpixels produced by an oriented watershed transform. They use the output of the learned globalPb boundary detector [101] and can represent the full hierarchy elegantly by a single ultrametric contour map. The hierarchy is a natural representation for segmentation, as it lends itself to compositional representations. However, inaccuracies in one level (due to incorrect merging of two regions from the previous level, for example), tend to propagate to all coarser levels. Therefore, given the same segmentation technique, generating a single hierarchy is likely to be less robust than using independent segmentations.

Differently, our region sampling methodology generates multiple independent binary hierarchies constrained at different positions in the image. Each level of the hierarchy corre-

sponds to a partitioning into figure and ground, where only the figure region is retained, and regions at finer levels are nested inside coarser levels regions (this is a property induced by our parametric max-flow methodology [53]). In this way, we aim to better sample the space of plausible regions popping up at different image locations. We compute these partitionings using energies mostly related to the ones developed for interactive segmentation applications, where, however, computing a single figure-ground solution is typical. In these applications, max-flow algorithms are quite popular because they can obtain exact optima for certain energy minimization problems that involve region and boundary properties [18]. Generally the user assigns some pixels to the foreground and background regions manually and these constrain an energy function, which is optimized using a global minimization algorithm. The two steps are repeated until the set of manually assigned pixels constrain the solution sufficiently to make the resulting binary segmentation satisfactory. Variants requiring less manual interaction have been developed, such as GrabCut [122], where a simple rectangular seed around the object of interest is manually initialized and an observation model is iteratively fitted by expectation maximization (EM). Alternatively, Bagon *et al*. [6] require a user to simply click a point inside the object of interest, and use EM to estimate a sophisticated self-similarity energy.

Max-flow techniques can only globally optimize energies defined on local features such as contrast along the boundary and good pixel fit to a color or texture model. Interesting relaxation approaches exist for some energies whose minimization is NP-hard, such as curvature regularity of the boundary [127] and approximations have been developed for energies with connectivity priors [151]. However, many other more global properties, such as convexity or symmetry, are significantly more challenging to optimize directly. This motivates our segment generation and ranking procedure. We differ from existing methods not only in leveraging an efficient parametric max-flow methodology to solve for multiple breakpoints of the cost, thus exploring a much larger space of plausible segment hypotheses in polynomial time, but also in using regression methods on generic mid-level features, in conjunction with ranking diversification techniques, to score the generated segments. This fully automates the process of distilling a representative, yet compact segment pool. No manual interaction is necessary in our method.

One of the big challenges in segmentation is to leverage the statistics of real world images in order to obtain more coherent spatial results. Methods that learn low-level statistics have been applied to distinguish real from apparent contours [50, 36, 77] and similar from dissimilar superpixels [68]. Ren and Malik [120] use a random search algorithm to iteratively hypothesize segmentations by combining different superpixels, and use a classifier to distinguish good segmentations from bad ones. Pen and Veksler [114] learn to select the best segment among a small set generated by varying the value of one parameter, in the context of interactive segmentation. Models based on mid-level properties have also been learned to distinguish good from bad regions [120]. High-level shape statistics can be incorporated into binary segmentation models, usually as non-parametric distributions of templates [92, 32, 129]. Expressive part-based appearance models have also been developed [14, 91, 93, 83]. As objects in real images exhibit large variability in pose, have high intra-class variation and are often occluded, it is likely that such methods may require bottom-up initialization, which an algorithm like ours can provide. Effectively leveraging high-level shape priors in the initial steps of a visual processing pipeline may not always be feasible.

Our method aims to learn what distinguishes meaningful regions, covering full objects,

from accidental pixel groupings. Since our original presentation at VOC2009 [28] and publication [26], related ideas have been pursued. Endres and Hoiem [37] follow a processing pipeline related to ours, but employ a learned affinity measure between superpixels, rather than pixels, and a structured learning approach on a maximum marginal relevance measure similar to the one we originally proposed to diversify ranking. To generate figure-ground segments, Levinshtein *et al*. [94] developed a procedure based on parametric max-flow principles similar to ours, but use a graph where new similarity measures are constructed on superpixels. In parallel work, Alexe *et al*. [1] learn a naive Bayes model to distinguish bounding boxes enclosing objects from those containing amorphous background, without knowledge of the shape and appearance of particular object classes. They also show how to sample bounding boxes from the model efficiently but do not provide segmentations. Salient object detection [98] approaches are also relevant to our work, but they focus on selection criteria inspired by attention mechanisms. We are instead interested in computing regions that cover every object in an image well, independently of whether they 'pop out' from the rest of the scene or not.

2.3. Constrained Parametric Min-Cuts (CPMC)

In order to generate a pool of segments with high probability of not missing regions with good object overlap, multiple constrained parametric min-cut (CPMC) problems are solved with different seeds and unary terms. This leads to a large and diverse pool of segments at multiple spatial scales. The segments corresponding to implausible solutions are subsequently discarded using simple ratio cut criteria. The remaining are clustered so that all but representative segments with low energy are retained, among those extremely similar. The final working set of segments is significantly reduced, but at the same time the most accurate segments are preserved.

2.3.1. Setting up the Energy Functions

For each image, alternative sets of pixels, called seeds, are hypothesized to belong to the foreground and the background. The foreground seeds are placed on a grid, whereas background seeds are associated with sets of pixels along the image border. For each combination of foreground and background seeds we compute figure-ground segmentations with multiple levels of foreground bias. The levels of bias are induced by varying the cost of assigning non-seed pixels to the foreground. Inference consists of finding minimum cuts for the different values of foreground bias — in fact searching over multiple foreground biases is intrinsic to our parametric max flow procedure. The optimization problem is formulated next.

Let $I(\mathcal{V}) \rightarrow R^3$ be an image defined on a set of pixels $\mathcal{V}$. As commonly done in graph-based segmentation algorithms, the similarity between neighboring pixels is encoded as edges of a weighted graph $G = (\mathcal{V}, \mathcal{E})$. Here, each pixel is a node in the set $\mathcal{V}$. The foreground and background partitions are represented by labels 1 and 0, respectively. Seed pixels $\mathcal{V}_f$ are constrained to the foreground and $\mathcal{V}_b$ to the background by setting infinity energy to any labeling where they receive the contrasting label. Our overall objective is to minimize an energy function over pixel labels $\{x_1, ..., x_N\}, x_i \in \{0, 1\}$, with N the total number of pixels. In particular, we optimize the following energy function:

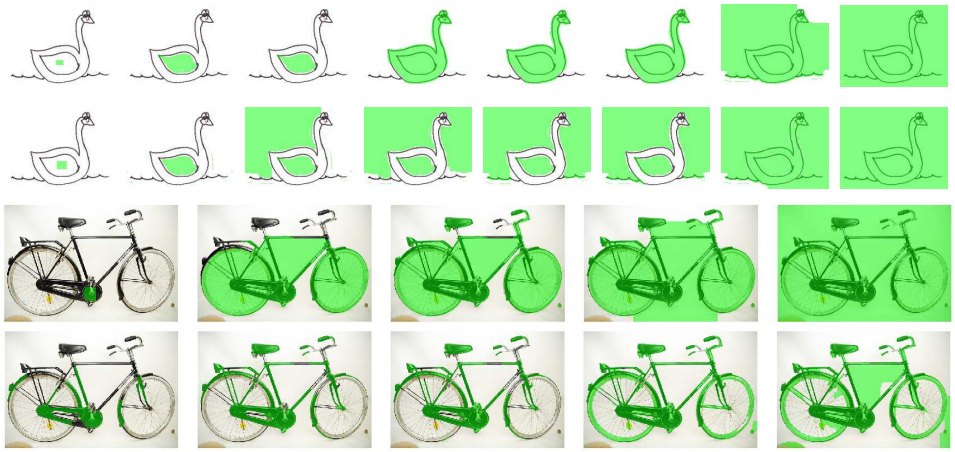


Figure 2.2.: Different effects of uniform and color-based unary terms. For illustration, a single foreground seed was placed manually at the same location for two energy problems, one with uniform and another with color unary terms. Shown are samples from the set of successive energy breakpoints (increasing λ values) from left to right, as computed by parametric max-flow. Uniform unary terms are used in rows 1 and 3. Color unary terms are used in even rows. Uniform unary terms are most effective in images where the background and foreground have similar color. Color unary terms are more appropriate for objects with elongated shapes.

$$E^{\lambda}(X) = \sum_{u \in \mathcal{V}} D_{\lambda}(x_u) + \sum_{(u,v) \in \mathcal{E}} V_{uv}(x_u, x_v) \tag{2.1}$$

with $\lambda \in \mathbb{R}$, and unary potentials given by:

$$D_{\lambda}(x_u) = \begin{cases} 0 & \text{if } x_u = 1,\ u \notin \mathcal{V}_b \\ \infty & \text{if } x_u = 1,\ u \in \mathcal{V}_b \\ \infty & \text{if } x_u = 0,\ u \in \mathcal{V}_f \\ f(x_u) + \lambda & \text{if } x_u = 0,\ u \notin \mathcal{V}_f \end{cases} \tag{2.2}$$

The foreground bias is implemented as a cost incurred by the assignment of non-seed pixels to background, and consists of a pixel-dependent value $f(x_u)$ and an uniform offset λ. Two different functions $f(x_u)$ are used in practice. The first is constant and equal to 0, resulting in a uniform (variable) foreground bias. The second function uses color. Specifically, RGB color distributions $p_f(x_u)$ on seed $\mathcal{V}_f$ and $p_b(x_u)$ on seed $\mathcal{V}_b$ are estimated to derive $f(x_u) = \ln p_f(x_u) - \ln p_b(x_u)$. The probability distribution of pixel j belonging to foreground is defined as $p_f(i) = \exp[-\gamma \cdot \min_j(||I(i) - I(j)||)]$, with γ a scaling factor, and j indexing representative pixels in the seed region, selected as centers resulting from a k-means algorithm (k is set to 5 in all of our experiments). The background probability is defined similarly. This choice of function is motivated by efficiency, being much faster to estimate compared to the frequently used Gaussian mixture model [122]. Color-based unary terms are more effective when the color of the object is distinctive with respect to the background, as well as when objects have thin parts. Uniform unary terms are more useful in the opposite case. The complementary effects of these two types of unary energy terms are illustrated in fig. 2.2.

The pairwise term V_{uv} penalizes the assignment of different labels to similar neighboring

pixels:

$$V_{uv}(x_u, x_v) = \begin{cases} 0 & \text{if } x_u = x_v \\ g(u,v) & \text{if } x_u \neq x_v \end{cases} \tag{2.3}$$

with similarity between adjacent pixels given by $g(u,v) = \exp\left[-\frac{\max(gPb(u),gPb(v))}{\sigma^2}\right]$. gPb returns the output of the multi-cue contour detector globalPb [101] at a pixel. The square distance is also an option we experimented with, instead of the $\max$ operation, with similar results. The *boundary sharpness* parameter σ controls the smoothness of the pairwise term.

The function defined by eq. 2.1 is submodular. Given a pair of foreground and background seeds and $f(x_u)$, the cost can be minimized exactly for all values of λ in the same complexity as a single max-flow problem, using a parametric solver [80]. In canonical form, parametric max-flow problems differ from their max-flow counterparts in that capacities from the source node are allowed to be linear functions of a parameter, here λ. As λ (effectively our foreground bias) varies there are at most $(N-1)$ different cuts in the transformed graph, where N is the number of nodes, although for the graphs encountered in vision problems there are generally far fewer (see our study in §2.3.3). The values of λ for which the cut values change are usually known as *breakpoints*. When the linear capacity functions from the source are either non-increasing or non-decreasing functions of λ, the problem is said to be monotonic. Our energy problems are monotonic because, for all unary terms, λ is multiplied by the same factor, 1. This important property implies that all cuts computed for a particular choice of source and sink seeds are nested.

In this work we use the *highest label pseudoflow* solver [67], which has complexity $O(mN\log(N))$ for image graphs with N nodes and m edges. The complexity of the CPMC procedure is thus $O(kmN\log(N))$, as we solve multiple parametric max-flow problems, for each of the k combinations of foreground and background seeds, and for different choices of $f(x_u)$. The pseudoflow implementation we used requires a set of λ parameters for which to compute cuts. For the study in §2.3.3, we additionally use an implementation based on Gallo *et al.* [53] in order to analyze the segmentation results produced by a push-relabel parametric max-flow solver which retrieves all breakpoints [5].

The graph construction that maps to the energy functions in (2.1), for each choice of foreground and background seed, augments the original problem dependency graph G with two special nodes, source s and sink t that must be in separate partitions in any binary cut [18]. The unary energy terms are encoded as edges between these special nodes and the nodes in $\mathcal{V}$.

2.3.2. Effect of Grid Geometry

As *foreground seeds*, we chose groups of pixels that form small solid squares. We have experimented with three different strategies to place them automatically: rectangular grid geometry, centroids of superpixels obtained with normalized cuts, and centroids of variable size regions, closest to each rectangular grid position, obtained using segments obtained by the algorithm of [45]. As shown in table 2.1, the performance differences are not very significant (see section §2.5 for details about the datasets and the evaluation criteria).

The *background seeds* are necessary in order to prevent trivial cuts that leave the background set empty. We used four different types: seeds including pixels that cover the full image boundary, just the vertical edges, just the horizontal edges and all but the bottom image edge.

This selection strategy allows us to extract objects that are only partially visible, due to clipping at different image boundaries.

In practice we solve around 180 instances of problem (2.1) for each image, for 30 λ values each (during processing, we skip duplicate breakpoints), defined on a logarithmic scale. The set of figure-ground segmentations is further enlarged by splitting the ones with multiple connected foreground components. The final pool has up to 10,000 segments per image.

As an alternative to multiple 'hard' background seeds, it is possible to use a single 'soft' background seed. This can be a frame one pixel wide covering the border of the image, with each pixel having a finite penalty associated to its assignment to the foreground. This construction is more efficient, as it decreases the number of energy problems to solve by 75%. We used this type of background seeds in an extension of the basic algorithm, presented in section §2.5.3.

Seed placement	MSRC score	Weizmann score
Grid	0.85 ± 0.1	0.93 ± 0.06
NCuts	0.86 ± 0.09	0.93 ± 0.07
FH	0.87 ± 0.08	0.93 ± 0.07

Table 2.1.: Effect of spatial seed distribution. The use of superpixel segmentation algorithms (*e.g.* Normalized Cuts or FH [45]) to spatially distribute the foreground seeds does not significantly improve the average covering score on the MSRC dataset, over regular seed geometries. On Weizmann, the average best F-measure is the same for all distributions, perhaps because the objects are large and any placement strategy eventually distributes some seeds inside the object.

2.3.3. Effect of λ Schedule

The effect of solving problem (2.1) for all λ values, instead of a preset logarithmic λ schedule, was evaluated on the training set of the PASCAL VOC 2010 segmentation dataset (the typical distinction into training and testing is not relevant for the purpose of this experiment, where the goal is only to analyze the number of breakpoints obtained using different search strategies). We use a 6x6 regular grid of square seeds and solve using two procedures: (1) 20 values of λ sampled on a logarithmic scale (only the distinct energy optima are recorded) and, (2) all λ values, as computed as breakpoints of (2.1). We have recorded the average computational time per seed, the ground truth covering score, and the number of breakpoints obtained under the two λ-search strategies. The results are shown in table 2.2, suggesting that a preset λ schedule is a sensible option. Using only 20 values produces almost the same covering as the one obtained using all values, it is 4 times faster and generates 10% of the total number of breakpoints, hence fewer segments. We also plot the distribution of the number of breakpoints per seed in fig. 2.3, under the same experimental conditions. The frequency of breakpoints has a dominantly unimodal (bell) shape, with mean 110, but a slightly heavier tail towards larger numbers of segments. There are never less than 15 breakpoints in this dataset.

2.3.4. Fast Segment Rejection

Generating a large set of segments increases the hit rate of the algorithm, but many segments are redundant or do not obey the statistics of real-world surfaces imaged by a camera. For

# λ values	# breakpoints	Time (s)	Covering
20	12.3	1.8	0.713
all	114.6	7.5	0.720

# objects	1-2	3-4	5-6	7-13
# breakpoints all λ	112.19	124.60	125.29	142.83
# breakpoints 20 λ	12.27	12.64	13.08	13.45
# images	717	147	68	32

Table 2.2.: Covering results obtained on the training set of VOC2010, based on a 6x6 grid of uniform seeds. The table compares the results of solving CPMC problems for 20 values of λ, sampled on a logarithmic scale, with the results obtained by solving for all possible values of λ. Shown are the average number of breakpoints per seed, and the average time required to compute the solutions for each seed. Computing all breakpoints for each seed provides modest ground truth covering improvements, at the cost of generating a larger number of segments and an increased computation time. The second table shows that images containing a larger number of ground truth objects tend to generate more breakpoints per seed.

images with large homogeneous regions, the original hypothesis generation step can also produce many copies of the same segment because of the seeding strategy — every seed placed inside the region would tend to generate the same segment for the same λ. Moreover, sometimes visually arbitrary segments are created, as artifacts of the foreground bias strength and the seed constraints employed.

We deal with these problems using a fast rejection step. We first filter very small segments (up to 150 pixels in our implementation), then sort the segments using a simple criterion (we have used the ratio cut [156] as this is scale invariant and very selective) and retain up to 2,000 of the highest scoring segments. Then we hierarchically cluster the segments using overlap as a similarity measure, to form groups with all segments of at least 0.95 spatial overlap. For each cluster, we retain the segment with the lowest energy.

The number of segments that pass the fast rejection step is usually small, being indicative of how simple or cluttered the structure of an image is. In general, simple datasets have lower average number of segments. But even in the difficult PASCAL VOC 2009 dataset, the average was 154.

2.4. Mid-level Segment Ranking

Gestalt theorists [157, 111] argued that properties such as proximity, similarity, symmetry and good continuation are key to visual grouping. One approach would be to model such properties in the segmentation process, as long-range dependencies in a random field model [163, 165]. However, this poses significant modeling and computational challenges. With a segment set generated using weaker constraints, leveraging Gestalt properties becomes easier: rather than guide a complex inference procedure based on higher-order, long-range dependencies, we only need to check conformance with Gestalt regularities. It is therefore interesting to explore how the qualitative Gestalt theories can be implemented in such a framework and what effects they produce in practice. An important question is whether Gestalt properties can be used to predict if segments have regularities typical of projections of real objects, without leveraging prior knowledge about the classes of objects present in the image. This is a poten-

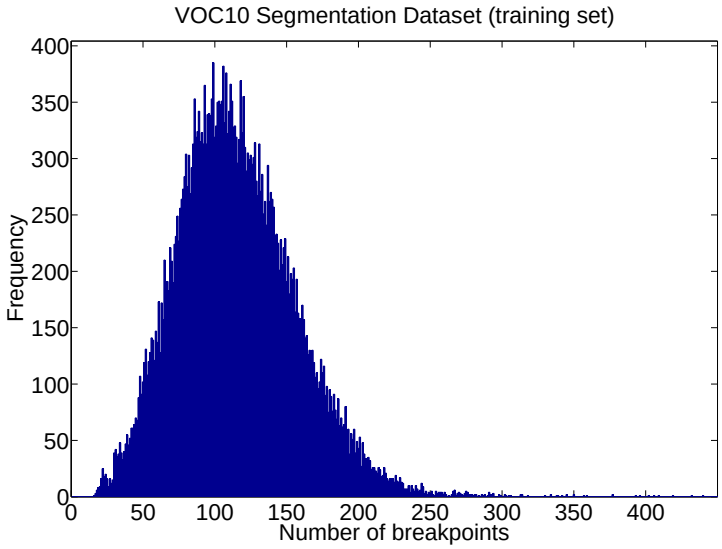


Figure 2.3.: Frequency of the parametric max flow breakpoints for each seed, on the training set of the VOC2010 segmentation dataset. These results were obtained using a 6x6 uniform grid of seeds. The number of breakpoints has mean 110, and a heavier tail towards a larger number of breakpoints.

tially challenging decision problem, since the visual aspects of objects are extremely diverse. However, if object regularities can be identified, images could be represented by a handful of segments, which are easier to interpret and process by higher-level visual routines than a large set of pixels or superpixels.

In this work, we take an empirical approach: we compile a large set of features and annotated examples of segments of many objects from different categories, and use machine learning techniques to uncover their significance. Three sets of features (34 in total) are considered to describe each segment, representing graph, region and Gestalt properties. Graph properties, in particular variations of cut values, have long been used as cost functions in optimization methods for segmentation. Region properties encode mainly the statistics of where and at what scale objects tend to appear in images. Finally, Gestalt properties include mid-level cues like convexity and continuity, which can encode object regularities (*e.g.* objects background segments are usually non-convex and object boundaries are usually smoother than the boundaries of accidental segments).

Graph partition properties (8 features) include the *cut* (sum of affinities along the segment boundary) [158], the *ratio cut* (sum of affinity along the boundary divided by their number) [156], the *normalized cut* (ratio of cut and affinity inside foreground, plus ratio of cut and affinity on background) [131], the *unbalanced normalized cut* (cut divided by affinity inside foreground) [130], and the *boundary fraction of low cut*, 4 binary variables signaling if the fraction of the cut is larger than a threshold, normalized by segment perimeter, for different thresholds.

Region properties (18 features) include area, perimeter, relative coordinates of the region centroid in the image, bounding box location and dimensions, major and minor axis lengths

of the ellipse having the same normalized second central moments as the region, eccentricity, orientation, convex area, Euler number, diameter of a circle with the same area as the region, ratio of pixels in the region to pixels in the total bounding box, perimeter and absolute distance to the center of the image. Some of these features can be easily computed in Matlab using the *regionprops* function.

Gestalt properties (8 features) are implemented mainly as normalized histogram distances based on the χ^2 comparison metric: $\chi^2(x,y) = \sum_i \frac{(x_i-y_i)^2}{x_i+y_i}$ [29]. Let the texton histogram vector on the foreground region be t_f, and the one on the background be t_b. Then *inter-region texton similarity* is computed as the $\chi^2(t_f, t_b)$. *Intra-region texton similarity* is computed as $\sum_i \mathbf{1}(t_f(i) > k)$, with $\mathbf{1}$ the indicator function, and k a threshold, set to 0.3% the area of the foreground in our implementation. The textons are obtained using the globalPb implementation [4], which uses 65 nearest neighbor codewords.

Another two features we use are *inter-region brightness similarity*, defined as $\chi^2(b_f, b_b)$, with b_f and b_b intensity histograms with 256 bins, and *intra-region brightness similarity* defined as $\sum_i \mathbf{1}(b_f(i) > 0)$.

We also extract the *intra-region contour energy* as the sum of edge energy inside the foreground region, computed using globalPb, normalized by the length of the region perimeter. We also extract an *inter-region contour energy*, as the sum of edge energies along the boundary normalized by the perimeter.

Other Gestalt features we consider include *curvilinear continuity* and *convexity*. The first is the integral of the segment boundary curvature. We use an angle approximation to the curvature [23] on triplets of points sampled regularly (every 15 pixels in our tests). Convexity is measured as the ratio of areas of the foreground region and its convex hull.

All features are normalized by subtracting their mean and dividing by their standard deviation.

2.4.1. Learning

The objective of our ranking process is to identify segments that exhibit object-like regularities and discard most others. One quality measure for a set of segments with respect to the ground truth is **covering** [4]. Let S be the set of ground truth segments for an image, S' be the set of machine segments and $S'(r)$ the subset of machine segments at rank r or higher. Then, the covering of S by $S'(r)$ can be defined as:

$$C(S, S'(r)) = \frac{1}{N} \sum_{R \in S} |R| * \max_{R' \in S'(r)} O(R, R') \tag{2.4}$$

where N is the total number of pixels in annotated objects in the image, $|R|$ is the number of pixels in the ground truth segment R, and O is a similarity measure between two regions.

We cast the problem of ranking the figure-ground hypotheses as regression on $\max_{R \in S} O(R, R')$, the maximum similarity a segment has with a ground truth object, against the segment features. The idea is that if regression is accurate, the generated segments most similar to each ground truth will be placed at high ranks. Then many lower ranked segments can be dis-

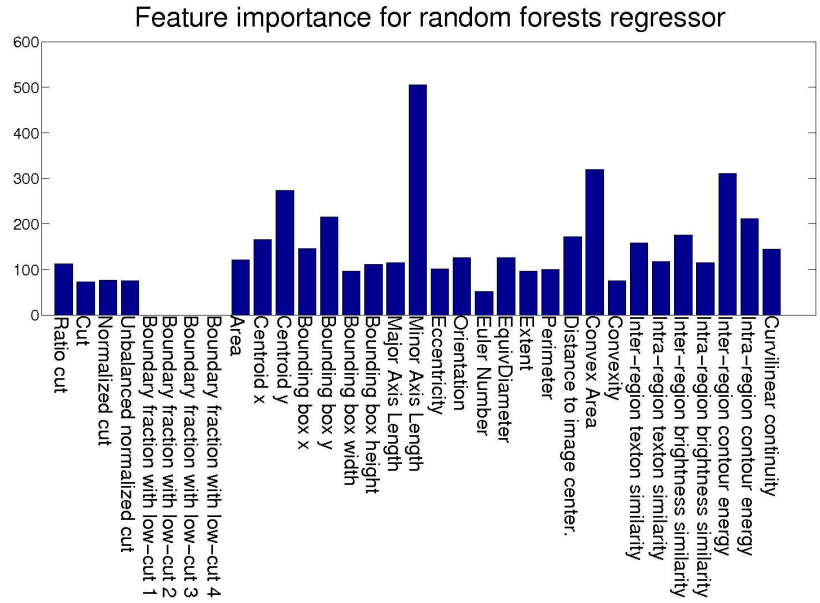


Figure 2.4.: Feature importance for the random forests regressor learned on the VOC2009 segmentation training set. The minor axis of the ellipse having the same normalized second central moments as the segment (here 'Minor Axis Length') is, perhaps surprisingly, the most important. This feature used in isolation results in relatively poor rankings however (see fig. 2.5a). The Graph properties have small importance. The 'Boundary fraction of low cut' features, being binary, do not contribute at all. Gestalt features have above average importance, particularly the contour energies.

carded without reducing the covering measure. As similarity measure O we use **overlap** [38]:

$$O(S,G) = \frac{|S \cap G|}{|S \cup G|} \tag{2.5}$$

which penalizes both under-segmentations and over-segmentations and is scale invariant. An alternative to overlap, which we used in one of our experiments, is the **F-measure** [124]:

$$F = \frac{2RP}{P+R} \tag{2.6}$$

where P and R are the precision and recall of pixels in a machine segment relative to a ground truth segment.

For ranking, we experimented with both linear regression and random forests [21], a competitive non-linear model that averages over multiple regression trees. We used a random forests implementation available online [75] and used default parameters, except for the number of trees, 200, and the number of candidate variables to select from, at random, at each split node, which we set to 10.

The *importance* of our features as learned by the random forests regressor [21], is shown in fig. 2.4. Some region properties appear to be quite informative, particularly features such as segment width and height and the location in the image. The 'Minor Axis Length' feature, which gets the highest importance works quite poorly in isolation, however (as illustrated

in fig. 2.5a), suggesting that some cues are only effective in conjunction with other features. Convexity and the edge energy along the boundary are also assigned large importance, as expected.

2.4.2. Maximum Marginal Relevance Diversification

Applying standard regression for ranking does not come without issues. Similar segments have similar features, which causes them to regress to the same values and be ranked in adjacent positions. The covering measure only considers the best overlap with each ground truth object, hence redundant segments in adjacent positions do not increase covering and tend to lower the ranks of segments that best overlap other objects. More segments then need to be retained to achieve the same score.

An effective way to deal with such effects is to **diversify** the ranking, in order to prevent that minor variations of a segment saturate the pool. We achieve this based on Maximal Marginal Relevance (MMR) measures [24]. To our knowledge this is the first application of this technique to image segmentation. Starting with the originally top-scored segment, the MMR induces an ordering where the next selected segment (with maximum marginal relevance) is the one maximizing the original score minus a redundancy measure with respect to segments already selected. This procedure is iterated until all segments have been re-ranked. The redundancy measure we employ is the overlap with the set of previously selected segments based on the MMR measure.

Formally, let H be the full set of figure-ground segmentations and $H_p \subset H$ hypotheses already selected. Let $s(H_i)$ be our predicted score for a given figure-ground segmentation and $o(H_i, H_j)$ the overlap between two figure-ground segmentations. The recursive definition for the next maximal marginal relevance selection [24] is given as:

$$MMR = \underset{H_i \in H \setminus H_p}{\operatorname{argmax}} \left[\theta \cdot s(H_i) - (1-\theta) \cdot \max_{H_j \in H_p} o(H_i, H_j)) \right]$$

The first term is the score and the second is the redundancy. Parameter θ regulates the trade-off between the predicted score and the diversity measures in the first N selections. For example with $\theta = 0$ the ranking will ignore individual scores, and select the next element in the set, having minimal overlap with any of the previously chosen elements. In contrast, with $\theta = 1$ the element with the highest score will always be selected next. The best trade-off depends on the application. If high precision is desired then a higher weight should be given to the predicted score, whereas if recall is more important, then a higher weight should be given to diversity. If θ is very small, then ranking will be close to random. For our VOC experiments we have cross-validated at $\theta = 0.75$.

2.5. Experiments

We study both the quality of the pool of object hypotheses generated by CPMC and the loss in quality incurred by selecting the topmost N object hypotheses, as opposed to the use of a much larger pool. We experiment with three publicly available datasets: Weizmann's Segmentation Evaluation Database [124], MSRC [133] and the VOC2009 train and validation sets for the object-class segmentation problem [38].

Weizmann consists of 100 gray-valued images having a single prominent foreground object. The goal is to generate coverage of the entire spatial support of the object in the image using a single segment, and as accurately as possible. We compare the performance of CPMC with published results from two state of the art segmentation algorithms. The results are reported using the **average best F-measure** criterion. For each ground truth object the most similar segment with respect to F-measure (eq. 2.6) is selected and the value of the similarity is recorded. These top similarities are then averaged.

The MSRC dataset is quite different, featuring 23 different classes, including some 'stuff' classes, such as water and grass. It has up to 11 objects present in each of its nearly 600 images. We use this dataset to evaluate the quality of the pool of segments generated, not the individual rankings.

The VOC 2009 dataset is challenging for segmentation, as it contains real-world images from Flickr, with 20 different classes of objects. The background regions are not annotated. In both MSRC and VOC2009, which contain multiple ground-truth objects per image we use the **covering** (eq. 2.4) with **overlap** (eq. 2.5) as a segment similarity measure.

2.5.1. Segment Pool Quality

The automatic results obtained using CPMC on the Weizmann dataset are shown in table 2.3a together with the previous best result, by Bagon et al [6], which additionally requires the user to click a point inside the object. We also compare with the method of Alpert *et al*. [124], which is automatic. Results for CMPC were obtained using an average of 53 segments per image. Visibly, it generates an accurate pool of segments. Results on MSRC and VOC2009 are compared in table 2.3b to Arbeláez *et al*. [4], which is arguably one of the state of the art methods for low-level segmentation. The methodology of the authors was followed, and we report average coverings. We use all the unique segments in the hierarchy returned by their algorithm [4] to compute the score. The pool of segments produced by CPMC appears to be significantly more accurate and has an order of magnitude fewer segment hypotheses. A filtering procedure could be used for gPb-owt-ucm to reduce the number segments, but at a potential penalty in quality. The relation between the quality of segments and the size of the ground truth objects is shown in fig. 2.7.

2.5.2. Ranking Object Hypotheses

We evaluate the quality of our ranking method on both the validation set of the VOC2009 segmentation dataset, and on hold-out sets from the Weizmann Segmentation Database. The training set of VOC2009 consists of 750 images, resulting in $114,000$ training examples, one for each segment passing the fast rejection step. On the Weizmann Segmentation Database we randomly select 50 images, resulting in $2,500$ training examples, and we test on the remaining 50 images.

We plot curves showing how well the ground truth for each image is covered on average, as a function of the number of segments we retain per image. The segments are added to the retained list in the order of their ranking.

The curve marked as 'upper bound' describes the maximum quality measure possible given the generated segments, which can be obtained if the segments are ranked by their known overlap with ground truth. Note that on Weizmann the upper bound is flat because each image has one single ground truth object, whereas on VOC images there can be multiple objects,

Weizmann	F-measure
CPMC	0.93 ± 0.009
Bagon *et al*. [6]	0.87 ± 0.010
Alpert *et al*. [124]	0.86 ± 0.012

(a) Average best F-measure scores over the entire Weizmann dataset. Bagon's algorithm produces a single figure-ground segmentation but requires a user to click inside the object. Alpert's results were obtained automatically by partitioning the image into one full image segmentation typically having between 2 and 10 regions. The table shows that for each image, among the pool of segment hypotheses produced by CPMC, there is usually one segment which is extremely accurate. The average number of segments that passed our fast rejection step was 53 in this dataset.

MSRC	Covering	N Segments
CPMC	0.85 ± 0.1	57
gPb-owt-ucm [3]	0.78 ± 0.15	670
VOC2009	**Covering**	**N Segments**
CPMC	0.78 ± 0.18	154
gPb-owt-ucm [3]	0.61 ± 0.20	1286

(b) Average of covering scores on MSRC and VOC2009 train+validation datasets, compared to Arbeláez *et al*. [4], here gPb-owt-ucm. Scores show the covering of ground truth by segments produced using each algorithm. CPMC results before ranking are shown, to evaluate the quality of the pool of segments from various methods.

Table 2.3.: CPMC segment quality on multiple datasets.

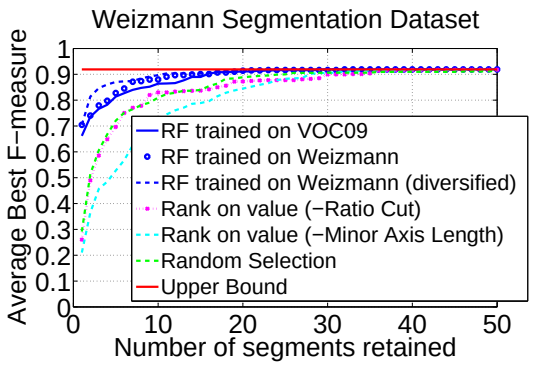
hence the upper bound increases as more than one segment is considered per image (on the horizontal axis). The curve labeled as 'random' is based on randomly ranked segments. It is a baseline upon which the ranking operation should improve in order to be useful.

On Weizmann we compare a random forests regressor trained on the images in that dataset with a predictor trained on VOC2009. The results in fig. 2.5a are similar, showing that the model is not overfitting to the statistics of the individual datasets. This also shows that it is possible to learn to rank segments of arbitrary objects, using training regions from only 20 classes. The learned models are significantly better than ranking based on the value of any single feature such as the cut or the ratio cut. On VOC2009 we have also run experiments where we have complemented the initial feature set with additional appearance and shape features — a bag of dense gray-level SIFT [99] features computed on the foreground mask, a bag of local shape contexts [9] computed on its boundary, and a HOG pyramid [17] with 3 levels computed on the bounding box fitted on the boundary of the segment, for a total of 1,054 features. In this case, we trained a linear regressor for ranking (this is significantly faster than random forests, which takes about 8 hours to train for the model with 34 features). The results are shown in fig. 2.5b. Clearly the new features help somewhat, producing results that are slightly better than the ones obtained by the linear regressor on the basic feature set. We will revisit them in §2.5.3. However, these are not better than a random forests model trained on the basic feature set. This shows that the set of basic features is already quite expressive in conjunction with nonlinear models.

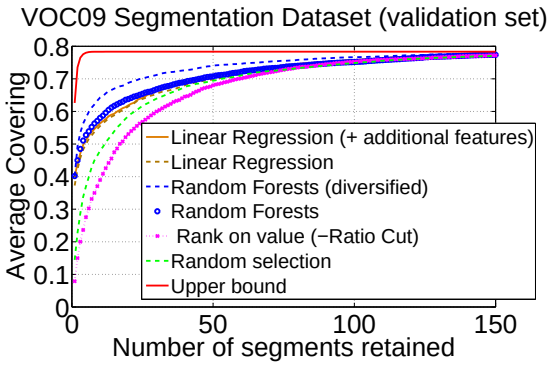
Notice that by using this ranking procedure, followed by diversification, we can obtain more accurate object hypotheses than those provided by the best existing segmentation algorithm of [4]. In fact, by using the top 7 segments produced by our ranking procedure, we obtain the same covering, 0.61, as obtained using the full hierarchy of $1,286$ distinct segments in [4].

2.5.3. Subframe-CPMC Extension

We have experimented with a different variant of the algorithm, the Subframe-CPMC, on the Pascal VOC2010 dataset. The goal was to achieve high object recall while at the same



(a) Average best segment F-measure as we vary the number of retained segments given by our ranking procedure. Results were averaged over three different splits of 50 training and 50 testing images. Note that when working with our top-scored 5 segments per image, the results already equal the ones obtained by the interactive method of Bagon *et al*. [6]. Note also that using this learned ranking procedure, it is possible to compress the original pool of segments to a fifth (10 segments), at negligible loss of quality.



(b) Complementing the basic descriptor set with additional appearance and shape features improves the ranking slightly, but the basic set is still superior when used in conjunction with a more expressive random forests regressor. Further diversifying the ranking improves the average covering given by the first top N segments significantly.

Figure 2.5.: Ranking results on the Weizmann and VOC2009 datasets. Different rankers are compared with the optimal ranker ("Upper bound") and with random ranking ("Random selection").

time preserve segmentation quality, with a mindset towards detection applications. To score a detection hypothesis as correct, benchmarks such as the Pascal VOC require a minimum overlap between a correctly classified region and the ground truth. In addition, benchmarks disregard the area of the ground truth regions (*e.g*. an object with 500 pixels is just as important as one occupying the full image), hence what matters is not so much achieving high *covering* scores (which explicitly take into account the size of the segments), but high *overlap*.

Subframe-CPMC uses an additional type of seed, and is configured to generate a larger number of segments. First we make the overall process faster by solving the energy problems at half the image resolution. Quantitative results were equivalent. We also changed the seeding strategy to use a single soft background seed and increased the number of foreground seeds, by using a grid of 6x6 instead of the previous 5x5. We reduced the value of the σ parameter by 30% in eq. 2.3, resulting in more segments due to reduced affinities between neighboring pixels.

We have also complemented the existing seeds with *subframes*, background seeds composed of the outside of rectangles covering no more than 25% of the area in the image, with a single square foreground seed in the center. These seeds constrain segments to smaller regions in the image, as they force the possible contours to lie inside the rectangular region. This is especially helpful for segmenting small objects in cluttered regions, as can be seen in fig. 2.7. For this type of seed we also solve problems with and without a color unary term. Two alternative types of subframe seeds were tried: a 5x5 regular grid of square subframes of fixed dimension, with width set to 40% of the image, and bounding boxes from a deformable parts detector [43, 46] with default parameters, set to the regime of high recall but low precision. For the detector, we discard class information and keep the 40 top-scored bounding boxes smaller than a threshold C, in this case 25% of the image area. Subframe energy problems

Figure 2.6.: Segmentation and ranking results obtained using the random forests model learned on the VOC2009 training set, with the features described in sec. §2.4. The green regions are the segment foreground hypotheses. The first image on each row shows the ground truth, the second and third images show the most plausible segments given by CPMC, the last two images show *the least* plausible segments, and the fourth and fifth images show segments *intermediately* placed in the ranking. The predicted segment scores are overlaid. The first three images are from the VOC2009 validation set and rows 2, 4 and 6 show the diversified rankings, with $\theta = 0.75$. Note that in the diversified ranking, segments scored nearby tend to be more dissimilar. The last three rows show results from the Weizmann Segmentation Database. The algorithm has no prior knowledge of the object classes, but on this dataset, it still shows a remarkable preference for segments with large spatial overlap with the imaged objects, yet there are neither chariots nor vases in the training set, for example. The lowest ranked object hypotheses are usually quite small reflecting perhaps the image statistics in the VOC2009 training set.

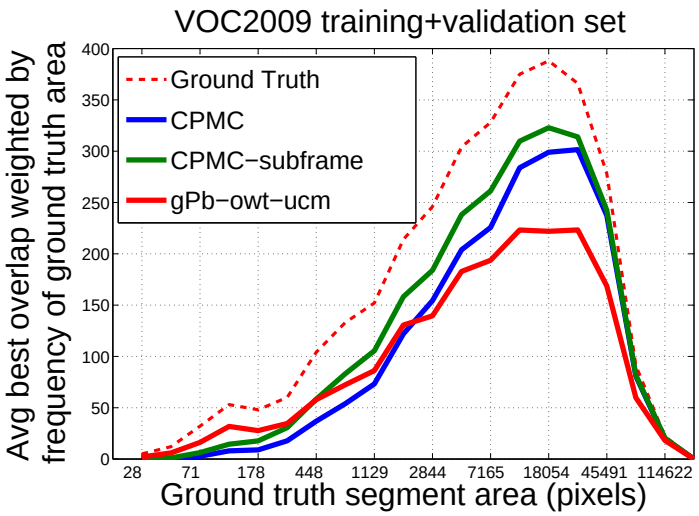


Figure 2.7.: Quality of the segments in the combined VOC2009 train and validation sets, as a function of the area of the ground truth segments. Object area has been discretized into 20 bins on a log scale. In the case of the ground truth curve the y-axis corresponds to the number of segments assigned in each bin (ground truth segments have an overlap value of 1 with themselves). Medium and large size objects, that are more frequent, are segmented significantly more accurately by CPMC than by gPb-owt-ucm [4]. Subframe-CPMC is competitive with gPb-owt-ucm on small objects, but generates a larger segment pool than plain CPMC (in the order of 700 instead of 150 elements).

are optimized efficiently by contracting all nodes corresponding to pixels belonging to background seeds into a single node, thereby reducing the size of the graph significantly.

The parameter σ, controlling the sharpness of the boundary, has an important influence on the number of generated segments. A value of 2.5 with the color-based seeds leads to 225 segments, average overlap of 0.61 and covering of 0.74, whereas for $\sigma = 1$ the method produces an average of 876 segments, average overlap of 0.69 and covering 0.76. We used $\sigma = 1$ for the uniform seeds, $\sigma = \sqrt{2}$ for the color seeds, and $\sigma = \sqrt{0.8}$ for the subframe seeds. This leads to a larger pool of segments, but also of higher quality, as noticeable in table 2.4.

Additional Features: Working with a larger pool of segments poses additional demands on the accuracy of ranking. An improvement we pursued was to enlarge the set of mid-level features with shape and texture descriptors. In §2.5.2 this was shown to improve results, but the dimensionality of these features made linear regression the most practical learning choice. A nonlinear random forests regressor on the basic feature set was still superior.

The additional shape and texture features we use are histograms, which are known to be most effective when used with certain nonlinear similarities, such as a Laplacian-RBF embedding $k(\mathbf{x}, \mathbf{y}) = \exp(-\sum |x_i - y_i|)$ [29]. Here we handle one of these similarity functions with linear regression, by first applying a randomized feature map to linearly approximate the Laplacian-RBF kernel [119, 96].

We adjusted the extended feature set from §2.5.2 slightly. To represent texture we extracted two bags of words for each segment, one defined over gray-level SIFT features as before and a new one over color SIFT features, both sampled every 4 pixels and at 4 different scales (16, 24,

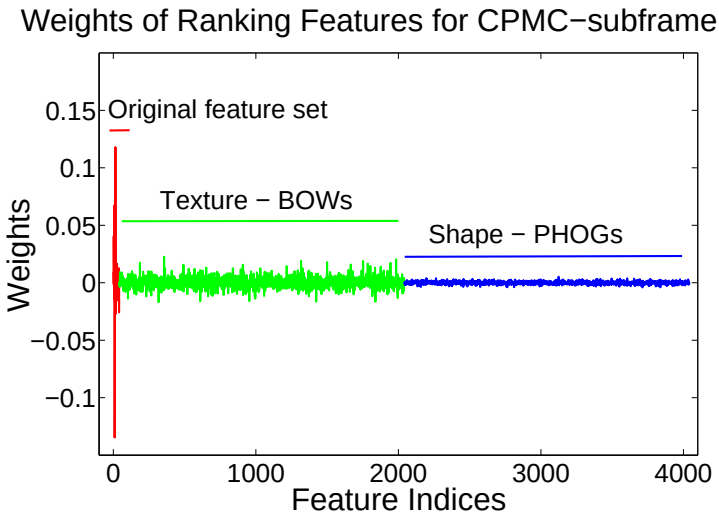


Figure 2.8.: Learned feature weights for the Subframe-CPMC model. The original set of mid-level features and region properties gets higher weights, texture features get intermediate weights and shape features get smaller weights. Texture features might help discard amorphous 'stuff' regions such as grass, water and sky.

36 and 54 pixels wide) to ensure a degree of scale invariance. Each feature was quantized using a 300-dimensional codebook. To represent shape we computed two pyramid HOGs, both with gradient orientation quantized into 20 bins, the first with the background segment gradients masked out on a pyramid composed of four levels, for a total of $1,700$ dimensions. The other PHOG was computed directly on the contour of the segment, with both foreground and background gradients masked out and a pyramid of three levels for a total of 420 dimensions. We map the joint vector of the two bags of words for texture features into a $2,000$-dimensional randomized feature map drawn from the Fourier transform of the Laplacian-RBF kernel [119], and process similarly the two PHOGs corresponding to shape features. We also append our original 34-dimensional feature set resulting in a total of $4,034$ features.

VOC2010 Results: The overlap measure is popular for distinguishing hits from misses in detection benchmarks. In the VOC2010 dataset we evaluate the recall under two different hit-metrics: 50% minimum segment overlap and 50% minimum bounding box overlap. Using the 50% segment overlap criterion, the algorithm obtains, on average per class, 87.73% and 83.10% recall, using 800 and 200 segments per image, respectively. Under a 50% bounding box overlap criterion, the algorithm achieves 91.90% when using 800 segments and 87.65%, for 200 segments.

The top 200 ranked segments gave on average 0.82 covering and 0.71 best overlap, which improves upon the results of CPMC without subframes on the VOC2009 (0.78 and 0.66 with all segments). These results are made possible because of the richer pools of segments, but also because the ranking is accurate. A reduction of on average around 500 segments per image results only in a loss of 0.03 average best overlap.

Details are shown in figs. 2.11 and 2.12, whereas image results are shown in fig. 2.9. The learned weights of the linear regressor for all features are displayed in fig. 2.8.

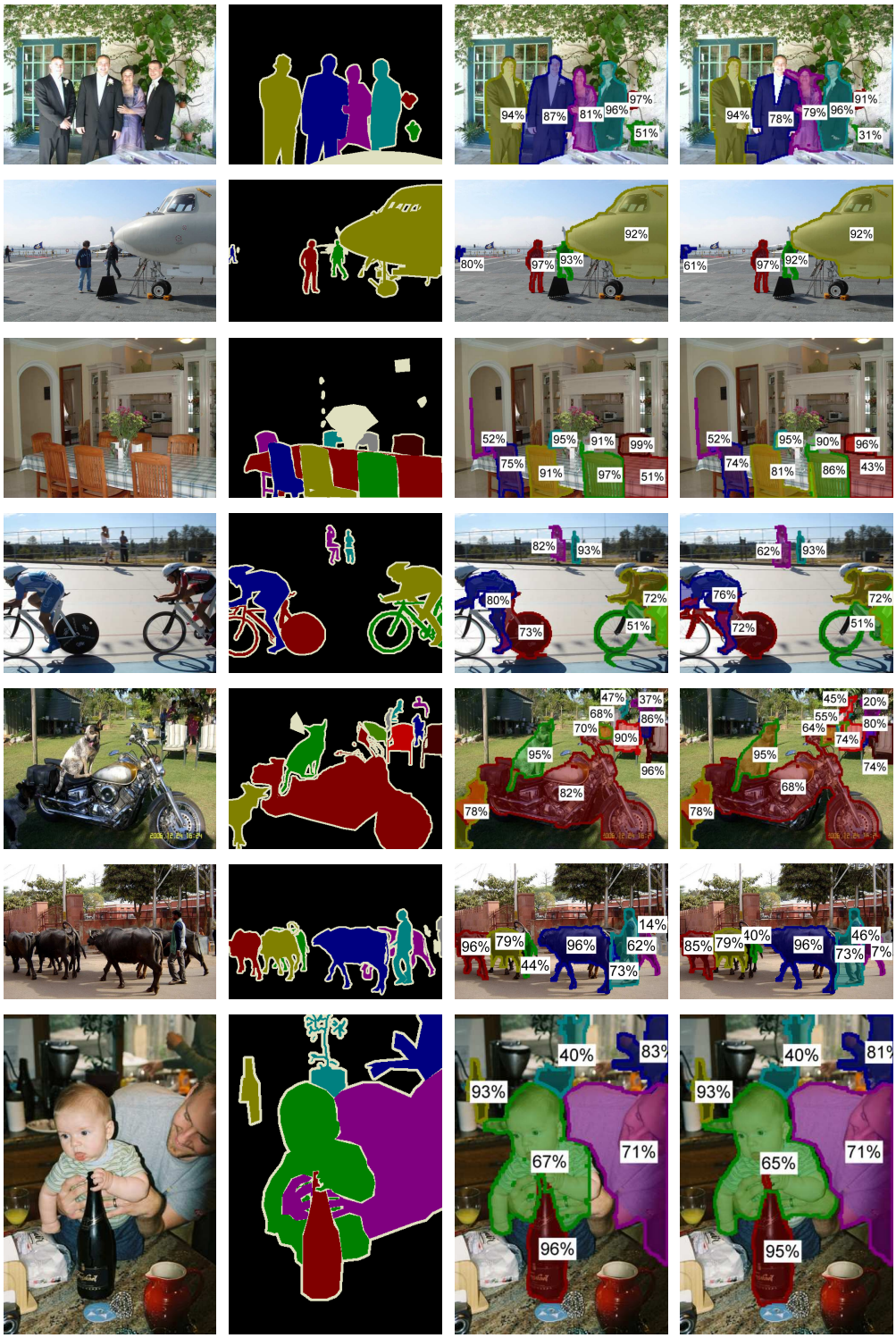


Figure 2.9.: Segmentation results on images from the validation set of the VOC2010 database. The **first** column contains the original images, the **second** gives the human ground truth annotations of multiple objects, the **third** shows the best segment in the Subframe-CPMC pool for each ground truth object, the **fourth** shows the best segment among the ones ranked in the top-200. The proposed algorithm obtains accurate segments for objects at multiple scales and locations, even when they are spatially adjacent. See fig. 2.10 for challenging cases.

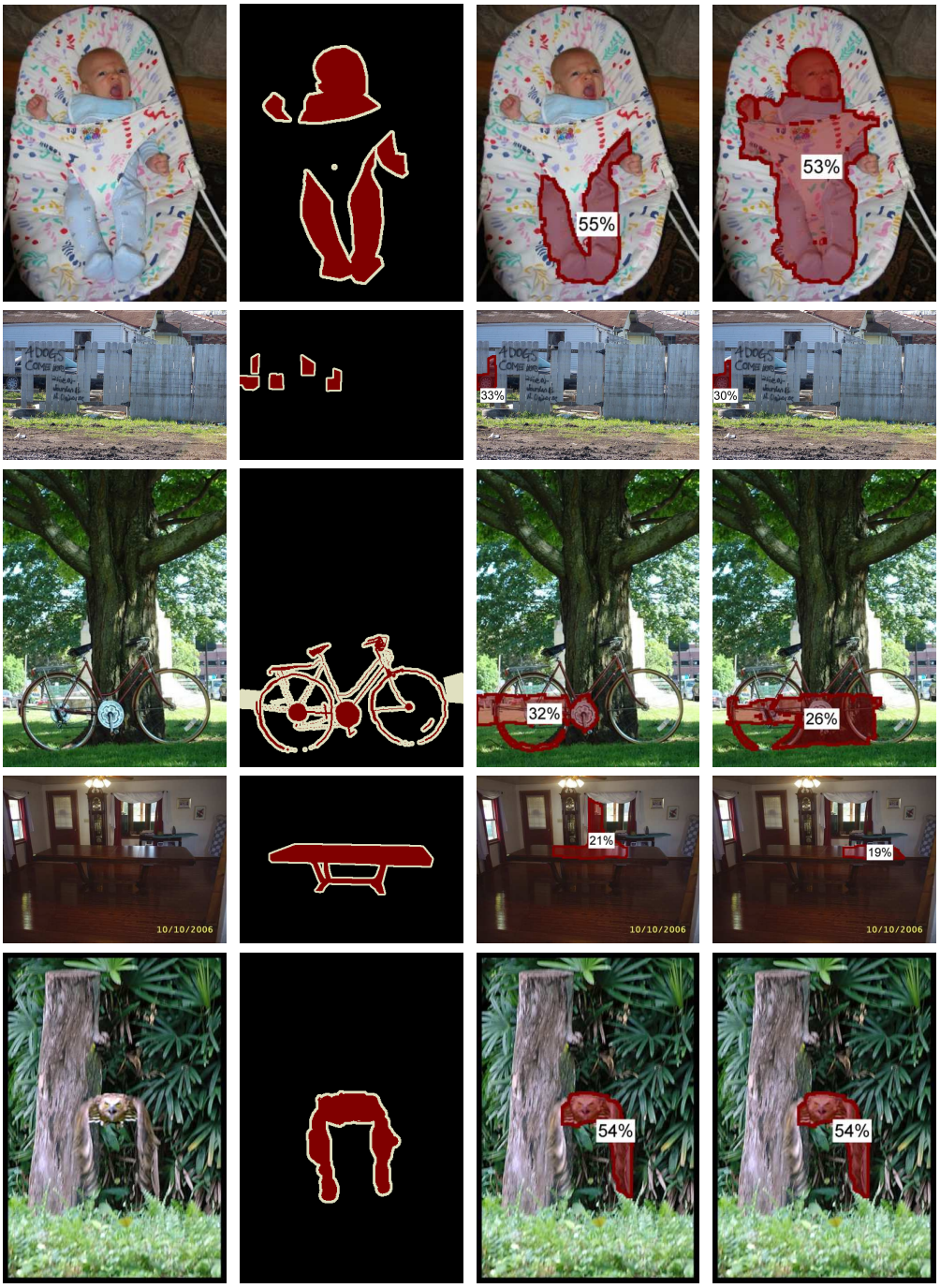


Figure 2.10.: Examples, taken from the validation set of VOC2010, where the CPMC algorithm encounters difficulties. The **first** column shows the images, the **second** the human ground truth annotations of multiple objects, the **third** shows the best segment in the entire Subframe-CPMC pool for each ground truth object, the **fourth** shows the best segment among the ones ranked in the top-200. Partially occluded objects (first two rows), wiry objects (third row) and objects with low background contrast (fourth and fifth row) can cause difficulties.

Quality Measure	Grid Subframes	BB Detector	No Subframes
Overlap	0.74	0.76	0.71
Covering	0.83	0.84	0.82
N segments	736	758	602

Table 2.4.: Results on the training set of the VOC2010 segmentation dataset. Color and uniform seeds are complemented with subframe seeds, either placed on a regular grid or obtained from a bounding box detector. Using a regular grid gives only slightly inferior results compared to results obtained using detector responses. Both give a large improvement in the recall of small objects, compared to models that do not use subframes. This is reflected in the overlap measure, which does not take into account the area of the segments.

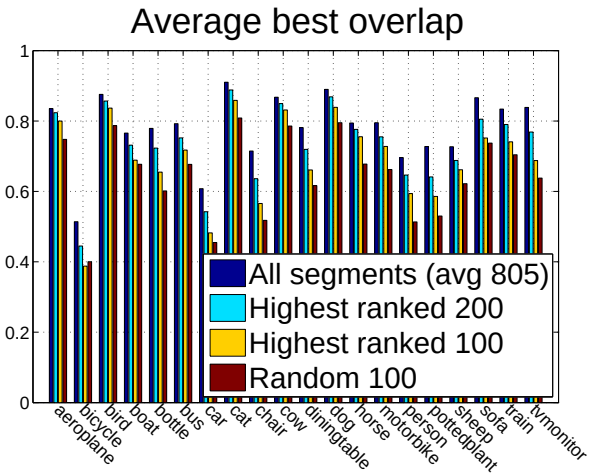


Figure 2.11.: Average overlap between ground truth objects and the best Subframe-CPMC segments on the validation set of VOC2010. We compare results obtained when considering all segments, just the top ranked 100 or 200 and a baseline that selects 100 segments randomly from the pool of all segments. Certain classes appear to be considerably harder to segment, such as bicycles, perhaps due to their wiry structure.

2.6. Conclusions

We have presented an algorithm that casts the automatic image segmentation problem as one of generating a compact set of plausible figure-ground object hypotheses. It does so by learning to rank figure-ground segmentations, using ground truth annotations available in object class recognition datasets and based on a set of low and mid-level properties. The algorithm uses a very powerful new procedure to generate a pool of figure-ground segmentations — the Constrained Parametric Min-Cuts (CPMC). This uses parametric max-flow to efficiently compute non-degenerate figure-ground hypotheses at multiple scales on an image grid, followed by maximum relevance ranking and diversification. We have shown that the proposed framework is able to compute compact sets of segments that represent the objects in an image more accurately than existing state of the art segmentation methods. These sets of segments have been used successfully in segmentation-based recognition frameworks [95, 72], as well as for multi-region image segmentation [27, 71] and cosegmentation [153].

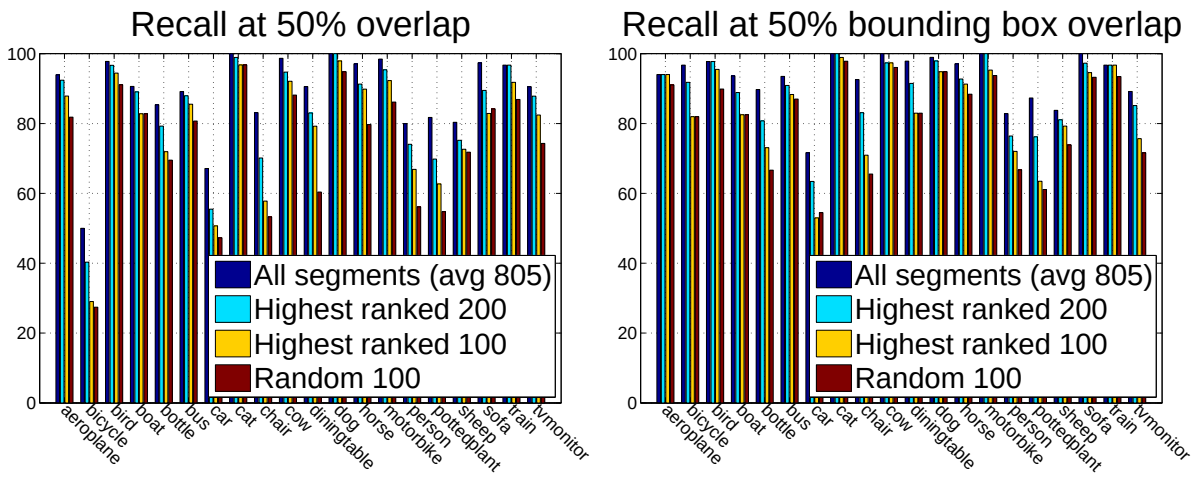


Figure 2.12.: Recall at 50% overlap between regions of ground truth objects and the best Subframe-CPMC segments (**top**) and between ground truth bounding boxes and best Subframe-CPMC segment bounding boxes (**bottom**). Note that bicycles are difficult to segment accurately due to their wiry structure, but there is usually some segment for each bicycle that has an accurate bounding box, such as the ones shown in the third row of fig. 2.2. These results are computed on the validation set of the VOC2010 segmentation dataset.

One difficulty for the current method is handling objects composed of disconnected regions that may arise from occlusion. While the energy minimization problems we solve sometimes generate such multiple regions, we chose to separate them into individual connected components, because they only rarely belong to the same object. In fact, in many such cases it may not be possible to segment the object correctly without top-down information. For exemple segmenting people embraced might require the knowledge of the number of arms a person has, and the configurations they can be in. It might be possible to handle such scenarios in a bottom-up fashion in simple situations, when cues like strong continuity may be exploited, but it appears more adequate to do this analysis at a higher level of scene interpretation.

The low-level segmentation and ranking components are also susceptible to improvement. Both components perform satisfactorily conditioned on the current state-of-the-art and datasets. One promising direction to improve the segmentation is the development of more sophisticated unary terms. Other advances may come from minimizing more powerful energy functions or the use of additional representations beyond regions. For example curves [142] may be more appropriate for objects that have long 'wiry' structures such as bicycles. The ranking component can be improved by developing better learning methodology, better features and by using more training data. At this point the segmentation component seems to allow the most improvement, but if applications set stringent constraints with respect to the maximum number of segments retained per image then ranking can become a bottleneck.

A somewhat suboptimal aspect of the proposed method is that energy minimization problems are solved independently, and the same number of problems is generated for all images, notwithstanding some having a single object and others having plenty. An interesting extension would make the process dynamic by making decisions on where and how to extract more segments conditioned on the solutions of the previous problems. This would be conceivably more efficient and would make the transition to video smoother. A sequential, conditional process could also make for a more biologically plausible control structure.

Chapter 3.

Object Recognition as Ranking Holistic Figure-Ground Hypotheses

Abstract We present an approach to visual object-class segmentation and recognition based on a pipeline that combines multiple figure-ground hypotheses with large object spatial support, generated by bottom-up computational processes that do not exploit knowledge of specific categories, and sequential categorization based on continuous estimates of the spatial overlap between the image segment hypotheses and each putative class. We differ from existing approaches not only in our seemingly unreasonable assumption that good *object-level segments* can be obtained in a feed-forward fashion, but also in formulating recognition as a regression problem. Instead of focusing on a one-vs.-all winning margin that may not preserve the ordering of segment qualities inside the non-maximum (non-winning) set, our learning method produces a *globally consistent* ranking with close ties to segment quality, hence to the extent entire object hypotheses are likely to spatially overlap the ground truth. We demonstrate results beyond the current state of the art for image classification, object detection and semantic segmentation, in a number of challenging datasets including Caltech-101, ETHZ-Shape as well as PASCAL VOC 2009-11.

This chapter corresponds to the journal article *Object Recognition by Sequential Figure-Ground Ranking*, by João Carreira, Fuxin Li and Cristian Sminchisescu, IJCV, which extends *Object Recognition as Ranking Holistic Figure-Ground Hypotheses*, by Fuxin Li, João Carreira and Cristian Sminchisescu, CVPR 2010. In both cases the first two authors contributed equally.

3.1. Introduction

Recognizing and localizing different categories of objects in images is essential for scene understanding. Approaches to object-category recognition based on sliding windows have recently been demonstrated convincingly in difficult benchmarks [155, 46, 149]. By scanning the image at multiple locations and scales, recognition is phrased as a binary decision problem for which many powerful classifiers exist. Recent developments have shown that scanning hundreds of thousands of windows efficiently can be feasible for certain types of features and classifiers [149, 11]. The bounding box approach to recognition has proven successful for object categories with stable features that can 'fill' the correct window significantly, like faces or motorbikes, it nevertheless tends to be unsatisfactory for objects with more complex appearance and geometry, or for advanced tasks such as pose prediction and action recognition where the knowledge of an object's shape is also important.

This motivates the focus on *semantic segmentation*, where the objective is to both identify the spatial support of objects, and to recognize their category. In semantic segmentation, the brute-force sliding windows approach to generic category recognition may not be feasible.

(a) (b)

Figure 3.1.: (a) A girl relaxing on a bench. Both top-down approaches and bottom-up sliding window methods can encounter difficulties segmenting or detecting a person in this non-canonical pose. (b) Semantic segmentation results produced by our algorithm.

Consider fig. 3.1 (a). A reliable object detector might locate the person and place a bounding box around her. However, the non-canonical pose may impose a large bounding box, or alternatively a large search space if different rotations of the bounding box are scanned, still leaving a non-trivial contour hypothesis space to be explored, even inside the correct bounding box, *e.g.* fig. 3.1 (b).

The semantic segmentation problem could be approached top-down [15, 91], by storing exemplars to guide the search in new images. However, since the variability of object shapes is large, only an approximate contour alignment between the training exemplars and new object instances can be expected. Interesting solutions have been proposed recently, although generalization to a large class of shapes remains non-trivial [84, 93]. In fact, some of the best performing methods for semantic segmentation currently do not employ shape priors but directly classify individual pixels, based on statistics of patches enclosing them [133, 33, 85].

An open problem for segmentation and recognition is the design of tractable models capable to make more informed decisions using increased spatial support. It appears necessary to be able to work at some intermediate spatial scale, ideally on segments that can model entire objects, or at least sufficiently distinct parts of them. The idea of doing recognition on segments larger than just piecewise uniform regions (superpixels) is not new, but has been barred for a long time by the lack of progress in reliably obtaining such segments. However, recent developments in segmentation algorithms provide a surprisingly effective solution [26]. For most images, the Constrained Parametric Min Cuts (CPMC) algorithm can generate a set of $20-200$ figure-ground hypotheses, among which segments covering full objects are extracted with high probability (see fig. 3.2). This motivates our exploration of visual recognition directly from a pool of holistic segment hypotheses extracted bottom-up. Recognition proceeds similarly with sliding windows methods, but in the drastically reduced search-space of plausible object segments. This enables the use of more powerful learning machinery based on multiple features and nonlinear kernels, trained with a large number of segments with different degree of overlap with the target object.

Besides leveraging recent progress in figure-ground segmentation methods for recognition, we contribute with a formulation that casts recognition as a one-against-all regression problem of predicting the quality of segments. The quality of a segment for a given category is measured as the maximum amount of overlap between the segment and a ground truth object of that category. Therefore, the correct category can be simultaneously determined from the predicted qualities for each of the multiple classes. This makes it possible to use all in-

formation available in those segments that only partially overlap with the ground truth and, we show, gives a significant boost in the recognition performance. We further develop a sequential recognition strategy that can identify multiple spatial supports and analyze images containing several objects from different categories.

The chapter is organized as follows, Section 2 reviews related work. Section 3 describes the overall framework. Sections 4-6 describe the three main components of the framework: segment ranking (§4), segment scoring and categorization (§5), and sequential segment post-processing (§6). In section 7, we test the various components of the system and report state-of-the-art results on three object recognition tasks: image classification, object detection and semantic segmentation. Section 8 concludes the chapter and discusses ideas for future work.

Figure 3.2.: Examples of segments used in the recognition process. Clearly, among the multiple figure-ground hypotheses generated by CPMC [26] there are good segments that cover the object of interest entirely. The challenge for recognition is to pull them out.

3.2. Related Work

We will confine our review of the state of the art to recognition techniques that estimate the spatial layout of objects. These techniques can be broadly classified as bottom-up or data-driven and top-down or model-based, although the separation is to some extent blurred as many methods have both bottom-up and top-down components.

Bottom-up Recognition. Bottom-up recognition techniques use no prior shape knowledge to obtain the object regions. They often either categorize among a set of predefined region hypotheses, like our method, or directly classify pixels.

Rabinovich *et al*. [118] use a stability heuristic [117] to select a reduced list of segmentations obtained using normalized cuts [131] for different number of segments and different cue combinations. Segments are described by bags of features and those with the highest label confidence given by a k-nearest neighbor classifier are retained. Malisiewicz and Efros [103] generate a large pool of segments [102] and recognize them using a nearest-neighbor classifier based on learned distance functions. Todorovic and Ahuja [139], compute a hierarchical segmentation and find object subtrees similar to those learned during training. Unlike other methods they also model the relationship between objects and their subregions. A difficulty to overcome is the reliance on the structure of the hierarchical segmentation, which may not always be stable.

Another set of bottom-up approaches decides the object category directly at the level of image pixels [65, 134], or superpixels [52, 56], based on features extracted over a supporting neighborhood. Textonboost [134] classifies each pixel using a linear predictor on texton-layout features, learned using boosting. These features count the number of occurrences of a particular texton in a rectangular region at locations relative to each pixel. Because the output of local predictors can be noisy, often these approaches impose spatial constraints in a Conditional Random Field (CRF) framework to obtain smoother solutions. Smoothness can be obtained using contrast-sensitive pairwise potentials [19], which facilitate label transitions at image discontinuities, or higher-order P^n potentials [78] defined over extended image segments. These aim to bias the results towards solutions with small label variation inside homogeneous segments.

A common property of many approaches is the extraction of features over overlapping spatial supports, in order to increase robustness. One variant combines pixel and global image predictions [34, 56]. Another variant adds predictions over extended regions obtained from low-level image segmentations [85]. Instead of reconciling predictions over overlapping regions, Gould *et al*. [57, 58] minimize an energy function over both the set of image segmentations and their labeling. Pantofaru *et al*. [112] notice that pixels grouped together by all segments in different image partitionings should have the same label and average category predictions on superpixels obtained by intersecting all segments.

A difficulty for pixel-level methods is segmenting multiple nearby instances of the same object without modeling the objects globally. This limitation has been partially addressed recently by adding rectangular bounding box detection constraints [58, 86] to a global energy formulation. In our method segments and their associated class scores are used instead. Arguably these are closer to the desired ground truth spatial object layout than bounding boxes.

Model-based Recognition. An alternative to bottom-up recognition is the use of shape models to constrain estimates of the spatial support of objects. This does not rule out models with bottom-up components that still use high-level information to obtain the final segmentation.

One class of model-based approaches assumes that object parts correspond to homogeneous image regions and these can be computed reliably. The methods assemble homogeneous image segments into full objects [108, 136, 31] using knowledge of their part decomposition. Mori *et al*. [108] first detect key parts among salient segments obtained using the output of the Normalized Cuts algorithm, then solve a constraint satisfaction problem to find probable configurations. Srinivasan and Shi [136] compute several independent Normalized Cut segmentations by varying the number of clusters, then search for high-scoring interpretations obtained by assembling parts starting from those positioned lower in the image. Partial object segmentations obtained after each merge operation are matched against shape exemplars and used to prune implausible hypotheses. Cour and Shi [31] show how to efficiently select sets of superpixels that best match an object template under a Hamming distance comparison metric. They first locate a set of parts, then repeat the process to assemble them into complete object hypotheses.

The difficulty of consistently segmenting object parts motivates another class of approaches that does not rely on low-level image segmentation. One possibility is to search densely for object parts, then form segmentations by assembling stored partial ground truth responses associated with each part. Borenstein and Ullman [15] segment objects in new images by combining partial ground truth segmentations associated with object fragments in training

images. They identify putative fragments at image locations where the value of a predefined correlation function is maximal, then select those that locally optimize a cost function that combines the relevance of identified fragments, the value of their image correlation and a global consistency criterion. Leibe *et al*. [91] employ a related top-down idea, but instead of convolving the image with masked fragments, compute descriptors on scale-invariant interest points and use a voting scheme to select consistent subsets.

As objects appear in a large variety of poses and shapes, dominantly top-down methods produce object segmentations that are often qualitative and can miss image detail. One way to improve such results is to integrate low-level information as image edges [84, 140] or bottom-up hierarchical segmentations [16]. Yu and Shi [161] solve a constrained eigenvalue problem to find object segmentations biased by both object patch correlation and low-level edge alignment. Schoenemann and Cremers [128] solve a minimum ratio cycle problem on a product graph consisting of responses on the boundary of a shape template. The Objcut method [84] computes a segmentation biased both by low-level image cues and the output of a part-based probabilistic object-class model (pictorial structure) by solving a single min-cut problem. Toshev *et al*. [140] developed a boundary structure segmentation technique that uses new chordiogram shape descriptors that make possible to match an image to an exemplar and simultaneously compute a binary segmentation as the result of a semi-definite programming relaxation.

Some techniques use more detailed processing only after a bounding box is obtained, being natural extensions to object detection methods. Yi *et al*. [159] compute object bounding boxes using a deformable parts detector [46] and use color cues and simple shape priors on the bounding box and the rectangular parts returned by the detector to obtain a segmentation. Gu *et al*. [63] vote for the location and scale of bounding boxes based on matches between regions in the image and regions inside exemplar bounding boxes. They assign confidence scores to foreground and background regions and propagate these decisions to the rest of the image based on low-level similarities, by constraining an initial segmentation obtained using Ultrametric Contour Maps [2].

3.3. Method Overview

Our recognition methodology relies on figure-ground segments generated by bottom-up computational processes. Our initial processing step produces a set of figure-ground segmentation hypotheses (out of which only figure segments are retained) for each image using the combinatorial CPMC segmentation algorithm [26] (fig. 3.2). The number of segments in this set depends on the image content: images with more edge structures tend to have more segments. Once segmentation hypotheses are obtained, the recognition framework consists of three stages: (1) segment ranking and filtering, (2) segment categorization and, (3) sequential aggregation and post-processing of multiple categorized segments.

The full recognition pipeline is depicted in fig. 3.3. In the first stage, a *class-independent* quality function is learned in order to rank all segment hypotheses. This mid-level step separates segments with object-like regularities from those that do not have them. Based on the ranking produced in this step, a maximum (fixed) number of segments is selected for each image. These will be used for training and testing in later stages. This number depends on the difficulty of the dataset and is usually much smaller than the average number of segments generated by the algorithm (40—100 in our experiments). While our segmentation method is

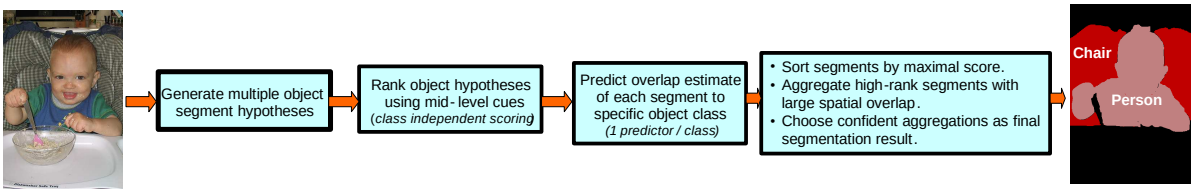


Figure 3.3.: Our semantic segmentation pipeline. Initially, an image is segmented into multiple figure-ground hypotheses constrained at multiple image locations and spatial scales, these are ranked (using mid-level cues) based on their plausibility to exhibit 'object-like' regularities (CPMC algorithm [26]). Quality functions for different categories are learnt to rank the likelihood of segments to belong to each class. Several top-scoring segments are selected for post-processing. The final spatial support and the category labels are obtained sequentially from these segments, based on a weighted sum of selected segment scores.

based on CPMC [26], additional processing is implemented in the framework, and this will be described in detail in section 3.4.

In the second stage, we learn a continuous scoring function for each object category, to assess the likelihood that a segment hypothesis belongs to that class. We follow a one-against-all methodology: the scoring function for each category is trained with all the input segment hypotheses that correspond or not to that category. In this way, each of the scoring functions is also discriminative and separates well one class from the others.

In the final stage, we sort the segment hypotheses by their scores and sequentially make detection and segmentation decisions based on a weighted combination of responses collected at high-rank segments. Image classification results are generated by taking maximal scores over all classes and among all image segments.

One of the main innovative points of this work, besides using *multiple figure-ground segmentations* from CPMC (rather than, *e.g.*, different multi-region image segmentations at different scales), is that category learning is performed by *regressing* on a quality function measuring the spatial overlap with the ground truth segments. Different segments carry different levels of information. For instance, in fig. 3.2, a segment capturing the entire cow carries the most significant amount of information in determining its category. Parts of the animal, like the head, contain a lower, yet significant level of information. Segments that cover the cow and surrounding grassland provide context about where the cows can typically be found. Even background segments carry some information, *e.g.*, persistent mountain-grass segments show that this is a wilderness picture, and some objects like a sofa or a TV are unlikely in the scene.

Our regression-based training scheme is designed to more effectively (and accurately) exploit the various levels of information available in different segments. The quality function measures overlap with ground truth, which is a smooth measure of quality that degrades gracefully: full object segments have the highest overlap, parts of objects and surrounding segments have moderate overlap and dominantly background segments have the lowest (or no) overlap. By regressing on overlap, we more judiciously use partial information in all segments.

Prediction from our regression model generates a natural ranking of all segments based on their importance. This is illustrated in fig. 3.4. Our decision stage exploits this ranking to create an accurate object mask. We group together high-confidence segments that cover a similar region and attempt to consolidate a single mask (and its label) by integrating information from all segments. To achieve this, a confidence score is computed for each pixel as the

Figure 3.4.: An illustration of our segment categorization process. Each segment is given as input to regressors specialized for each category, producing estimated qualities. The maximal score across categories is used to sort segments and decide on their category.

weighted sum of scores of the segments that cover it. If all segments agree that a given pixel should belong to a given category, the likelihood of this assignment will be high. If there are conflicts, for example one segment votes that a pixel is more likely part of a dog whereas the other three vote for a cat, the confidence would decrease (see fig. 3.7). A learned threshold on the pixel confidence score determines if the pixel should be included in the final mask.

3.4. Segment Generation and Filtering

3.4.1. Basic Approach

The inputs to our processing pipeline are multiple figure-ground segmentations obtained by CPMC [26]. These are obtained by solving a series of constrained min-cut problems, for putative foreground seeds constrained on a regular image grid and for background seeds sampled as various subsets of pixels on image borders. Multiple significant scale breakpoints (solutions) for these problems are computed using parametric max-flow in polynomial time [53].

Ranking segments based on their mid-level properties is the second step in the framework. During this phase, the segments generated by CPMC are filtered based on a quality function learned by regression, with covariates chosen as mid-level segment properties and Gestalt features (see [26]). We additionally use SIFT and HOG descriptors computed on the foreground to augment the feature set used to predict segment quality. Section 3.5.1 provides detail on the computation of these histogram features.

The regression function we learn for segmentation is class-independent (there is a single such function in the framework), with input given by segment features and output given by the maximal overlap between a segment and all the ground truth segments. The scale of the problem rapidly runs into millions: for instance, a dataset of 2000 images and 1000 segments for each image gives rise to a problem with 2 million examples. Therefore, at first linear methods appear to be the only practical choice for learning. However, random Fourier approximations can be used to transform the features linearly, to accurately approximate non-linear similarity measures [119, 12, 150]. In the Fourier methodology we consider an initial kernel and generate a new set of features based on randomly sampling multiple components from its Fourier transform. A linear regressor working on the transformed rep-

resentation usually offers performance close to those of nonlinear kernel machines [119]. In this chapter, we use random Fourier approximations for all image features and for all kernels employed for class-independent ranking. The mid-level segment descriptors are transformed using random Fourier projections corresponding to a Gaussian kernel, and the histogram features (SIFT and HOG) are transformed separately using Fourier embeddings derived from the skewed chi-square kernel [96]. The resulting dimensions are concatenated to generate the final covariate vector.

Beside random Fourier approximations, we employ additional processing for segment ranking. In the next subsection we define a customized overlap measure that is better tailored to the performance metric used on the PASCAL VOC challenge [41]. In section 3.4.3 we show how to learn the class-independent ranking function using linear regression, for problems where it is no longer possible to load the entire training set into memory.

3.4.2. Quality Function

A common measure used to assess segmentation quality is the 'intersection-over-union' overlap, or IOU-overlap. Let S_p and S_q be two generic segments and G_q be a ground truth segment. IOU-overlap is defined as:

$$O_{iou}(S_p, S_q) = \frac{|S_p \bigcap S_q|}{|S_p \bigcup S_q|}. \tag{3.1}$$

Sample segments from an image and their IOU-overlap to the ground truth are shown in fig. 3.5. To show how different these can be, the best 4 segments (w.r.t. the ground truth segment) and the worst 4 segments are shown on the top and bottom rows. On the second and third row, selected segments that partially overlap the object are shown.

The choice of quality function *for training* is not confined to the original IOU-overlap used in [26]. Depending on the task, different quality functions can be used. For example, in the PASCAL VOC segmentation challenge, the performance measure places more importance on larger objects. Moreover, the accuracy of the background class is also measured, therefore segmentations that handle the background correctly are also preferred. These two constraints are not entirely accounted for by the standard IOU-overlap measure (3.1). It can be seen from fig. 3.5 that some of the very large segments have significant IOU-overlap with the ground truth object, although this is not desirable, in order to accurately classify the background.

To palliate some of these effects, we propose a new overlap measure for training that we refer to as the Foreground-Background Overlap, or FB-overlap. It accounts for both overlap with the foreground and overlap with the background, and compensates against large segments. The measure is computed as:

$$\begin{aligned} O(S_p, G_q) \quad = \quad & \frac{C\sqrt{|S_p|}}{\log|S_p|\sqrt{N_c^{fg}}} \frac{|S_p \bigcap G_q|}{|S_p \bigcup G_q|} \\ & + \frac{C\sqrt{|\overline{S_p}|}}{\log|\overline{S_p}|\sqrt{N^{bg}}} \frac{|\overline{S_p} \bigcap \overline{G_q}|}{|\overline{S_p} \bigcup \overline{G_q}|} \end{aligned} \tag{3.2}$$

where N_c^{fg} and N^{bg} are the number of foreground and background pixels in the entire training set, with c the class of the ground truth segment G_q, and $\overline{S}$ is the image complement of a

Figure 3.5.: (Best viewed in color) Segments with different overlaps with the ground truth. The two numbers shown are the proposed FB-overlap on the left and the standard IOU-overlap on the right. It can be seen that FB-overlap favors segments that do not contain a lot of background, whereas IOU-overlap is indifferent to such effects.

segment hypothesis. $C = 90$ is a normalization constant that scales the range of the measure so as to match the range of IOU-overlap on the VOC dataset. The class-independent quality function of the segment is computed as

$$O(S_p, I) = \max_{G_q \in I} O(S_p, G_q) \tag{3.3}$$

where I is the image where the segment resides in.

FB-overlap emphasizes large segments mildly, while still not penalizing significantly small to moderately sized segments – because the background is also considered, oversized segments are not preferred. From fig. 3.5, it can be seen that under the new measure, the segments that correspond to objects and parts tend to have higher rankings under FB-overlap than under IOU-overlap. Segments that overlap significantly with the background are given comparatively lower FB-overlap scores. Besides, FB-overlap provides a mechanism to balance the training set sizes among different classes. For example the class `person` in VOC has around 8 million training pixels, whereas `bicycle` has only around $300,000$. The overlap in the class of `bicycles` are made mildly higher under the FB-overlap measure in order to equalize the prediction accuracy among different classes.

The formula is derived using ideas from residual analysis [143] on the maximal predicted scores of the regression model (Section 3.5). Our principle in designing the scoring function is that although larger segments are to be favored in general, random segments (that do not correspond to any ground truth) of different size should have roughly the same predicted scores. During the design phase of the measure, the entire framework has been tested several times and changes to the measure were made. The end result is formula (3.2). In fig. 3.6 it can be seen that after tuning, the lower bound scores on all the segment sizes are roughly

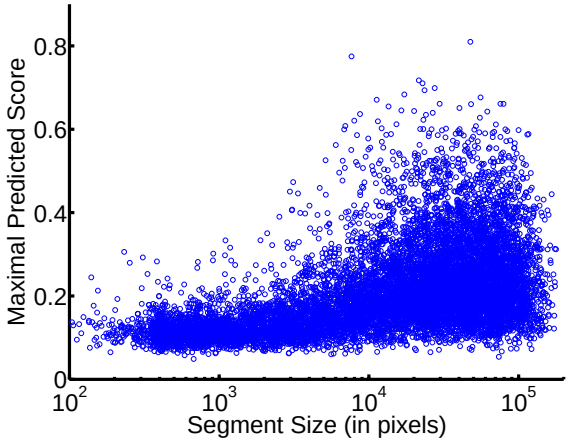


Figure 3.6.: Predicted FB-overlap on VOC 2010 validation dataset against size of the segment (in pixels). It can be seen that the lowest predicted score on segments of different size is roughly the same under the new FB-overlap measure.

similar. Overall, the use of FB-overlap improves the VOC result by around 1%. We will use the notation O for either overlap measure in the sequel. Notice however that FB-overlap will only be used in PASCAL VOC training, whereas IOU-overlap is used for all the other datasets.

3.4.3. Linear Regression with Partial-Storage

As the number of images and segments increases, they no longer fit into memory. Since SIFT and HOG features are not very sparse, a dense representation needs to be used. For instance, in the VOC 2010 dataset, there are around 10,000 images. We use 800 segments for each image and 3,600 Fourier feature dimensions as training data for segment ranking. This sums to 8 million examples, each having 3,600 dimensions. Storing the features using single precision (4 bytes) requires 107 gigabytes, which is beyond the current memory capacity of many personal computers. Some progress has been made in designing large-scale SVM classifiers [160], but those generally require loading the data into memory multiple times and are extremely time-consuming. Previous work on large-scale learning mostly focused on text categorization, but because those features are considerably sparser than in computer vision, the storage problem is less stringent.

In this work we take a simple approach. It is well-known that for least-squares and related methods, the problem can always be transformed into an optimization problem on the mean and the covariance matrix – the sufficient statistics of the Gaussian distribution [10]. These can be built from the data in chunks. Formally, in regression, our goal is to solve the quadratic optimization:

$$\min_w \sum_i (w^{\mathrm{T}} x_i - y_i)^2 + C\Omega(w) \tag{3.4}$$

where x_i represents segment features, y_i the overlap of a segment, e.g. eq. (3.2), and $\Omega(w)$ can

be any regularizer applied on w, e.g., $\|w\|_2^2$, $\|w\|_1$. This is equivalent to

$$\min_w w^{\mathrm{T}} X^{\mathrm{T}} X w - 2 w^{\mathrm{T}} X^{\mathrm{T}} y + C\Omega(w) \tag{3.5}$$

where $X^{\mathrm{T}}X = \sum_{i=1}^{n} x_i x_i^{\mathrm{T}}$ and $X^{\mathrm{T}}y = \sum_i x_i^T y_i$ can be computed by loading a single or a chunk of x_i into memory at a time. Therefore, all methods that use a quadratic loss function can work without loading all training data into memory. This includes ordinary least squares, ridge regression, lasso and group lasso methods. We work with ridge regression, under a quadratic regularization term $\Omega(w) = \|w\|_2^2$.

One common pitfall in applying the approach is normalization. For instance, if a standard normalization is to be performed ($x = \frac{x-\bar{x}}{\mathrm{std}(x)}$, where $\bar{x}$ is the mean and $\mathrm{std}(x)$ is the standard deviation), it is tempting to compute the mean and variance for each chunk of data separately because not all data can be loaded into memory simultaneously. However this shortcut does not work well—in our experiments we observed a performance drop of up to 2%. The correct mean and variance still need to be computed, although this means tediously loading the data chunk by chunk, computing $\sum_i x_i$ and $\sum_i x_i^2$ for each chunk, summing it up to obtain the mean and variance and loading the data again, in chunks, to normalize.

3.5. Segment Categorization

For categorization, we compute multiple figure-ground segmentations and extract multiple sets of features for them. A weighted sum of kernels on different types of features is used, with hyperparameters learned on the validation set. Based on the features and the coefficients of the kernel combination, support-vector regression on the overlap measure generates a scoring function for each object category.

3.5.1. Multiple Features

Features are extracted for each segment. We use 7 feature types. In order to model the object appearance we extract four bags of words of gray-level SIFT [99] and color SIFT [126], on a regular grid, two on the foreground and two on the background of each segment. Computing bags of words on the background of a segment models a coarse scene context.

To encode shape information we extract three pyramid HOGs (pHOG) [17], which are concatenations of histograms of gradients extracted at different resolutions. Each level of the pyramid divides each cell from the previous level into four higher resolution cells. The first level has a single cell. The first of our three pHOGs is defined directly on the contour of the foreground, whereas the other two operate on edges detected by globalPB [101] inside the foreground. The first two pHOGs adapt the cell dimensions in order to tightly fit the bounding box of the foreground segments, whereas the third uses square cells. The pHOG with square cells always covers a square region of the image, so we pad the image with zero, whenever this square region is partially outside the image. We use these different pHOGs so they can complement each other. The gradient orientation is discretized into 16 bins with values restricted between 0 and 180 degrees, as we chose to ignore the contrast direction.

A chi-square kernel $K(x, y) = \exp(-\gamma\chi^2(x, y))$ is used for each type of histogram features and we use a weighted sum of such kernels for regression. The coefficient and the width

hyperparameters of each chi-square kernel are learned using an optimization scheme detailed in subsection 3.5.3.

3.5.2. Learning Scoring Functions with Regression

Let us consider an image I with ground truth segments $\{G_q^I\}$. The segmentation algorithm provides a set of segments $\{S_p^I\}$ for image I. Denote also the K object categories $\{c_1, c_2, \ldots, c_K\}$. Let $\mathbf{1}(x)$ be the indicator function.

As discussed in the previous section, we learn K functions $f_1(S_p^I), \ldots, f_K(S_p^I)$ by regression on a quality measure for segments. For each putative segment S_p^I, we compute its overlap, given by (3.2), against all ground truth segments $\{G_q^I\}$ in the image. The target value y_{kp}^I for a segment S_p^I and a category c_k is the maximal overlap with ground truth segments that belong to c_k:

$$y_{kp}^I = \max_{G_q^I \in c_k} O(S_p^I, G_q^I). \tag{3.6}$$

Usually a segment S_p^I overlaps with at most a few ground truth segments. For categories that do not appear in an image I, $y_{kp}^I = 0$. After training, the estimated qualities for S_p^I on improbable categories tend to be close to 0. Therefore, this regression scheme is able to both estimate the quality of segments and classify them into categories.

To learn the function $f_k(S_p^I)$ for each c_k, we use a nonlinear support vector model (SVR) to regress on y_{kp}^I against x_p^I, the features extracted from segments S_p^I. The SVR optimization problem can be derived as:

$$\begin{aligned} \min_{w,\xi,\eta} \quad & \tfrac{1}{2}\|w\|^2 + C\textstyle\sum_{i=1}^n \xi_i + C\sum_{i=1}^n \eta_i \\ \text{s.t.} \quad & \xi_i \geq 0; \eta_i \geq 0, \forall i \\ & \langle w, \phi(x_i)\rangle \geq O(y_i, y) - \epsilon - \eta_i \\ & \langle w, \phi(x_i)\rangle \leq O(y_i, y) + \epsilon + \xi_i \end{aligned} \tag{3.7}$$

where $\phi(x_i)$ is a nonlinear feature transform of the input x_i, defined implicitly by the kernel $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j)\rangle$ detailed in the next section; ϵ is a small constant, usually 0.05 or 0.1. Using the kernel trick, it is possible to represent $f(S_p^I)$ in dual form as $f(S_p^I) = \sum_i \alpha_i K(x_i, x_p^I)$, where x_i are support vectors from the training set, and the α are coefficients obtained by the SVR optimizer.

The maximal score and the final segment category are given, respectively, by $\max_k f_k(S_p^I)$ and $\arg\max_k f_k(S_p^I)$. However, scores on all categories will be used in the post-processing stage. One can avoid this type of post-processing and directly choose the segment with maximum responses, $\arg\max_{k,p} f_k(S_p^I)$, as output. We call this a *simple decision rule*. In experiments we test this rule against more complex post-processing rules.

A main challenge is, once again, the training set size. Since each segment is used as an example, the number of training examples could be large. We mine hard negatives, an approach that has become popular recently [46]. First, regressors are trained only on ground truth segments and putative segments that best overlap the ground truth for each training object. Then, we classify all training segments, find misclassifications, and re-estimate the model parameters with these segments added to the training set. Given a memory budget, we often add only a subset of the misclassified segments and repeat the process multiple

times. Using this procedure, we are able to train on the Caltech-101 and the VOC 2009/2010 datasets in only a few hours.

3.5.3. Learning the Kernel Hyperparameters

Fundamental to equation (3.7) is the form of the kernel function [82]. Existing multiple kernel learning methods that optimize performance measures on the training set suffer from overfitting in many cases [82, 54]. Therefore, we optimize the kernel hyperparameters on the validation set. Since we employ a weighted addition of multiple kernels, it is infeasible to estimate all kernel hyperparameters by means of grid search. Instead, we use gradient descent on an objective function defined on the validation set. To speed-up the process, we apply the algorithm only on a subsample of the data, consisting of segments that best overlap the ground truth. The idea is that kernels need to at least model well the similarity between the clean segments in different classes. Given two exemplars x_i and x_j the additive kernel model is

$$K(x_i, x_j) = \sum_k \beta_k K_k(x_i, x_j; \gamma_k), \tag{3.8}$$

where γ_k is the width of the chi-square kernel. We learn β and γ jointly by directly minimizing the misclassification rate over all images in a (hold-out) validation set:

$$\min_{\beta,\gamma} \sum_{S_p^I \in c_k} \mathbf{1}(f_k(S_p^I) < \max_i f_i(S_p^I)). \tag{3.9}$$

where $f_k(S_p^I) = \sum_{j,k} \alpha_j \beta_k K_k(x_i, x_j; \gamma_k)$ is trained with SVR using the kernel (3.8) on the current β and γ.

To be able to employ gradient-based optimization algorithms, we use the sigmoid function as a continuous approximation to the indicator:

$$\sum_{S_p^I \in c_k} u(f_k(S_p^I) < \max_i f_i(S_p^I)), \tag{3.10}$$

where $u(x) = \frac{1}{1+e^{-\sigma_0 x}}$. A quasi-Newton method is used to find a local optimum for the parameters. Since both the number of kernel parameters and the number of examples are small, this process is fast.

We found that hyperparameters obtained by this procedure are very stable. We learned them on the VOC 2009 train and validation sets and used them throughout all our experiments, both in the VOC 2009 and 2010 (validation and test sets) and for the ETHZ Shape, with consistently good performance.

3.5.4. Connections with Structural SVM

There are interesting connections between our learning approach and the method of Blaschko and Lampert [11], which uses a structural SVM [141] to learn a model for detection. For a bounding box y_i and a ground truth bounding box y, let x_i be the feature vector for y_i and x

the feature vector for y. The structural SVM formulation for sliding window prediction is:

$$\begin{aligned} \min_{w,\xi} \quad & \tfrac{1}{2}\|w\|^2 + C\textstyle\sum_{i=1}^{n}\xi_i \\ \text{s.t.} \quad & \xi_i \geq 0, \forall i \\ & \langle w, \phi(x,y) - \phi(x_i,y_i)\rangle \geq 1 - O(y_i,y) - \xi_i. \end{aligned} \tag{3.11}$$

Structural SVMs have a larger feature space than standard SVMs because the output is kernelized and y appears jointly in the embedding function $\phi(x,y)$. However, the output vector of [11] is 5-dimensional: the class label and the locations of the bounding box. This makes the difference between the input and the joint feature space dimensionality unimportant.

Another difference to [11] is that all possible rectangular regions are considered. This is feasible within a branch-and-bound procedure [88] that can rapidly prune out irrelevant regions of the search space, for the restricted class of features and linear models used in [11]. However, it is difficult to adapt both the structural SVM and the branch-and-bound methodology for the much more powerful nonlinear SVM predictors and image features we want to be able to use. Our task is easier, however, because our use of a compact pool of image segments eliminates the need to process a large number of bounding boxes.

Ignoring these two differences, the structural SVM (3.11) looks superficially similar to our SVR formulation (3.7). It could be seen that if we assume $\langle w, \phi(x,y)\rangle = 1 - \epsilon$, then the last constraint in (3.11) would be the same as the last constraint in (3.7). The difference is clear, however: (3.11) scores the ground truth bounding box and ensures its quality is better than other tentative bounding boxes, with margin determined by the overlap. Meanwhile, (3.7) simply scores all the segments and measures an absolute quality of the segments. We argue that our approach has important advantages. It does not only guarantee the highest rank for the ground truth, but also the correct ranking for all remaining (putative) segments: those with higher overlap will simply have higher scores. For structural SVM, only the *smallest* margin between the best segment and other segments is imposed based on the overlap. Since each segment may have an arbitrarily low score without violating margin constraints, the segment ordering is not preserved inside the non-maximum subset.

3.6. Sequential Segment Post-Processing

3.6.1. Generating Segmentation Results

The challenge of this stage is to form a consistent segmentation and labeling for images containing multiple objects, given a set of plausible, reasonably high ranked segments with initial category labels. The simple decision rule of only using the highest scoring segment cannot handle multiple objects in an image. The non-maximum suppression method that removes all regions overlapping the highest scoring one is standard in bounding box detection, and can be used similarly for segmentation, but we argue that a better approach can be constructed by exploiting the redundancy of class predictions from multiple overlapping segments. Our methodology employs a weighted consolidation of segments and a sequential interpretation strategy, in order to analyze images with multiple objects.

Figure 3.7 shows an example. After classification, the highest-ranked segment was assigned the correct category, `cat`, but this segment also contains background around the object. The next two segments located the cat exactly, but were classified as `dog`. One can see that pre-

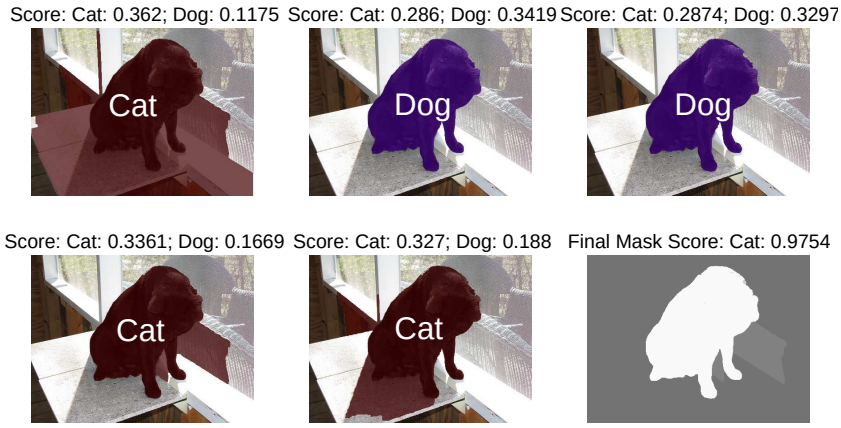


Figure 3.7.: (Best viewed in color) An image of a cat from the VOC2009 dataset. We show the cat/dog scores of the 5 top scoring segments from the image. It is relatively difficult to distinguish if this instance is a cat or a dog, from the foreground/object information only (e.g., top-middle and top-right segments). However, our algorithm takes advantage of multiple slightly different overlapping segments to produce a robust decision, that consistently improves upon the simple decision rule. In the Final Mask, the cat itself has the strongest score (indicated by high intensity values).

dictions for these two segments are not very decisive, since `cat` and `dog` have very similar scores. By taking into account the class predictions of such multiple overlapping segments, it is possible to achieve more robust decisions.

Since the higher-ranked segments should have higher probability of representing full objects, we proceed iteratively. First, we consider the highest-scoring segment as a seed and group segments that intersect it. To decide which segments to group, we compute a segment intersection measure:

$$\mathrm{Int}(S_p, S_q) = \frac{|S_p \bigcap S_q|}{\min(|S_p|, |S_q|)}. \tag{3.12}$$

Under this criterion, parts have 100% intersection with full objects, therefore they are always grouped together. We consider segments with intersection $> \tau_1$ ($\tau_1 = 75\%$ chosen based on the validation sets) as candidates for combination. In the end, a list L_1^I (1 is used as index because this is the first candidate mask in the image) of segments is generated, in which partially overlapping segments are sorted according to their descending scores.

We then generate the scores for each pixel and each class in the image by weighted voting based on the segments in the list

$$g_k(p_j) = \sum_{S_i \in L_1^I} w_i \mathbf{1}(p_j \in S_i) f_k(S_i). \tag{3.13}$$

where S_i represents the i-th ranked segment in the list L_1^I, k is a certain class, p_j is a pixel, $f_k(S_i)$ is the predicted score for S_i on class k. Through this equation, scores on segments are transferred to scores on pixels inside segments. Then a weighted combination is taken, with segments with higher prediction having higher weights. For a pixel, its scores are only counted on the segments that overlap it, as given by the term $\mathbf{1}(p_j \in S_i)$. Therefore, pixels that appear in all segments get higher scores, whereas pixels that only rarely appear get lower scores. Besides, because scores are computed for each class separately, if all overlapping

segments agree on the label, that class is supported strongly. Finally, each pixel is assigned to the class that has the highest score: $g(p_j) = \max_k g_k(p_j)$.

We define the term *mask* as a figure-ground segmentation with each pixel on the foreground classified to some category, in order to differentiate it from *segments*. To separate foreground and background, only pixels with final scores $> \tau_2$ are displayed in the aggregated mask M_1^I ($\tau_2 = 0.55$ is selected, based on validation data). The score of the mask is given by

$$g(M_1^I) = \max_{p_j \in I} g(p_j) \tag{3.14}$$

The last image in fig. 3.7 shows an example of the final mask, where it can be seen that the classification is now correct and the scores are highest in the cat region and much lower in other regions.

The weights w_i in (3.13) are associated with the rank (in the list m) of the segment only, uniformly across different images and classes. These are learned using linear regression on targets that measure the overlap of the generated masks with ground truth, in the validation set.

After we have generated a final mask M_1^I from segments in L_1^I, we remove the segment set L_1^I and the foreground region in M_1^I from the image and consider it consolidated. Then we proceed with the next highest-ranked segment. Based on the same procedure we generate L_2^I and M_2^I, *etc*. Altogether in the VOC dataset usually 6-7 final masks are sufficient. In the end, the final masks are filtered, and only those with mask score $g(M_j^I) \geq \tau_3 (\tau_3 = 0.66$ chosen based on validation data) are retained in the final result. It can be seen that the false positive rate is high, therefore so many stages are needed to reduce variance. With more training data and improved regression accuracy, we can probably remove some of the filtering steps. The post-processing method is detailed as Algorithm 1.

We also implement a simple filter based on the class co-occurrence frequencies in the VOC training set [56]. A co-occurrence frequency matrix is computed, whose ij-th entry counts the number of times two objects of class i and j co-occur in the same image. During testing, we filter object pairs that never co-occur. This only improves performance slightly in our experiments (see Table 3.1). Further discussion on alternative decision rules appears in Section 3.7.1.

3.6.2. Generating Detection and Classification Results

To generate detection results, the method changes slightly. We use overlap (3.1) to replace the intersection measure (3.12) used for grouping segments. This is because when using an intersection measure, small objects are combined within a larger segment containing them. For instance, sometimes we combine two bottles placed next to each other into one large segment enclosing both. This may not affect the segmentation performance measure, but for detection, a single bounding box would enclose both bottles and would count as one false positive and two false negatives. Adapting the criterion from intersection to overlap makes the method work well for detection. Also, we do not use a threshold to determine whether to output a segment as in Algorithm 1. Instead, we simply output all the generated final masks. For classification, in each image we simply find the mask with the highest score and output its label.

Algorithm 1 Postprocessing pipeline for image I. Sequential aggregation of multiple categorized segments.

input Segments $\mathbf{S} = \{S_1^I, \ldots, S_m^I\}$, with predicted scores $f^k(S_i^I)$ for each class k.
output Final masks $\{M_i\}$ on the image I.

Sort the segments descending by maximal score $f(S_i^I) = \max_k f^k(S_i^I)$ on all classes.
n = 1
while S is not empty **do**
  Select $S_n^I = \arg\max_i f(S_i^I)$, the segment with the highest maximal score.
  Find all segments that have at least τ_1 intersection with S_n^I, let them be L_n^I, still sorted by maximal score.
  For each pixel p_j in the image, compute pixel score $g_k(p_j)$ for each class k by

$$g_k(p_j) = \sum_{S_i \in L_n^I} w_i \mathbf{1}(p_j \in S_i) f_k(S_i). \tag{3.15}$$

  for each pixel p_j **do**
    if $\max_k g_k(p_j) < \tau_2$ **then**
      $M_n(p_j) =$ background
      {Classify p_j as background.}
    else
      $M_n(p_j) = \arg\max_k g_k(p_j)$
      {Classify p_j as class k.}
  if $\max_{k,j} g_k(p_j) > \tau_3$ **then**
    Output M_n
    {The score of the mask is given by the highest pixel score in the mask. It must exceed a threshold to be retained in the final semantic segmentation.}
  Delete all segments in L_n^I from $\mathbf{S}$.
  n = n + 1

3.7. Experiments

The experiments are divided in two parts. The first section shows proof-of-concept studies, where various important aspects of the algorithm are tested. In the second section, we show results of our recognition framework (denoted SvrSegm, abbreviated from SVR on SEGMentations) applied to three key tasks in image understanding: image classification, object localization and object segmentation. We also compare with previously reported results.

The segments used in all experiments except those on PASCAL VOC 2010 were generated by CPMC based on the same 5x5 grid of seeds and the same parameters detailed in the original CPMC paper [26]. The experiments on PASCAL VOC 2010 used CPMC with a different set of parameters tuned for producing a larger initial pools of segments. Additionally, these experiments used an expanded set of seeds. Further detail on the PASCAL VOC 2010 segments can be found in the documentation provided with the publicly available CPMC segmentation implementation [25].

The initial pools of segments have, averaged over all images, 95 segments for the ETHZ shape dataset, 64 for Caltech 101, 145 for VOC2009 and 736 (with the new parameters) for PASCAL VOC 2010. One possible way to measure the CPMC performance on a dataset is to compute the maximum IOU-overlap between each ground truth object and any generated segment, then average over all objects. This score also illustrates how difficult low-level segmentation is in each dataset. Our pools of CPMC segments obtain 0.83 on Caltech 101, 0.85 on ETHZ Shapes and 0.66 on PASCAL VOC 2009. With the new CPMC configuration, on PASCAL VOC 2010 we obtain a maximum IOU-overlap of 0.74. Note that the PASCAL VOC datasets are considerably more challenging for low-level object segmentation. More details about these datasets will be given in the next subsections.

3.7.1. Proof-of-Concept Experiments

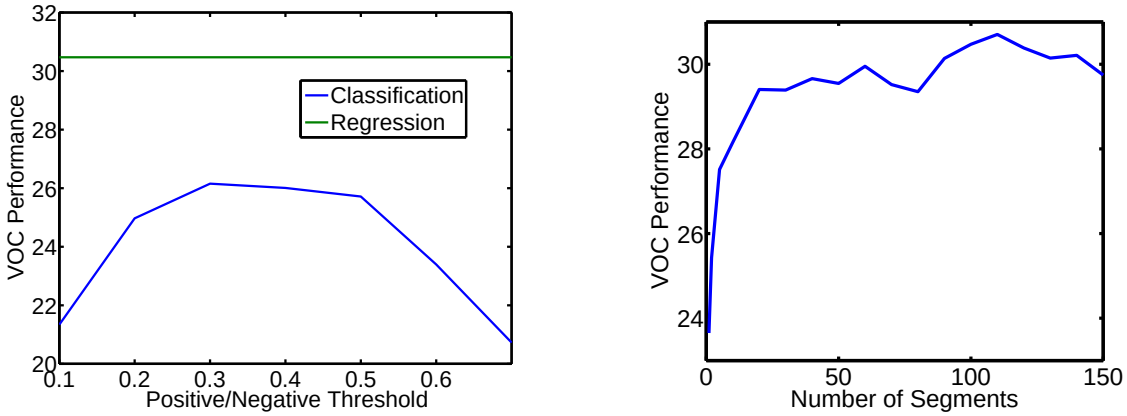
In this subsection we test two concepts presented in the chapter: 1) Regression against overlap and 2) Post-processing. We use the PASCAL VOC 2010 dataset to perform these tests.

The PASCAL VOC 2010 segmentation dataset contains 1928 images (with 4203 objects) for training, which are divided into 964 images (2075 objects) in the `train` set and 964 images (2028 objects) in the `val` set. Objects are selected from 20 classes. A hold-out `test` set of 964 images is used to evaluate the performance of the algorithm. For this data, annotations are not available and one must submit results to an external evaluation server[1]. The performance is measured using per-class overlap, defined as:

$$\text{segmentation accuracy} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}} \tag{3.16}$$

where TP is the number of true positive pixels of the class, FP is the number of false positive pixels and FN is the number of false negative pixels. The TP, FP, and FN values are summed across all the images of the test set. In the end, the 21 per-class overlaps (all the 20 classes plus the background class) are shown, and the mean performance is an average over the 21 individual accuracies. Naturally, this performance measure favors big segments, which may often be more important in understanding the image, although this is perhaps arguable. Our

[1] available at `http://host.robots.ox.ac.uk:8080/`



(a) Comparison of classification and regression approaches. Even the best threshold for classification gives results vastly inferior to regression.

(b) Performance as a function of the number of segments. Performance improves very quickly initially, as more segments are added and reaches its peak for 110 segments. Beyond that value, it deteriorates slightly.

Figure 3.8.: Studies on the VOC2010 segmentation validation set.

FB-overlap measure (3.2) is designed to reflect the evaluation objective in a principled manner, and shows the flexibility of our approach in adapting to different objectives.

In this subsection we perform experiments by training on the `train` set and testing on the `val` set. This is consistent with the recommended usage of the two sets: to test the model and identify parameters. We use the VOC mean performance to evaluate the models.

First, we test our one-vs-all regression scheme against the more commonly used classification approach. We set an acceptance threshold on the overlap so that segments with overlap higher than a threshold are considered positives for the class and the remaining ones are considered negative; we varied this parameter from 0.1 to 0.7. All the other parameters are the same except that we use SVM classification instead of regression. To avoid interference from external factors, post-processing is disabled in this experiment, and only the best segment for each image is reported. The result is shown in fig. 3.8a.

The regression scheme obtained 30.47% as VOC mean score. Among the threshold values tested for classification, the best threshold (0.3) achieved 26.15%. Therefore, the one-against-all regression approach brings at least a 4% performance improvement, and has one less parameter to tune compared to classification (the acceptance threshold).

Another relevant aspect of study is the number of segments required by the algorithm in order to obtain good results. This can also be seen as a test on the performance of the class-independent segment ranking method (Section 3.4). For this study we again disabled post-processing operations and output only the best segment for each image. The results in fig. 3.8b, perhaps surprisingly, show that even by using only a few segments, the results are not much lower than the best ones that we achieved. Moreover, when using more than 110 segments, the accuracy does not saturate but deteriorates slightly. Since the classifier has limited inductive power, it seems that when there are too many low quality segments in both the training and testing sets, spending too much capacity on predicting those well negatively impacts the ability to correctly generalize on good segments. This justifies our need of a multi-stage segment filtering approach.

We also test the importance of various factors in post-processing. Compared with the straightforward approach of selecting the best segment for each image, there are two improvements from post-processing: 1) Improving the quality of the segment; 2) Obtaining multiple segments per image instead of just one. In order to separate these factors, we compare the full post-processing results with strategies that only extract one segment per image.

We show detailed results of this experiment in Table 3.1, where the improvement provided by each step is recorded. From the results, we note that post-processing improves the quality of the segmentation by about 3% (improvements are observed in 17 out of 21 classes) when moving from `Simple` to `1-Seg`. Besides, our approach significantly outperforms non-maximum suppression (`NMS`). However, allowing for multiple segments leads to mixed results: the performance deteriorates in 8 out of 21 classes and only improves in 12. The co-occurrence criterion is not entirely satisfactory either: from the simpler `No co-occur` to `Full`, only 4 classes show significant performance improvement.

3.7.2. Performance Experiments

Image Classification: Caltech-101

We also test the image classification performance of our algorithm in the Caltech-101 benchmark[42]. As in standard approaches, we report results averaged on all the 101 classes, over 3 different random splits. For each class, we use 5, 15 or 30 images for training and up to 15 images for testing, following the common setting in the literature. We train the model using ground truth segmentation masks provided with the dataset. In fig. 3.9, we compare our results against existing approaches. Our scores consistently improve the current state-of-the-art in all training regimes. In particular, our approach outperforms other multiple kernel frameworks such as [54] and the segmentation-based framework of [63].

We have also run some of the proof-of-concept experiments on this dataset, in order to compare our regression scheme with SVC (support vector classification). We also evaluate the impact of post-processing. Since the outputs of our SVR are different from those of SVC, we do not employ the post-processing algorithm in this comparison, but use only the simple decision rule. It turns out that in Caltech-101, the simple decision rule works well. Table 3.2 confirms that regression works significantly better than classification. More sophisticated post-processing does not outperform the simple decision rule in this case, except for the small training regimes (5 training images). Two experiments were pursued further. The first uses only the best segment in our hypothesis pool for both training and testing; the second uses only the ground truth segment for the same purpose. The experiments show that we are very close to saturation: the results generated by training and testing only on our best segment for each image are not significantly better than results based on multiple segments. Arguably, in this dataset, improvements are more likely to emerge from better features and better segments, than the decision rule itself.

Detection: ETHZ Shape Classes

We compare our detection results with the ones reported in [63], a competitive segmentation-based recognition approach. We use the ETH Zurich database[47] which contains 5 shape categories and 255 images. We follow the experimental settings in [47], and use the PASCAL criterion to decide if a detection is correct. The image set is evenly split into training and test-

Class Name	`Simple`	`NMS`	`1-Seg`	`No new segment`	`No co-occur`	`Full`
Mean	30.47	31.84	33.28	33.76	33.91	**34.30**
Background	79.01	80.74	81.60	81.71	**82.03**	**82.03**
Aeroplane	35.65	41.66	**44.47**	42.13	43.80	43.97
Bicycle	16.66	16.03	**16.92**	16.03	16.14	16.29
Bird	30.99	31.22	**34.76**	33.24	32.38	32.55
Boat	29.65	32.21	**34.42**	33.59	33.61	33.81
Bottle	40.72	41.94	40.81	42.26	**43.07**	**43.07**
Bus	44.88	48.25	47.72	47.64	49.55	**49.70**
Car	**56.92**	53.63	55.64	55.58	53.94	56.19
Cat	34.35	36.20	37.10	35.86	**37.26**	36.28
Chair	4.94	**7.35**	4.24	6.26	6.79	6.79
Cow	8.51	8.80	11.57	13.08	**13.48**	13.13
Dining Table	12.53	14.43	19.84	**24.12**	23.56	23.31
Dog	13.94	14.98	16.57	17.43	17.35	**17.52**
Horse	**32.53**	29.03	31.14	29.44	30.30	30.33
Motorbike	42.04	41.36	**47.61**	46.42	45.47	46.80
Person	26.26	30.85	27.67	33.35	**33.73**	33.71
Potted Plant	**20.54**	20.15	18.74	18.70	19.01	19.01
Sheep	30.36	35.62	33.20	36.74	36.31	**38.67**
Sofa	14.90	15.79	15.94	**20.19**	17.47	19.93
Train	35.28	37.20	41.93	41.86	41.94	**42.39**
TV/Monitor	29.25	31.16	**36.94**	33.33	35.00	34.75

Table 3.1.: Study of the effects of post-processing on the VOC2010 validation set. The `Simple` scheme uses no post-processing and outputs only the best segment. `NMS` is the result obtained using non-maximum suppression. `1-Seg` outputs at most 1 best segment from post-processing, but allows to combine multiple segments. `No new segment` allows an arbitrary number of segments, but selects the segment from the original pool that is closest to the post-processing result. In `No co-occur`, the result is not filtered by the frequency matrix of segment co-occurrence. `Full` uses the full post-processing pipeline described in the chapter.

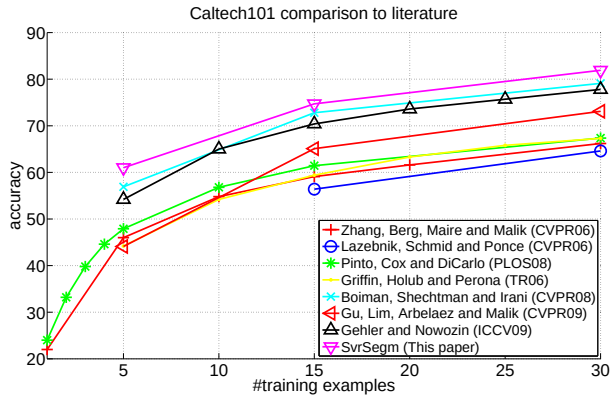


Figure 3.9.: Comparisons on Caltech-101. SvrSegm outperforms the current state of the art for all training regimes.

Method	5 Train	15 Train	30 Train
Classification	58.6	72.6	79.2
Regression	59.6	74.7	82.3
Reg. w/ Post-Processing	60.9	74.7	81.9
Best Segment	62.4	75.8	82.5
Ground Truth Segment	71.7	83.7	89.3

Table 3.2.: Comparisons of different settings of SvrSegm for learning in Caltech-101. Our regression on overlap framework significantly outperforms classifier-based implementations. Post-processing helps somewhat for small training sets. We also show the result produced by using only the best ranked segments and ground truth segments (in both training and testing), to give an idea of the best performance the current recognition framework could obtain by improving the segmentation.

ing sets and performance is averaged over 5 random splits. For training with just bounding box data, we automatically extracted an object mask inside each bounding box and set it as the ground truth segmentation mask. This mask is obtained by first generating multiple segments inside the bounding box, then selecting the one that maximizes a mid-level segment quality score—the output of the predictor in [26], from which we subtract the sum of Euclidean distance of the segment to each edge of the ground truth bounding box, as a penalty for deviation from the frame constraint.

ETHZ results are given in fig. 3.10. Our method outperforms the state of the art by nearly an order of magnitude—at 0.02 FPPI (false positives per image) our detection rate is comparable with the detection rate at 0.2 FPPI in Gu et al. [63]. Comparisons between algorithms at 0.02 FPPI are shown in Table 3.4. We achieve 98.3%, a nearly perfect detection rate for the Swans category, at less than 0.02 FPPI.

We also evaluate the quality of our object segmentations using the ground truth segmentation masks made available by Gu *et al*. [63]. Following [63], we report pixel average precision (AP) on each class. For each, a ROC curve is computed by varying the detection threshold on the mask scores of segments. AP is computed as the area under the curve. Comparisons with [63] in Table 3.3 show improvement in most classes.

Results of SvrSegm for various training conditions are shown in fig. 3.11. We use three variants for the scoring function: overlap with the bounding box (named Bounding Box in the figure); overlap with automatic object mask generated from the bounding box (Automatic Overlap) and overlap with the ground truth object mask (Ground Truth). The algorithm appears to be robust to noise in the overlap measure. We also trained and tested our recognition framework using segments from [4] (denoted OWT-UCM Segments). We observe that this setting produces lower scores than the one obtained using CPMC segments. A possible explanation is that the OWT-UCM segments usually do not correspond to full objects but to parts and other image regions. This type of input does not appear to be effective in conjunction with our recognition framework.

Segmentation and Labeling: VOC 2009 and 2010

The SvrSegm algorithm was used in BONN_SVM-SEGM entry for the PASCAL VOC 2009 Challenge and the BONN_SVR_SEGM entry for the PASCAL VOC 2010 Challenge. The system was declared a winner in the VOC 2009 challenge and a joint winner of the 2010 challenge.

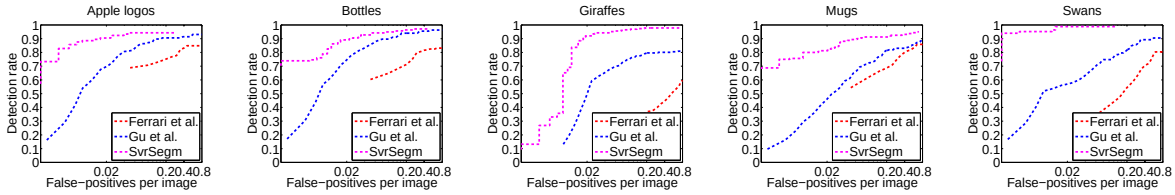


Figure 3.10.: Comparisons on ETHZ-Shape classes. SvrSegm is trained using only bounding box data.

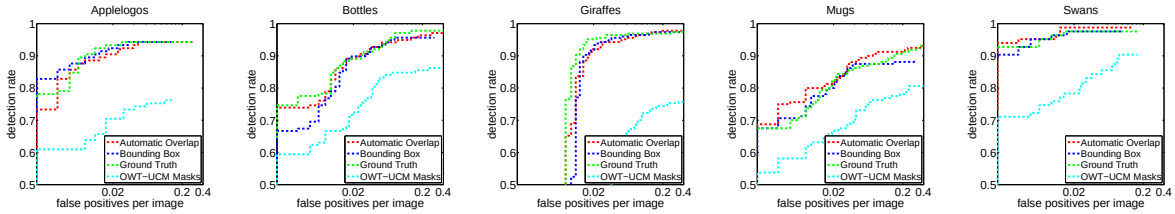


Figure 3.11.: Comparisons on ETHZ-Shape classes for different training conditions. SvrSegm is trained to predict overlap with object masks generated from the bounding box (Automatic Overlap), overlap with the bounding box (Bounding Box) and ground truth object masks (Ground Truth). We also both trained and tested with segments from Arbelaez et al.[4] (OWT-UCM Masks).

Categories	Gu et al.	SvrSegm
Applelogos	77.2 ± 11.1	89.0 ± 1.9
Bottles	90.6 ± 1.5	90.0 ± 2.1
Giraffes	74.2 ± 2.5	75.4 ± 1.9
Mugs	76.0 ± 4.4	77.7 ± 5.9
Swans	60.6 ± 1.3	80.5 ± 2.8
Average	75.7 ± 3.2	82.5 ± 1.2

Table 3.3.: Segmentation results for ETHZ-Shape. Performance (%) is measured as pixel-wise mean AP over 5 trials, following [63].

Categories	Ferrari et al.	Gu et al.	SvrSegm
Applelogos	68.83	69.75	90.48
Bottles	60.32	74.59	89.13
Mugs	46.06	54.33	81.25
Giraffes	23.75	49.63	92.07
Swans	31.60	56.98	98.31
Average	47.76	59.40	90.25

Table 3.4.: Detection rate at 0.02 FPPI in ETHZ-Shape. SvrSegm noticeably improves on the state-of-the art in this regime.

This section describes the results obtained in these challenges, and our subsequent efforts on the VOC 2010 dataset, after the challenge, which results in the best performance reported so far for this dataset on the test set: 43.8% accuracy.

The 2009 segmentation challenge provides 1,499 images (containing 3211 objects) in the `trainval` dataset and 750 images in the hold-out `test` set to evaluate the performance of submitted algorithms. Additionally there are 5,555 images (with 14,007 objects) where only bounding box annotations are available. We did not use images with bounding box annotations at the time of the challenge, where our entry was declared as winner with an accuracy of 36.3% (in evaluating different methodologies, notice that some of the participants used these additional images to train their system [56]). The results of the challenge are reproduced in Table 3.5. Some systems from the detection challenge have automatic entries in the segmentation challenge, since a trivial segment from the bounding box can be generated. However this often gives relatively uncompetitive results that we omit in the table.

After the challenge, we have also exploited bounding box annotations crudely (only one segment which best overlaps the bounding box is used, with overlap value always set to 0.8) to produce the slightly improved 37.24% accuracy reported in [95]. This result is not included in this chapter because the methodology is slightly different, but see our work in [95] for details.

As described in section 3.7.1, in the 2010 segmentation challenge, the `trainval` set is augmented to 1,928 images (with 4,203 objects) and the `test` set is augmented to 964 images. An additional 8,175 images (containing 19,171 objects) have only bounding box annotations. This approach was one of the joint winners with an accuracy of 39.7%. The version we submitted to the challenges was trained only based on segmentation annotation and without taking advantage of the information in the additional images that contain only bounding box annotations. After the challenge we included those additional images in the training set. For each ground truth object, we selected the 10 segments whose bounding-box had the best IOU-overlap with the object bounding box, and set those overlap values as desired outputs. With this additional training data, we obtain a further 4% performance improvement on VOC 2010, resulting in 43.8%. To our knowledge this is the best result reported on this dataset so far. Table 3.6 provides details.

Fig. 3.12 illustrates some successfully segmented images from the VOC test set. It can be seen that our method handles background clutter, partially occluded objects, objects with low contrast with the background, as well as multiple objects in the same image. The first two images shown in the last row have particularly low contrast—the sheep in the first image or the black suit of the child in the second one are almost the same color as the background. Our approach nevertheless succeeds in identifying the correct spatial support of those objects and also predicts their category correctly.

However, despite our moderate success, the performance on the VOC dataset remains at around 44%, which means there is still substantial room for improvement. In order to gain intuition on directions for future development, we also show images where the method fails. Fig. 3.13 shows images that illustrate various types of failure. We partition the errors into 4 groups. In group 1, errors come from the inability to correctly select segments. Usually, the segments selected by the algorithm are to some degree intuitive. For instance, in the last image where we classified a segment as boat, a background segment shaped as a boat was selected and wrongly labeled as boat. In turn, the aircraft is also quite hard to detect since it is small and almost entirely occluded.

Name	SvrSegm	BROOKES MSRC	CVC	LEAR	MPI	NEC UIUC	UC3M	UCI	UCLA	UoC TTI
Mean	**36.3**	24.8	34.5	25.7	15.0	29.7	14.5	24.7	13.8	29.0
background	**83.9**	79.6	80.2	79.1	70.9	81.8	69.8	80.7	51.2	78.9
aeroplane	64.3	48.3	**67.1**	44.6	16.4	41.9	20.8	38.3	13.9	35.3
bicycle	21.8	6.7	26.6	15.5	8.7	23.1	9.7	**30.9**	7.0	22.5
bird	21.7	19.1	**30.3**	20.5	8.6	22.4	6.3	3.4	3.9	19.1
boat	**32.0**	10.0	31.6	13.3	8.3	22.0	4.3	4.4	6.4	23.5
bottle	**40.2**	16.6	30.0	28.8	20.8	27.8	7.9	31.7	8.1	36.2
bus	**57.3**	32.7	44.5	29.3	21.6	43.2	19.7	45.5	14.4	41.2
car	49.4	38.1	41.6	35.8	14.4	**51.8**	21.8	47.3	24.3	50.1
cat	**38.8**	25.3	25.2	25.4	10.5	25.9	7.7	10.4	12.1	11.7
chair	5.2	5.5	5.9	4.4	0.0	4.5	3.8	4.8	6.4	**8.9**
cow	**28.5**	9.4	27.8	20.3	14.2	18.5	7.5	14.3	10.3	**28.5**
diningtable	22.0	**25.1**	11.0	1.3	17.2	18.0	9.6	8.8	14.5	1.4
dog	19.6	13.3	23.1	16.4	7.3	**23.5**	9.5	6.1	6.7	5.9
horse	33.6	12.3	**40.5**	28.2	9.3	26.9	12.3	21.5	9.7	24.0
motorbike	45.5	35.5	**53.2**	30.0	20.3	36.6	16.5	25.0	23.6	35.3
person	33.6	20.7	32.0	24.5	18.2	34.8	16.4	**38.9**	20.0	33.4
pottedplant	27.3	13.4	22.2	12.2	6.9	8.8	1.5	14.8	2.3	**35.1**
sheep	**40.4**	17.1	37.4	31.5	14.1	28.3	14.2	14.4	12.6	27.7
sofa	18.1	18.4	**23.6**	18.3	0.0	14.0	11.0	3.0	12.3	14.2
train	33.6	37.5	**40.3**	28.8	13.2	35.5	14.1	29.1	17.0	34.1
tv/monitor	**46.1**	36.4	30.2	31.9	13.2	34.7	20.3	45.5	13.2	41.8

Table 3.5.: VOC 2009 segmentation results on the test set, for various research teams participating in the challenge. SvrSegm is the method presented in this chapter.

Name	SvrSegm WITH DET	SvrSegm W/O DET	BROOKES	CVC	STANFORD	UC3M	UOCTTI
Mean	**43.8**	39.7	30.3	40.1	29.1	27.8	31.8
background	**84.6**	84.2	70.1	81.1	80.0	73.4	80.0
aeroplane	**59.0**	52.5	31.0	58.3	38.8	45.9	36.7
bicycle	**28.0**	27.4	18.8	23.1	21.5	12.3	23.9
bird	**44.0**	32.3	19.5	39.0	13.6	14.5	20.9
boat	35.5	34.5	23.9	**37.8**	9.2	22.3	18.8
bottle	**50.9**	47.4	31.3	36.4	31.1	9.3	41.0
bus	**68.0**	60.6	53.5	63.2	51.8	46.8	62.7
car	53.5	54.8	45.3	**62.4**	44.4	38.3	49.0
cat	**45.6**	42.6	24.4	31.9	25.7	41.7	21.5
chair	**15.3**	9.0	8.2	9.1	6.7	0.0	8.3
cow	**40.0**	32.9	31.0	36.8	26.0	35.9	21.1
diningtable	**28.9**	25.2	16.4	24.6	12.5	20.7	7.0
dog	33.5	27.1	15.8	29.4	12.8	**34.1**	16.4
horse	**53.1**	32.4	27.3	37.5	31.0	34.8	28.2
motorbike	53.2	47.1	48.1	**60.6**	41.9	33.5	42.5
person	37.6	38.3	31.1	**44.9**	44.4	24.6	40.5
pottedplant	35.8	**36.8**	31.0	30.1	5.7	4.7	19.6
sheep	48.5	**50.3**	27.5	36.8	37.5	25.6	33.6
sofa	**23.6**	21.9	19.8	19.4	10.0	13.0	13.3
train	39.3	35.2	34.8	**44.1**	33.2	26.8	34.1
tv/monitor	42.1	40.9	26.4	35.9	32.3	26.1	**48.5**

Table 3.6.: VOC 2010 segmentation results on the test set. For our method, SvrSegm, models trained both *with* and *without* additional bounding box data and images from the training set for object detection are shown (WITH DET and W/O DET, respectively).

Figure 3.12.: Successful semantic segmentations produced by our method on the VOC test set. Notice that the object boundaries are relatively accurate and that our method can handle partial views and background clutter.

In the second failure group, the algorithm does not successfully handle multiple interacting objects, such as men on motorbike. These types of images are difficult to segment purely bottom-up because of the complicated patterns of mutual occlusion between objects. It might also be, in part, a problem of the current post-processing method, whose sequential nature (fix a mask before considering the next one) does not always allow for a joint analysis of multiple segments and categories. We have recently developed alternative formulations to address some of these issues [27, 71].

The third failure group illustrates errors in classification. Currently, confusions mostly arise between a few relatively similar category pairs: cow—horse, dog—cat, dog—horse, dog—sheep, other tables (which are labeled as background in the challenge)—dining table, sofa—chair, and TV/Monitor—other similar shaped objects (e.g., windows, glasses on doors). Otherwise, if a segment is correctly recovered, it is usually correctly classified. Considering the relatively small training set, we believe that such errors are not very problematic in the long run, as more training data becomes available.

The fourth failure group shows that it is sometimes difficult for the method to determine the proper spatial extent of objects. This can happen when parts of objects are recovered (the table and the bottle in the group), an overly large segment contains the object (the sofa and the bird in the group) or reflections occur (the boat in the group).

It is also worth mentioning that because normal tables are not classified as dining tables in the VOC dataset, the trained dining table classifier mainly looks for dishes, plates, glasses and other stuff on the table, instead of the table itself. This annotation may just be too fine-grained considering the dataset size and distribution. At the same time it is to some degree ambiguous as in principle almost any table can be used as a dining table.

3.8. Conclusion

We have described a semantic image interpretation framework based on a novel front end algorithm, CPMC [26], that generates multiple figure-ground segmentations, followed by sequential object labeling. Unlike previous methods that rely on classification, we frame recognition as a regression problem of estimating the spatial overlap of generated segments with the target object of the desired category. Instead of selecting only one segment, we produce a ranking in the space of all putative segments based on spatial overlap. This makes it possible to better exploit segments that partially overlap the ground truth in order to consolidate recognition. We demonstrate state-of-the-art results in image classification, object detection and semantic segmentation in Caltech-101, ETHZ-Shapes and PASCAL VOC 2009 and VOC 2010. Our approach is dominantly bottom-up: object class knowledge is used only after plausible object segmentations have been obtained. In the long run, a closer integration of top-down information could improve performance. In this work, however, we make a case that bottom-up modules that extract object-level segments beyond superpixels can achieve good performance. They are a plausible front-end for both segmentation and recognition tasks.

Figure 3.13.: Failure modes of our semantic segmentation on the VOC testset, split into four groups. See text for discussion.

Chapter 4.

Conclusions and Future Directions

Vision allows humans to build extremely rich internal representations of physical scenes. Many important tasks require reasoning about the elements of a scene and this often requires these elements to be linked to memories. For example, understanding that a circular object in the image is an 'apple' affects the decision of eating it. Visual recognition is the ability to 'link' elements of a scene pictured in an image with memories of related elements seen previously, often object categories, such as 'car', and 'person'.

Reproducing visual recognition in computers has been a subject of much research. While there has been progress, recognition is still challenging. Even small problems comprising a few dozen categories are notoriously difficult for current technology [39]. Visual recognition can be roughly decomposed into three main subproblems: how to explore the image and select regions to inspect, how to explore memory and select elements to compare with, and how to evaluate the merits of a match between a memory element and an image region.

The main focus of this thesis was the first problem: the development of algorithms for structuring image exploration to facilitate recognition and localization of objects in images. We approached this problem using ideas based on segmentation. Differently from sliding window approaches, which sample densely among the set of rectangles possessing a predetermined aspect-ratio, our techniques select regions having desired low and mid-level statistics learned from category-independent training data. For example, the appearance of objects usually contrasts with their background and this creates intensity gradients along their boundaries in an image. Such properties can be detected and exploited, and their use leads to a much reduced search space, independent of the number of categories to be recognized.

We introduced a recognition model that labels multiple objects sequentially and independently, and uses individual object segmentation proposals as input. We compute these proposals using the Constrained Parametric Min-Cuts algorithm (CPMC), introduced in chapter 2. CPMC generates figure-ground segmentation proposals with desired statistics, learned from annotated ground-truth regions in training images. Chapter 3 explains how we approach recognition with segments. We use support vector regressors operating on multiple non-linear kernels, which would be too computationally expensive to work with without the focused space of object hypotheses that CPMC produces.

An underlying assumption in this thesis is that the initial segmentation is not required to be perfect, but it should be sufficiently precise to allow for accurate recognition at some helpful level. For example, a segment covering the upper-body of a person may be enough for recognition of category 'person' and its identity 'Clyde', but not for determining full-body pose. Once a match is established with a visual pattern in memory, it should then be possible to retrieve associated knowledge corresponding to the expected region shape, which can aid completing and perfecting the segmentation. This new segmentation will then allow for finer recognition.

We evaluated the quality of our segmentations in a large number of datasets (almost all we could find that had segmentation annotation), and obtained better results than those of other top bottom-up methods in the literature. The quality of our segmentations is also validated by state-of-the-art results obtained in recognition tasks, namely in semantic segmentation problems.

In the remaining of this chapter we will suggest directions for future research.

4.1. Future Directions

Despite considerable progress, performance on tasks such as semantic segmentation on the VOC Pascal dataset is still low in absolute terms. This seems to be chiefly caused by an insufficient ability to rank (segment, label) tuples. For example on the training set of the VOC segmentation dataset, our models can obtain 70% VOC score, and the layout quality is good, but performance drops to slightly above 40% when those models are tested on the test set. Generalization, in the form of incorrect top candidate ranking, is unsatisfying. The main problem seems to be that our recognition models do not capture robustly the patterns that define the object categories, as well as the fine distinctions between better and worse object delineations. Additionally, we did not address important topics such as segmentation refinement using knowledge about object shape, as well as fine-grained recognition - our experiments were limited to recognition problems with up to 100 categories. We will now list the research directions we find most important for future work:

- **Large-scale visual memories and efficient learning**
  Future visual systems will handle millions of visual concepts, both abstract categories and particular object instances, such as particular faces, body poses, etc. Simplifying image exploration by reducing it to a small bag of segments, as pursued in this thesis, is helpful, but there is another bottleneck: memory exploration. Memory will need to be organized, perhaps into a hierarchy, so that search can be performed efficiently. Learning should be performed in an adaptive online fashion, as new data arrives, possibly using stochastic gradient methods. Such techniques are already becoming necessary even for static, immutable, datasets as high dimensional features seem to help performance [125]. One approach that incorporates these ideas has been recently proposed by Lai *et al*. [87].

  Annotating large amounts of data will be necessary, and may be partially facilitated by the usage of video, where information can be propagated between frames (recent interesting ideas in this area include [22, 90]).

- **Features**
  Defining segment-specific features is a powerful concept, as it allows segmentation to be modeled as a discriminative learning problem of automatically determining the patterns that differentiate ground truth segmentations from spurious mis-segmentations, for objects from each category. One issue that has received little attention in the literature is how to make a segment *affect* (leave its imprint on) the features. In this thesis we tried clearing local features and gradients centered outside the segment, and in some cases we also deformed the feature extraction coordinate frame. Possible ideas include defining the scale of local feature extraction based on the proportions of the segment and

defining the cells of a spatial pyramid [89] based on segment decomposition instead of on a regular grid.

Segmentation may also simplify the use of features having some degree of intrinsic, matching-free, geometric invariance, for example by using ideas from the literature on affine-invariant local feature detectors [107, 76, 144].

- **Determining correspondences**
  Objects are almost never fully visible, due to self-occlusion and/or occlusion by other objects. This makes it difficult to define what is a *complete* object segmentation in absolute terms. For example, sometimes a complete segmentation of a person in some images corresponds to its upper-body, due to occlusion by a desk, or due to zoom. In other cases, when the person is not occluded, the desired segment covers the full body. Sometimes CPMC computes both half-body and full-body segments for non-occluded people and in such cases it seems hard to decide which segment should be selected without reasoning jointly about pairs of conflicting segments. In order to resolve such issues elegantly the part decomposition of the objects could be modeled. Matching object parts is useful in itself, as it provides richer outputs, but it would also allow to understand 'what is missing', and what is 'extra' in a segment, to guide top-down segmentation. Correspondences, as opposed to holistic object descriptions, may also help achieve better recognition, and much work has been devoted to this [8, 44], but not yet with segmentation on realistic images such as those from the PASCAL VOC dataset.

- **Top-down processes**
  Most existing top-down segmentation techniques assume that the label of the object is known, or that there is a weak spatial initialization in the form of a bounding box enclosing the object [93, 159, 104]. Object segmentations obtained bottom-up are likely to be closer to the desired segmentation than a bounding box and may therefore provide a better initialization. There are however still many remaining challenges, including how to achieve registration between the image and an object model in order to be able to understand "what is wrong" with the current segmentation.

Appendix A.

Energy Minimization with Parametric Max-Flow

APPENDIX

A.1. Introduction

Many aspects of low and high-level vision are naturally modeled as pairwise relationships between variables. In stereo problems, depth varies smoothly in most areas. In scene understanding, a person is unlikely to be taller than a nearby building. Structured-output models are a type of models having dependencies between output variables so that they follow such statistics. A popular type is the *markov random field*, which defines a probability distribution over interdependent input and output variables, which usually factorize over a product of functions defined on cliques of the dependency graph.

One of the main challenges of structured-output models is inference (*e.g.* to find the *maximum a posteriori* solution), which is often intractable. A notable exception is the markov random field with discrete binary outputs and pairwise clique functions obeying certain restricted properties, like submodularity. Inference in these models can be performed exactly in polynomial time using st-mincut/max-flow solvers, as first shown by Greig *et al*. [60]. The precise types of pairwise terms that permit exact inference were later characterized by Kolmogorov and Zabih [81]. Importantly, an efficient max-flow solver for vision problems [20] was developed, which made it possible to handle very large problems (segmentation of an image with 1000x1000 pixels leads to a problem with one million variables).

Inference in markov random fields is also known as energy minimization, owing to the origin of these models in the statistical physics community [73]. We will use both names interchangeably.

Moreover, many approaches in vision focus on obtaining a single solution, either a global optimum or a strong local minimum. There is a particularly efficient algorithm, called parametric max-flow, that jointly solves a family of energy minimization problems that differ by a restricted type of changes in the unary term, that are parameterized by a single number. It retrieves all unique discrete solutions by determining the breakpoints of a piecewise linear cost function, obtained by varying the value of the parameter. Each such breakpoint corresponds to a unique discrete global minimum, conditioned on the value of the parameter.

Parametric max-flow has proved useful in a few recent computer vision methods [80, 66, 152, 97], mainly due to its low polynomial worst-case time complexity, and was an important technique in chap. 2 of this thesis.

We will first define formally the energy problems we are interested in, and how they can be represented as weighted directed graphs, for inference to be performed using max-flow tech-

niques. We will then describe in some detail the first parametric max-flow solver, introduced by Gallo *et al*. [53].

A.2. Parametric Sets of Graph-representable Energy Problems

Consider a set X of binary variables $\{x_1, ..., x_k\}, x_i \in \{0, 1\}$, associated with nodes $\mathcal{V}$ of an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. We will review minimization of a popular type of energy functions in computer vision [80], of the form:

$$E^{\lambda}(X) = \sum_{u \in \mathcal{V}} (a_u + \lambda b_u) \cdot x_u + \sum_{(u,v) \in \mathcal{E}} V_{uv}(x_u, x_v) \tag{1}$$

with $\lambda \in \mathbb{R}$, V_{uv} a submodular function (i.e. $V_{uv}(0,0) + V_{uv}(1,1) \geq V_{uv}(0,1) + V_{uv}(1,0)$. Assume further that all parameters b_u are either non-positive or non-negative.

These energy functions are interesting mainly because their solutions correspond to binary minimum cuts on appropriately defined graphs [81]. Finding minimum cuts is equivalent to finding maximum flows [48]. Hence, given a single λ value, $min_X E^{\lambda}(X)$ can be computed with efficient max-flow algorithms, using techniques such as push-relabel [55], augmenting paths [48, 20] or pseudoflow [67].

Additionally, it is possible to retrieve the solutions for all possible values of λ in the asymptotic complexity of retrieving a single solution using either the Gallo-Grioridis-Tarjan (GGT) algorithm [53], a push-relabel technique, or a version of the pseudoflow algorithm [67].

A.2.1. Graph Construction for Inference with Max-flow Techniques

Energy 1 can be efficiently minimized using max-flow techniques. First the energy must be mapped into a special directed weighted graph, known traditionally as a *network* [55], $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$, containing one node for each variable in X plus two new nodes $\mathcal{V}' = \mathcal{V} \cup \{v_s, v_t\}$, and a nonnegative *capacity* function $c(v, w)$ for each edge (v, w) in $\mathcal{E}'$. Let the number of nodes be n and the number of edges m. Nodes s and t are named respectively the *source* and the *sink*.

Once the capacities (the directed edge weights) are assigned in a proper way, the minimum cut s for the constructed network can be mapped to the solution of problem 1. We define that variables from X associated with nodes in a graph partition containing the source assume value 1 in the original problem, the others assume value 0.

The edges of G' need to be created in a certain way, for minimum cuts of G' to represent solutions of 1. Each edge in $\mathcal{E}$ is replaced by two directed edges in $\mathcal{E}'$, one in each direction, with $c(i, j) = V_{ij}$. These edges are usually known a *n-links* [18]. Two sets of additional edges, both called *t-links* [18] are created. One set connects the source and the nodes in $\mathcal{V}'$. The other set connects nodes in $\mathcal{V}'$ to the sink. Each node in $\mathcal{V}'$ gets one t-link, to either the source or the sink. If $(a_u + \lambda b_u) \geq 0$ then node v_u gets a t-link from the source, else it gets a link to the sink. In any case the capacity is $(a_u + \lambda b_u)$. The graph construction for a simple example problem with 4 variables is given in fig. A.1.

Proof that the minimum cut on this graph indeed provides the solution to problem 1 can be consulted, for example, in the paper by Boykov and Jolly [19]. Note that there are multiple ways to set up the edge weights that result in the same solution [81, 19, 79]. For example adding a constant to both t-links of a node does not affect the solution.

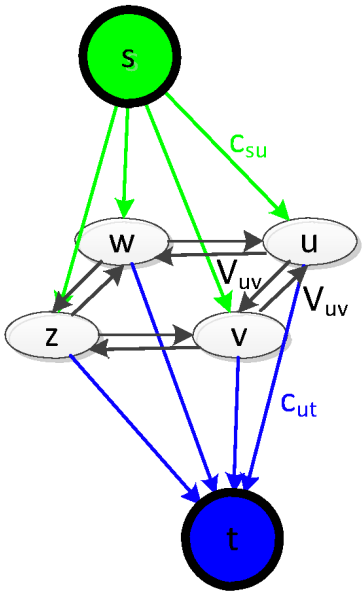


Figure A.1.: Graph construction for max-flow based inference on a problem with four binary variables: u, v, x and z. Two additional nodes are created, the source (s) and sink (t). If $(a_u + \lambda b_u) \geq 0$ then $c_{ut} = 0$ and $c_{su} = (a_u + \lambda b_u)$. Else $c_{ut} = (a_u + \lambda b_u)$ and $c_{su} = 0$.

A.3. Parametric Max-Flow

Given a fixed value of λ one can solve problem 1 in the complexity of a max-flow problem, $O(nm \log(n^2/m))$ with the push-relabel algorithm [55], for a network with n nodes and m edges. To solve the problem for various values of λ one could just repeat the process multiple times. However, the family of solutions has properties that allow for more efficient search procedures.

When the capacities from the source are non-decreasing and the edge weights to the sink are non-increasing functions of λ the problem is said to be monotonic. This is the case in the network corresponding to problem 1, as parameters b_u are assumed to be either non-positive or non-negative. Monotonicity is important because it implies that the different minimum cuts become nested: as λ increases, the source component of the cut grows. The values of λ that induce change in the two minimum cut partitions (node swaps from sink to source or vice-versa) are called breakpoints, and one consequence of nestedness is that there are at most $n - 1$ breakpoints (in the worst case where every breakpoint corresponds to adding a single node to the source side). Information about the non-monotonic case can be found in [80].

Nestedness allows for efficient parametric max-flow using the GGT algorithm [53]. GGT is an adaptation the push-relabel max-flow algorithm [55] that is able to compute minimum cuts for a $O(n)$ sequence of λ values in the worst-case time complexity of a single max-flow problem. An extension of the basic technique retrieves all unique minimum cuts induced by varying λ values, which subsumes the previous problem. It does so in the same worst-case complexity, although with larger constant factors.

We will review these algorithms in this section. We will start by defining additional flow terminology and the basic machinery: the push-relabel algorithm. Afterwards we will discuss

how push-relabel can be adapted to handle a sequence of λ values. The section will conclude with the presentation of the most general algorithm, which builds upon the previous two and is able to compute minimum cuts for all values of λ. We will follow the original exposition [53].

A.3.1. Network Flow Preliminaries

A *flow* f on a generic network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with source node s and sink node t is a real-valued function on pairs of nodes satisfying three constraints: bounded capacity $f(v, w) \leq c(v, w)$, antisymmetry $f(v, w) = -f(w, v)$, for $(v, w) \in E'$, and conservation $\sum_{v \in V} f(u, v) = 0$ for $v \in V' - s, t$. We define $c(v, w) = 0$ if $(v, w) \ni E'$, so the capacity functions extends to any node pair. The value of flow f is given by $\sum_{v \in V} f(v, t)$. The *capacity* of a pair of disjoint nodes Y, Z is $c(Y, Z) = \sum_{v \in Y, w \in Z} c(v, w)$. An important concept referred often in this thesis is the *cut* $(Y, \overline{Y})$ which divides the network into two partitions $(Y \cup \overline{Y} = V', Y \cap \overline{Y} = \emptyset)$ such that $s \in Y$ and $t \in \overline{Y}$. A *minimum cut* is a cut of minimum capacity. The *max-flow min-cut theorem* of Ford and Fulkerson [48] states that the maximum flow equals the minimum cut.

To discuss the push-relabel algorithm, two additional concepts need to be introduced: *preflow* and *valid labelings*. A *preflow* f on G is a real-valued function on node pairs similar to a flow, except it relaxes the conservation constraint: $\sum_{u \in V} f(u, v) \geq 0$ for all $v \in V - s$. Another important notion is that of a *residual edge* for f, which is any node pair (v, w) such that $f(v, w) < c(v, w)$. The difference between capacity and flow in an edge is named *residual capacity* and a path of residual edges is a *residual path*. Any edge that is not a residual edge is said to be *saturated*. Finally, a *valid labeling* d for a preflow f is a function from the nodes to the positive integers, such that $d(t) = 0$, $d(s) = n$, and $d(v) \leq d(w) + 1$ for every residual edge (v, w) .

The GGT parametric max-flow algorithm uses the push-relabel max-flow algorithm [55] as a subroutine. We will now review push-relabel briefly.

A.3.2. Max-flow Using Push-relabel

The push-relabel algorithm computes the maximum flow from s to t through the network. It uses the labeling function $d(v)$, which can be intuitively interpreted as a "height" function in a physical network. Gravity makes liquid flow from higher nodes to lower nodes. The role of the algorithm is then to adjust the heights of the nodes in the network so that, first, as many edges as possible become saturated, and then, any excess liquid in the network (liquid that did not reach t) returns to the source.

The algorithm maintains a preflow f, initially equal to the edge capacities on edges leaving s and zero on edges not incident to s. It improves f by pushing flow excess towards the sink along edges estimated (by using d) to be on shortest residual paths. The value of f gradually becomes larger, and f eventually becomes a flow of maximum value. As a distance estimate, the algorithm uses a valid labeling d, initially defined by $d(s) = n$, $d(v) = 0$ for $v \neq s$. This labeling increases as flow excess is moved among nodes; such movement causes residual edges to change. After algorithm termination and a maximum flow has been obtained, it is possible to find the minimum cut by breadth-first search from the source to the sink. Details can be found in [55].

The complexity of the push-relabel algorithm is dominated by the number of push operations required which is bounded by $O(n^2 m)$. It can be improved to $O(nm\dot{l}og(n^2/m))$ using a

dynamic tree data structure [135].

A.3.3. Parametric Max-flow as a Push-relabel Extension

Unlike regular networks, parametric networks have edge capacities that are functions of a real-valued parameter λ. Let $c_\lambda(v, w)$ be the capacity function in parametric networks. We are interested in a form of $c_\lambda(v, w)$ that is a nondecreasing function of λ on edges from the source, nonincreasing on edges to the sink, and constant on the remaining edges. Note that changing the value of λ can alter the minimum cut of the network.

One configuration of the parametric max-flow problem is concerned with computing minimum cuts for each value of an increasing sequence of parameters. Gallo *et al*. [53] proposed one algorithm for this problem, which assumes that the successive values are given *online*. The key observation exploited by his algorithm, named *parametric push-relabel algorithm* [53], is that increasing the capacity of the edges from the source keeps the labeling d valid. It is therefore feasible to "warm-start" push-relabel for a new parameter λ with the solution of the max-flow problem for the previous parameter.

The algorithm is as follows. All flows and labels are initialized to zero except the source label which is set to n. The algorithm then executes three steps for each parameter λ_i [53]:
Step 1. (Update preflow.) For $(v, t) \in E$, replace $f(v, t)$ by $\min c_{\lambda_i}(v, t), f(v, t)$. For $(s, v) \in E$ with $d(v) < n$, replace $f(s, v)$ by $\max c_{\lambda_i}(s, v), f(s, v)$.
Step 2. (Find maximum flow.) Apply the push-relabel algorithm to the network with the edge capacities corresponding to λ_i, beginning with the current f and d. Let f and d be the resulting flow and final valid labeling.
Step 3. (Find minimum cut.) Redefine $d(v) = \min d_f(v, s) + n, d_f(v, t)$ for each $v \in V$. The cut $(Y_i, \overline{Y}_i)$ is then given by $Y_i = v | d(v) \geq n$.

The outputs of the algorithm are a maximum flow f_i and a minimum cut $(Y_i, \overline{Y_i})$ for each value λ_i of the parameter. As in the push-relabel algorithm, the time complexity is $O(nm\dot{l}og(n^2/m))$ [53] if a dynamic tree data structure [135] is used.

To compute minimum cuts for a decreasing sequence of values of λ instead, it is enough to apply the same algorithm on the reversed network G^R. This network is obtained by reversing the direction of all edges and exchanging the source with the sink.

A.3.4. Retrieving All Breakpoints

Some problems require obtaining all breakpoints, not just minimum cuts for a list of λ values. For obtaining all breakpoints, additional computation needs to be performed. The values of λ where breakpoints occur are a priori unknown and need to be searched over, together with their associated minimum cuts.

Let us assume, without loss of generality, that the capacities $c_\lambda(s, v)$ and $c_\lambda(v, t)$ are given in the form $c_\lambda(s, v) = a_0(v) + \lambda a_1(v)$ and $c_\lambda(s, v) = b_0(v) - \lambda b_1(v)$, with arbitrary coefficients a_0, b_0 and nonnegative coefficients a_1, b_1. A minimum cut (Y, Y_0) at $\lambda = \lambda_0$ gives an equation for a line that contributes a line segment to the function $K(\lambda)$ at $\lambda = \lambda_0$. Note that edge capacities can be negative, because there is a simple transformation that makes them positive without affecting minimum cuts [115]. Additionally, for a given node v, suppose we add a constant

$\delta(v)$ to $c(s,v)$ and $c(v,t)$. The capacity of the minimum cut does not change since the capacity of every cut is increased by $\delta(v)$.

The GGT algorithm for retrieving all breakpoints requires initial bounds on the smallest and largest breakpoints. A lower bound on the smallest breakpoint can be obtained by a value λ_1 small enough so that for each node v such that $c(s,v)$ or $c(v,t)$ have nonconstant capacity, $c_{\lambda_1}(s,v) + \sum_{u \in V-s,t} c(u,v) < c_{\lambda_1}(v,t)$. This translates into the following condition

$$\lambda_1 = min_{v \in V-s,t} \frac{b_0(v) - a_0(v) - \sum_{u \in V-s, tc(u,v)}}{a_1(v) + b_1(v)} - 1 \tag{2}$$

Similarly, a value λ_2 that bounds the largest breakpoint from above can be found by

$$\lambda_2 = max_{v \in V-s,t} \frac{b_0(v) - a_0(v) + \sum_{w \in V-s, tc(v,w)}}{a_1(v) + b_1(v)} - 1 \tag{3}$$

In both cases v is such that $a_1(v) + b_1(v) > 0$.

If G is a network and Y is a set of nodes such that at most one of s and t is in Y, it is useful to define $G(Y)$, the *contraction of G by Y*, to be the network formed by shrinking the nodes in Y to a single node, eliminating loops, and combining multiple edges by adding their capacities.

We shall now describe the GGT algorithm. The core of the algorithm is a recursive function called *slice*, which is applied to contracted versions of G. There are 4 values associated with each of these contracted networks: λ_1 and λ_3 and respective flows f_1 and f_3. Each of these parameters induces a prescribed cut. Parameter λ_1 gives minimum cut $(s, V-s)$ and $(V-t, t)$ is the minimum cut for λ_3 (note $\lambda_1 < \lambda_3$). The λ parameters are initialized from eqs. 2 and 3. The GGT breakpoint algorithm consists of the following two steps [53]:

Step 1. Compute λ_1 according to eq. 2 and λ_3 according to eq. 3. Compute a maximum flow f_1 and minimum cut $(Y_1, \overline{Y_1})$ for λ_1 such that $|Y_1$ is maximum by applying the push-relabel algorithm to G. Compute a maximum flow f_3 and minimum cut $(Y_3, \overline{Y_3})$ for λ_3 such that $|Y_3|$ is minimum by applying the push-relabel algorithm to G^R. Form G' from G by shrinking the nodes in $\overline{Y_3}$ to a single node, shrinking the nodes in Y_1 to a single node, eliminating loops, and combining multiple edges by adding their capacities.
Step 2. If G' contains at least three nodes, let f_1' and f_3' be the flows in G' corresponding to f_1 and f_3, respectively; perform *slice*$(G', \lambda_1, \lambda_3, f_1', f_3')$, where **slice** is defined as follows:

Procedure **slice**$(G, \lambda_1, \lambda_3, f_1, f_3)$

Step 1. Let λ_2 be the value of λ such $c_{\lambda_2}(s, V-s) = c_{\lambda_2}(V-t, t)$. This value will satisfy $\lambda_1 \leq \lambda_2 \leq \lambda_3$.

Step 2. Run the push-relabel algorithm for the value λ_2 on G starting with the preflow f_1' formed by increasing f_1 on edges (s,v) to saturate them and decreasing f_1 on edges (v,t) to meet the capacity constraints. As an initial valid labeling, use $d(v) = \min d'_{f_1}(v,t), d'_{f_1}(v,s) + n)$. Stop when one of the concurrent applications stops, having computed a maximum flow f_2. Suppose the push-relabel algorithm applied to G stops first (the other case is symmetric). Find the minimum cuts $(Y_2, \overline{Y_2})$ and $(Y_2', \overline{Y_2}')$ for λ_2 such that $|Y_2|$ is minimum and $|Y_2'|$ is maximum. If $|Y_2| > n/2$ complete the execution of the push-relabel algorithm on G^R and let f_2 be the resulting maximum flow.

Step 3. If $c_\lambda(Y_2, \overline{Y_2}) \neq c_\lambda(Y_2', \overline{Y_2}')$ for some λ, report λ as a breakpoint.

Step 4. If $Y_2 \neq s$, perform **slice**$(G(\overline{Y_2}, \lambda_1, \lambda_2, f_1, f_2)$. If $\overline{Y_2'} \neq t$, perform **slice**$(G(Y_2'), \lambda_2, \lambda_3, f_2, f_3)$.

Bibliography

[1] B. Alexe, T. Deselaers, and V. Ferrari. "What is an object?" In: *IEEE International Conference on Computer Vision and Pattern Recognition*. 2010.

[2] Pablo Arbelaez and Laurent Cohen. "Constrained image segmentation from hierarchical boundaries". In: *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on* 0 (2008), pp. 1–8.

[3] Pablo Arbelaez et al. "Contour Detection and Hierarchical Image Segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 99.PrePrints (2010).

[4] Pablo Arbelaez et al. "From contours to regions: An empirical evaluation". In: *IEEE International Conference on Computer Vision and Pattern Recognition*. 2009, pp. 2294–2301.

[5] Maxim A. Babenko et al. "Experimental Evaluation of Parametric Max-Flow Algorithms". In: *WEA*. 2007, pp. 256–269.

[6] S. Bagon, O. Boiman, and M. Irani. "What Is a Good Image Segment? A Unified Approach to Segment Extraction". In: *European Conference on Computer Vision* (2008), pp. 30–44.

[7] H.G. Barrow and R.J. Popplestone. "Relational Descriptions in Picture Processing". In: *Machine Intelligence* VI (1971), pp. 377–396.

[8] S. Belongie, J. Malik, and J. Puzicha. "Shape Matching and Object Recognition Using Shape Contexts". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (4 2002), pp. 509–522.

[9] Serge Belongie, Jitendra Malik, and Jan Puzicha. "Shape Context: A new descriptor for shape matching and object recognition". In: *Advances in Neural Information Processing Systems*. 2000, pp. 831–837.

[10] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. 1st ed. 2006. Corr. 2nd printing. Springer, Oct. 2007.

[11] Matthew B. Blaschko and Christoph H. Lampert. "Learning to Localize Objects with Structured Output Regression". In: *European Conference on Computer Vision*. 2008, pp. 2–15. ISBN: 978-3-540-88681-5.

[12] L. Bo and C. Sminchisescu. "Efficient Match Kernels between Sets of Features for Visual Recognition". In: *Advances in Neural Information Processing Systems*. 2009.

[13] O. Boiman, E. Shechtman, and M. Irani. "In defense of Nearest-Neighbor based image classification". In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. 2008, pp. 1 –8.

[14] E. Borenstein, E. Sharon, and S. Ullman. "Combining Top-Down and Bottom-Up Segmentation". In: *Computer Vision and Pattern Recognition Workshop* (2004), pp. 46–46.

[15] E. Borenstein and S. Ullman. "Class-Specific, Top-Down Segmentation". In: *European Conference on Computer Vision*. 2002.

[16] E. Borenstein and S. Ullman. "Combined Top-Down/Bottom-Up Segmentation". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 30.12 (2008), pp. 2109 – 2125.

[17] Anna Bosch, Andrew Zisserman, and Xavier Munoz. "Representing shape with a spatial pyramid kernel". In: *ACM International Conference on Image and Video Retrieval* (2007), pp. 401–408.

[18] Y. Boykov and G. Funka-Lea. "Graph Cuts and Efficient N-D Image Segmentation". In: *International Journal of Computer Vision* 70.2 (2006), pp. 109–131.

[19] Y. Boykov and M.-P. Jolly. "Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in N-D images". In: *IEEE International Conference on Computer Vision*. Vol. 1. 2001, 105 –112 vol.1.

[20] Y. Boykov and V. Kolmogorov. "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26.9 (2004), pp. 1124 –1137. ISSN: 0162-8828.

[21] Leo Breiman. "Random Forests". In: *Machine Learning* 45.1 (2001), pp. 5–32.

[22] Thomas Brox and Jitendra Malik. "Object segmentation by long term analysis of point trajectories". In: *Proceedings of the 11th European conference on Computer vision: Part V*. ECCV'10. Heraklion, Crete, Greece: Springer-Verlag, 2010, pp. 282–295. ISBN: 3-642-15554-5, 978-3-642-15554-3. URL: `http://dl.acm.org/citation.cfm?id=1888150.1888173`.

[23] Alfred M. Bruckstein, Robert J. Holt, and Arun N. Netravali. "Discrete elastica". In: *Discrete Geometry for Computer Imagery*. 1996, pp. 59–72.

[24] Jaime Carbonell and Jade Goldstein. "The use of MMR, diversity-based reranking for reordering documents and producing summaries". In: *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. Melbourne, Australia: ACM, 1998, pp. 335–336.

[25] J. Carreira and C. Sminchisescu. *Constrained Parametric Min-Cuts for Automatic Object Segmentation, Release 1*. http://sminchisescu.ins.uni-bonn.de/code/cpmc/.

[26] J. Carreira and C. Sminchisescu. "Constrained Parametric Min-Cuts for Automatic Object Segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2010.

[27] Joao Carreira, Adrian Ion, and Cristian Sminchisescu. *Image Segmentation by Discounted Cumulative Ranking on Maximal Cliques*. Tech. rep. 06-2010. Computer Vision and Machine Learning Group, Institute for Numerical Simulation, University of Bonn, 2010.

[28] João Carreira, Fuxin Li, and Cristian Sminchisescu. *Ranking figure-ground hypotheses for object segmentation*. oral presentation at the PASCAL VOC 2009 Workshop, available online at `http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2009/workshop/`.

[29] Olivier Chapelle, Patrick Haffner, and Vladimir Vapnik. "Support vector machines for histogram-based image classification". In: *IEEE Transactions on Neural Networks* 10.5 (1999), pp. 1055–1064.

[30] D. Comaniciu and P. Meer. "Mean Shift: A Robust Approach Toward Feature Space Analysis". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.5 (2002), pp. 603–619.

[31] T. Cour and Jianbo Shi. "Recognizing objects by piecing together the Segmentation Puzzle". In: *IEEE International Conference on Computer Vision and Pattern Recognition*. 2007, pp. 1 –8.

[32] Daniel Cremers, Frank R. Schmidt, and Frank Barthel. "Shape Priors in Variational Image Segmentation: Convexity, Lipschitz Continuity and Globally Optimal Solutions". In: *IEEE International Conference on Computer Vision and Pattern Recognition* 0 (2008), pp. 1–6.

[33] Gabriela Csurka and Florent Perronnin. "A Simple High Performance Approach to Semantic Segmentation". In: *BMVC*. Leeds, UK, 2008.

[34] Gabriela Csurka and Florent Perronnin. "An Efficient Approach to Semantic Segmentation". In: *International Journal of Computer Vision* (2010), pp. 1–15.

[35] S. Dickinson et al. *Object Categorization: Computer and Human Vision Perspectives*. Cambridge University Press, 2009.

[36] P. Dollár, Z. Tu, and S. Belongie. "Supervised Learning of Edges and Object Boundaries". In: *IEEE International Conference on Computer Vision and Pattern Recognition*. 2006.

[37] Ian Endres and Andrew Hoiem. "Category Independent Object Proposals". In: *European Conference on Computer Vision*. 2010.

[38] M. Everingham et al. *The PASCAL Visual Object Classes Challenge 2009 (VOC2009) Results*. http://www.pascal-network.org/challenges/VOC/voc2009/workshop/index.html.

[39] M. Everingham et al. *The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results*. http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html.

[40] M. Everingham et al. *The PASCAL Visual Object Classes Challenge 2011 (VOC2011) Results*. http://www.pascal-network.org/challenges/VOC/voc2011/workshop/index.html.

[41] M. Everingham et al. "The Pascal Visual Object Classes (VOC) Challenge". In: *International Journal of Computer Vision* 88.2 (June 2010), pp. 303–338.

[42] Li Fei-Fei, Rob Fergus, and Pietro Perona. "Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories". In: *CVIU* 106.1 (2007), pp. 59–70.

[43] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester. *Discriminatively Trained Deformable Part Models, Release 4*. http://people.cs.uchicago.edu/ pff/latent-release4/.

[44] P.F. Felzenszwalb and J.D. Schwartz. "Hierarchical Matching of Deformable Shapes". In: *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*. 2007, pp. 1 –8. DOI: 10.1109/CVPR.2007.383018.

[45] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. "Efficient Graph-Based Image Segmentation". In: *International Journal of Computer Vision* 59.2 (2004), pp. 167–181.

[46] Pedro F. Felzenszwalb et al. "Object Detection with Discriminatively Trained Part-Based Models". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32 (2010), pp. 1627–1645.

[47] Vittorio Ferrari, Frédéric Jurie, and Cordelia Schmid. "Accurate Object Detection with Deformable Shape Models Learnt from Images". In: *IEEE International Conference on Computer Vision and Pattern Recognition*. 2007.

[48] L R Ford and D R Fulkerson. "Maximal flow through a network". In: *Canadian Journal of Mathematics* (1956).

[49] David Forsyth et al. *Finding Pictures of Objects in Large Collections of Images*. Tech. rep. Berkeley, CA, USA, 1996.

[50] C. Fowlkes, D. Martin, and J. Malik. "Learning affinity functions for image segmentation: combining patch-based and gradient-based approaches". In: *IEEE International Conference on Computer Vision and Pattern Recognition*. Vol. 2. 2003, II–54–61 vol.2.

[51] Eugene C. Freuder. "A computer system for visual recognition using active knowledge". In: *Proceedings of the 5th international joint conference on Artificial intelligence - Volume 2*. Cambridge, USA: Morgan Kaufmann Publishers Inc., 1977, pp. 671–677.

[52] Brian Fulkerson, Andrea Vedaldi, and Stefano Soatto. "Class segmentation and object localization with superpixel neighborhoods". In: *IEEE International Conference on Computer Vision*. 2009, pp. 670 –677.

[53] G. Gallo, M. D. Grigoriadis, and R. E. Tarjan. "A fast parametric maximum flow algorithm and applications". In: *SIAM J. Comput.* 18.1 (1989), pp. 30–55.

[54] Peter V. Gehler and Sebastian Nowozin. "On Feature Combination for Multiclass Object Classification". In: *ICCV*. 2009.

[55] A V Goldberg and R E Tarjan. "A new approach to the maximum flow problem". In: *STOC '86: Proceedings of the eighteenth annual ACM symposium on Theory of computing*. Berkeley, California, United States: ACM, 1986, pp. 136–146. ISBN: 0-89791-193-8.

[56] Josep M. Gonfaus et al. "Harmony Potentials for Joint Classification and Segmentation". In: *IEEE International Conference on Computer Vision and Pattern Recognition*. San Francisco, California, USA, 2010, pp. 1–8.

[57] Stephen Gould, Richard Fulton, and Daphne Koller. "Decomposing a Scene into Geometric and Semantically Consistent Regions". In: *IEEE International Conference on Computer Vision*. 2009.

[58] Stephen Gould, Tianshi Gao, and Daphne Koller. "Region-based Segmentation and Object Detection". In: *Advances in Neural Information Processing Systems*. Ed. by Y. Bengio et al. 2009, pp. 655–663.

[59] K. Grauman and T. Darrell. "The pyramid match kernel: discriminative classification with sets of image features". In: *IEEE International Conference on Computer Vision*. Vol. 2. 2005, 1458 –1465 Vol. 2.

[60] D. M. Greig, B. T. Porteous, and A. H. Seheult. "Exact Maximum A Posteriori Estimation for Binary Images". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 51.2 (1989), pp. 271–279.

[61] G. Griffin, A. Holub, and P. Perona. *Caltech-256 Object Category Dataset*. Tech. rep. 7694. California Institute of Technology, 2007.

[62] W. Eric L. Grimson. *Object recognition by computer: the role of geometric constraints*. Cambridge, MA, USA: MIT Press, 1990. ISBN: 0-262-07130-4.

[63] Chunhui Gu et al. "Recognition Using Regions". In: *IEEE International Conference on Computer Vision and Pattern Recognition*. 2009.

[64] Robert M. Haralick and Linda G. Shapiro. "Image segmentation techniques". In: *Computer Vision, Graphics, and Image Processing* 29.1 (1985), pp. 100–132.

[65] Xuming He, Richard S. Zemel, and Miguel Carreira-Perpiñán. "Multiscale Conditional Random Fields for Image Labeling". In: *IEEE International Conference on Computer Vision and Pattern Recognition* 2 (2004), pp. 695–702.

[66] D. Hochbaum. "Polynomial time algorithms for bi-criteria, multi-objective and ratio problems in clustering and imaging. Part I: Normalized cut and ratio regions". In: *arxiv* (2008).

[67] Dorit S. Hochbaum. "The Pseudoflow Algorithm: A New Algorithm for the Maximum-Flow Problem". In: *Oper. Res.* 56 (4 2008), pp. 992–1009.

[68] Derek Hoiem, Alexei A. Efros, and Martial Hebert. "Geometric Context from a Single Image". In: *IEEE International Conference on Computer Vision* 1 (2005), pp. 654–661.

[69] Daniel P. Huttenlocher and Shimon Ullman. "Recognizing solid objects by alignment with an image". In: *International Journal of Computer Vision* 5.2 (1990), pp. 195–212.

[70] S. Ioffe and D. A. Forsyth. "Probabilistic Methods for Finding People". In: *Int. J. Comput. Vision* 43 (1 2001), pp. 45–68. URL: `http://dl.acm.org/citation.cfm?id=543015.543018`.

[71] A. Ion, J. Carreira, and C. Sminchisescu. "Image Segmentation by Figure-Ground Composition into Maximal Cliques". In: *IEEE International Conference on Computer Vision*. 2011.

[72] A. Ion, J. Carreira, and C. Sminchisescu. "Probabilistic Joint Image Segmentation and Labeling". In: *Advances in Neural Information Processing Systems*. 2011.

[73] Ernst Ising. "Beitrag zur Theorie des Ferromagnetismus". In: *Zeitschrift für Physik A Hadrons and Nuclei* 31 (1 1925). 10.1007/BF02980577, pp. 253–258.

[74] M. Hebert C. Schmid J. Ponce and Andrew Zisserman. *Toward category-level object recognition*. Vol. 4170. Springer, 2006.

[75] Abhishek Jaiantilal. *Classification and Regression by randomForest-matlab*. Available at `http://code.google.com/p/randomforest-matlab`. 2009.

[76] T. Kadir, A. Zisserman, and J. M. Brady. "An Affine Invariant Salient Region Detector". In: *European Conference on Computer Vision*. Springer-Verlag, 2004.

[77] John Kaufhold and Anthony Hoogs. "Learning to Segment Images Using Region-Based Perceptual Features". In: *IEEE International Conference on Computer Vision and Pattern Recognition* 2 (2004), pp. 954–961.

[78] P. Kohli, L. Ladicky, and P.H.S. Torr. "Robust higher order potentials for enforcing label consistency". In: *IEEE International Conference on Computer Vision and Pattern Recognition*. 2008, pp. 1–8.

[79] Pushmeet Kohli and Philip H. S. Torr. "Dynamic Graph Cuts for Efficient Inference in Markov Random Fields". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29 (12 2007), pp. 2079–2088.

[80] V. Kolmogorov, Y. Boykov, and C. Rother. "Applications of parametric maxflow in computer vision". In: *IEEE International Conference on Computer Vision* (2007), pp. 1–8.

[81] V. Kolmogorov and R. Zabin. "What energy functions can be minimized via graph cuts?" In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26.2 (2004), pp. 147–159.

[82] A. Kumar and C. Sminchisescu. "Support Kernel Machines for Object Recognition". In: *IEEE International Conference on Computer Vision*. 2007.

[83] M. Pawan Kumar, P.H.S. Torr, and A. Zisserman. "OBJCUT: Efficient Segmentation Using Top-Down and Bottom-Up Cues". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32 (2010), pp. 530–545.

[84] M. Pawan Kumar, Philip H. S. Torr, and A. Zisserman. "OBJ CUT". In: *IEEE International Conference on Computer Vision and Pattern Recognition*. 2005. ISBN: 0-7695-2372-2.

[85] L. Ladicky et al. "Associative Hierarchical CRFs for Object Class Image Segmentation". In: *IEEE International Conference on Computer Vision*. 2009.

[86] Lubor Ladicky et al. "What, Where & How Many ? Combining Object Detectors and CRFs". In: *European Conference on Computer Vision*. 2010.

[87] Kevin Lai et al. "A Scalable Tree-Based Approach for Joint Object and Pose Recognition". In: *AAAI*. 2011.

[88] C.H. Lampert, M.B. Blaschko, and T. Hofmann. "Beyond sliding windows: Object localization by efficient subwindow search". In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. 2008, pp. 1 –8.

[89] S. Lazebnik, C. Schmid, and J. Ponce. "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories". In: *IEEE International Conference on Computer Vision and Pattern Recognition*. Vol. 2. 2006, pp. 2169 –2178.

[90] Y. J. Lee, J. Kim, and K. Grauman. "Key-Segments for Video Object Segmentation". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2011*.

[91] Bastian Leibe, Ales Leonardis, and Bernt Schiele. "Robust Object Detection with Interleaved Categorization and Segmentation". In: *International Journal of Computer Vision* 77.1-3 (2008), pp. 259–289.

[92] V. Lempitsky, A. Blake, and C. Rother. "Image Segmentation by Branch-and-Mincut". In: *European Conference on Computer Vision*. 2008, IV: 15–29.

[93] Anat Levin and Yair Weiss. "Learning to Combine Bottom-Up and Top-Down Segmentation". In: *International Journal of Computer Vision* 81.1 (2009), pp. 105–118.

[94] A. Levinshtein, C. Sminchisescu, and S. Dickinson. "Optimal Contour Closure by Superpixel Grouping". In: *European Conference on Computer Vision*. 2010.

[95] Fuxin Li, João Carreira, and Cristian Sminchisescu. "Object Recognition as Ranking Holistic Figure-Ground Hypotheses". In: *IEEE International Conference on Computer Vision and Pattern Recognition*. 2010.

[96] Fuxin Li, Catalin Ionescu, and Cristian Sminchisescu. "Random Fourier Approximations for Skewed Multiplicative Histogram Kernels". In: *Annual Symposium of the German Association for Pattern Recognition (DAGM)*. 2010.

[97] Yongsub Lim, Kyomin Jung, and Pushmeet Kohli. "Energy Minimization under Constraints on Label Counts". In: *European Conference on Computer Vision*. Vol. 6312. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2010, pp. 535–551.

[98] Tie Liu et al. "Learning to Detect A Salient Object". In: *IEEE International Conference on Computer Vision and Pattern Recognition*. 2007, pp. 1 –8.

[99] David G. Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". In: *International Journal of Computer Vision* 60.2 (2004), pp. 91–110.

[100] David G. Lowe. *Perceptual Organization and Visual Recognition*. Norwell, MA, USA: Kluwer Academic Publishers, 1985. ISBN: 089838172X.

[101] M. Maire et al. "Using contours to detect and localize junctions in natural images". In: *IEEE International Conference on Computer Vision and Pattern Recognition* 0 (2008), pp. 1–8.

[102] T. Malisiewicz and A. Efros. "Improving Spatial Support for Objects via Multiple Segmentations". In: *British Machine Vision Conference* (2007).

[103] T. Malisiewicz and A. Efros. "Recognition by Association via Learning Per-exemplar Distances". In: *IEEE International Conference on Computer Vision and Pattern Recognition*. 2008.

[104] Tomasz Malisiewicz, Abhinav Gupta, and Alexei A. Efros. "Ensemble of Exemplar-SVMs for Object Detection and Beyond". In: *ICCV*. 2011.

[105] David Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. New York, NY, USA: Henry Holt and Co., Inc., 1982. ISBN: 0716715678.

[106] D. Martin et al. "A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics". In: *ICCV*. Vol. 2. 2001, pp. 416–423.

[107] Krystian Mikolajczyk and Cordelia Schmid. "Scale and affine invariant interest point detectors". In: *International Journal of Computer Vision* 60.1 (2004), pp. 63–86.

[108] G. Mori et al. "Recovering human body configurations: combining segmentation and recognition". In: *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*. Vol. 2. 2004, II–326 –II–333 Vol.2.

[109] J.L. Muerle, and D.C. Allen. "Experimental evaluation of techniques for automatic segmentation of objects in a complex scene." In: *Pictorial Pattern Recognition*. 1968, pp. 3–13.

[110] Joseph L. Mundy. "Object Recognition in the Geometric Era: A Retrospective". In: *Toward Category-Level Object Recognition*. 2006, pp. 3–28.

[111] Stephen E. Palmer. *Vision Science: Photons to Phenomenology*. The MIT Press, 1999.

[112] Caroline Pantofaru, Cordelia Schmid, and Martial Hebert. "Object Recognition by Integrating Multiple Image Segmentations". In: *European Conference on Computer Vision*. 2008.

[113] Constantine Papageorgiou, Michael Oren, and Tomaso Poggio. "A General Framework for Object Detection". In: *IEEE International Conference on Computer Vision*. 1998, pp. 555–562.

[114] Bo Peng and Olga Veksler. “Parameter Selection for Graph Cut Based Image Segmentation”. In: *British Machine Vision Conference*. 2008.

[115] J. C. Picard and H. D. Ratliff. “Minimum cuts and related problems”. In: *Networks* 5 (1975), pp. 357–370.

[116] Nicolas Pinto, David D Cox, and James J DiCarlo. “Why is Real-World Visual Object Recognition Hard?” In: *PLoS Comput Biol* 4.1 (Jan. 2008), e27.

[117] Andrew Rabinovich et al. “Model Order Selection and Cue Combination for Image Segmentation”. In: *IEEE International Conference on Computer Vision and Pattern Recognition* 1 (2006), pp. 1130–1137.

[118] Andrew Rabinovich et al. “Objects in Context”. In: *IEEE International Conference on Computer Vision* (2007).

[119] Ali Rahimi and Ben Recht. “Random features for large-scale kernel machines”. In: *Advances in Neural Information Processing Systems*. 2007.

[120] Xiaofeng Ren and Jitendra Malik. “Learning a Classification Model for Segmentation”. In: *IEEE International Conference on Computer Vision*. Vol. 1. Los Alamitos, CA, USA: IEEE Computer Society, 2003, p. 10. ISBN: 0-7695-1950-4.

[121] Azriel Rosenfeld. “From image analysis to computer vision: an annotated bibliography, 1955-1979”. In: *Computer Vision and Image Understanding* 84 (2 2001), pp. 298–324.

[122] C. Rother, V. Kolmogorov, and A. Blake. “"GrabCut": interactive foreground extraction using iterated graph cuts”. In: *ACM Trans. Graph.* 23.3 (2004), pp. 309–314.

[123] B. Russell et al. “Using Multiple Segmentations to Discover Objects and their Extent in Image Collections”. In: *IEEE International Conference on Computer Vision and Pattern Recognition*. Vol. 2. Los Alamitos, CA, USA: IEEE Computer Society, 2006, pp. 1605–1614.

[124] M. Galun R. Basri S. Alpert and A. Brandt. “Image Segmentation by Probabilistic Bottom-Up Aggregation and Cue Integration”. In: *IEEE International Conference on Computer Vision and Pattern Recognition*. Vol. 0. Los Alamitos, CA, USA: IEEE Computer Society, 2007, pp. 1–8.

[125] Jorge Sánchez and Florent Perronnin. “High-dimensional signature compression for large-scale image classification”. In: *CVPR*. 2011, pp. 1665–1672.

[126] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. “Evaluating Color Descriptors for Object and Scene Recognition”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 9 (2010), pp. 1582–1596.

[127] T. Schoenemann, F. Kahl, and D. Cremers. “Curvature Regularity for Region-based Image Segmentation and Inpainting: A Linear Programming Relaxation”. In: *IEEE International Conference on Computer Vision* (2009).

[128] Thomas Schoenemann and Daniel Cremers. “A Combinatorial Solution for Model-Based Image Segmentation and Real-Time Tracking”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32 (2010), pp. 1153–1164.

[129] Thomas Schoenemann and Daniel Cremers. “Globally Optimal Image Segmentation with an Elastic Shape Prior”. In: *IEEE International Conference on Computer Vision* 0 (2007), pp. 1–6.

[130] E. Sharon et al. “Hierarchy and adaptivity in segmenting visual scenes”. In: *Nature* 442.7104 (2006), pp. 719–846.

[131] J. Shi and J. Malik. “Normalized Cuts and Image Segmentation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2000).

[132] Jamie Shotton et al. “Real-time human pose recognition in parts from single depth images”. In: *IEEE International Conference on Computer Vision and Pattern Recognition*. 2011, pp. 1297–1304.

[133] Jamie Shotton et al. “TextonBoost: Joint Appearance, Shape and Context Modeling for Multi-Class Object Recognition and Segmentation”. In: *European Conference on Computer Vision*. 2006, pp. 1–15.

[134] Jamie Shotton et al. “TextonBoost for Image Understanding: Multi-Class Object Recognition and Segmentation by Jointly Modeling Texture, Layout, and Context”. In: *International Journal of Computer Vision* 81 (1 2009), pp. 2–23.

[135] Daniel D. Sleator and Robert Endre Tarjan. “A data structure for dynamic trees”. In: *J. Comput. Syst. Sci.* 26 (3 1983), pp. 362–391.

[136] Praveen Srinivasan and Jianbo Shi. “Botom-up Recognition and Parsing of the Human Body”. In: *IEEE International Conference on Computer Vision and Pattern Recognition*. 2007.

[137] Andrew Stein, Thomas Stepleton, and Martial Hebert. “Towards Unsupervised Whole-Object Segmentation: Combining Automated Matting with Boundary Detection”. In: *IEEE International Conference on Computer Vision and Pattern Recognition*. 2008.

[138] J.M. Tenenbaum and H.G. Barrow. “Experiments in interpretation-guided segmentation”. In: *Artificial Intelligence* 8.3 (1977), pp. 241–274.

[139] Sinisa Todorovic and Narendra Ahuja. “Learning Subcategory Relevances for Category Recognition”. In: *IEEE International Conference on Computer Vision and Pattern Recognition*. 2008.

[140] A. Toshev, B. Taskar, and K. Daniilidis. “Object detection via boundary structure segmentation”. In: *IEEE International Conference on Computer Vision and Pattern Recognition*. 2010, pp. 950–957.

[141] Ioannis Tsochantaridis et al. “Support vector machine learning for interdependent and structured output spaces”. In: *International Conference on Machine Learning*. 2004.

[142] Zhuowen Tu and Song-Chun Zhu. “Parsing Images into Regions, Curves, and Curve Groups”. In: *Int. J. Comput. Vision* 69 (2 2006), pp. 223–249.

[143] John Wilder Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977.

[144] Tinne Tuytelaars and Luc Van Gool. “Matching Widely Separated Views Based on Affine Invariant Regions”. In: *Int. J. Comput. Vision* 59 (1 2004), pp. 61–85.

[145] S. Ullman. *High-level vision: object recognition and visual cognition*. Bradford Books. MIT Press, 2000. ISBN: 9780262710077.

[146] Ranjith Unnikrishnan, Caroline Pantofaru, and Martial Hebert. In: *Toward Objective Evaluation of Image Segmentation Algorithms* 29.1 (2007), pp. 929–944.

[147] L. Van Gool, P. Kempenaers, and A. Oosterlinck. "Recognition and semi-differential invariants". In: *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91., IEEE Computer Society Conference on*. 1991, pp. 454 –460.

[148] A. Vedaldi and B. Fulkerson. *VLFeat: An Open and Portable Library of Computer Vision Algorithms*. http://www.vlfeat.org/. 2008.

[149] A. Vedaldi et al. "Multiple Kernels for Object Detection". In: *IEEE International Conference on Computer Vision*. 2009.

[150] Andrea Vedaldi and Andrew Zisserman. "Efficient Additive Kernels via Explicit Feature Maps". In: *IEEE International Conference on Computer Vision and Pattern Recognition*. 2010.

[151] Sara Vicente, Vladimir Kolmogorov, and Carsten Rother. "Graph cut based image segmentation with connectivity priors". In: *IEEE International Conference on Computer Vision and Pattern Recognition* (2008), pp. 0–7.

[152] Sara Vicente, Vladimir Kolmogorov, and Carsten Rother. "Joint optimization of segmentation and appearance models". In: *IEEE International Conference on Computer Vision*. 2009, pp. 755–762.

[153] Sara Vicente, Carsten Rother, and Vladimir Kolmogorov. "Object cosegmentation". In: *IEEE International Conference on Computer Vision and Pattern Recognition*. 2011, pp. 2217–2224.

[154] P. Viola and M. Jones. "Rapid object detection using a boosted cascade of simple features". In: *IEEE International Conference on Computer Vision and Pattern Recognition*. 2001.

[155] P. Viola and M. Jones. "Rapid object detection using a boosted cascade of simple features". In: *IEEE International Conference on Computer Vision and Pattern Recognition* (2001).

[156] Song Wang and Jeffrey Mark Siskind. "Image segmentation with Ratio Cut". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25 (2003), pp. 675–690.

[157] M. Wertheimer. "Laws of Organization in Perceptual Forms (partial translation)". In: *A sourcebook of Gestalt Psychology*. 1938, pp. 71–88.

[158] Z. Wu and R. Leahy. "An Optimal Graph Theoretic Approach to Data Clustering: Theory and Its Application to Image Segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15.11 (1993), pp. 1101–1113.

[159] Yi Yang et al. "Layered Object Detection for Multi-Class Segmentation". In: *IEEE International Conference on Computer Vision and Pattern Recognition*. 2010.

[160] H.-F. Yu et al. "Large linear classification when data cannot fit in memory". In: *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2010.

[161] Stella X. Yu and Jianbo Shi. "Object-Specific Figure-Ground Segregation". In: *IEEE International Conference on Computer Vision and Pattern Recognition* 2 (2003), p. 39.

[162] Hao Zhang et al. "SVM-KNN: Discriminative Nearest Neighbor Classification for Visual Category Recognition". In: *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*. Vol. 2. 2006, pp. 2126 –2136.

[163] S. C. Zhu and D. Mumford. "Learning Generic Prior Models for Visual Computation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19.11 (1997).

[164] S.C. Zhu, T.S. Lee, and A.L. Yuille. "Region competition: unifying snakes, region growing, energy/Bayes/MDL for multi-band image segmentation". In: *Computer Vision, 1995. Proceedings., Fifth International Conference on*. June 1995, pp. 416 –423.

[165] Song-Chun Zhu. "Embedding Gestalt laws in Markov random fields". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21.11 (1999), pp. 1170 –1187. ISSN: 0162-8828.

[166] Andrew Zisserman et al. "3D object recognition using invariance". In: *Artificial Intelligence* 78.1-2 (1995). Special Volume on Computer Vision, pp. 239 –288.

List of Figures

List of Tables