

Hierarchical Image Segmentation Based on Semidefinite Programming

Jens Keuchel, Matthias Heiler, and Christoph Schnörr

CVGPR-Group, Dept. Math. and Comp. Science
University of Mannheim, D-68131 Mannheim, Germany
{jkeuchel,heiler,schnoerr}@uni-mannheim.de
<http://www.cvgpr.uni-mannheim.de>

Abstract. Image segmentation based on graph representations has been a very active field of research recently. One major reason is that pairwise similarities (encoded by a graph) are also applicable in general situations where prototypical image descriptors as partitioning cues are no longer adequate. In this context, we recently proposed a novel convex programming approach for segmentation in terms of optimal graph cuts which compares favorably with alternative methods in several aspects.

In this paper we present a fully elaborated version of this approach along several directions: first, an image preprocessing method is proposed to reduce the problem size by several orders of magnitude. Furthermore, we argue that the hierarchical partition tree is a natural data structure as opposed to enforcing multiway cuts directly. In this context, we address various aspects regarding the fully automatic computation of the final segmentation. Experimental results illustrate the encouraging performance of our approach for unsupervised image segmentation.

1 Introduction

The segmentation of images into coherent parts is a key problem of computer vision. It is widely agreed that in order to properly solve this problem, both data-driven and model-driven approaches have to be taken into account [1].

Concerning the data-driven part, graph-theoretical approaches are more suited for *unsupervised* segmentation than approaches working in Euclidean spaces: as opposed to representations based on (dis-)similarity relations, class representations based on Euclidean distances (and variants) are too restrictive to capture signal variability in low-level vision [2]. This claim also appears to be supported by research on human perception [3].

The unsupervised partitioning of graphs constitutes a difficult combinatorial optimization problem. Suitable problem relaxations like the mean-field approximation [4,5] or spectral relaxation [6,7] are necessary to compromise about computational complexity and quality of approximate solutions.

Recently, a novel convex programming approach utilizing semidefinite relaxation has shown to be superior regarding optimization quality, the absence of heuristic tuning parameters, and the possibility to mathematically constrain

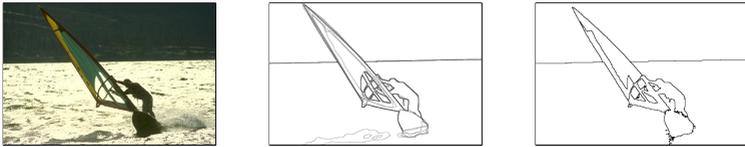


Fig. 1. A color image from the Berkeley segmentation dataset [9] (left). Comparing the segmentation boundaries calculated with the semidefinite programming relaxation (right) to the human segmentations (middle), the high quality of the SDP relaxation result is reflected by a high F -measure (see Section 5) of 0.92.

segmentations, at the cost of an increased but still moderate polynomial computational complexity [8]. This motivates to elaborate this approach towards a fully automatic and efficient unsupervised segmentation scheme providing a hierarchical data structure of coherent image parts which, in combination with model-based processing, may be explored for the purpose of scene interpretation (see Fig. 1 for an example result).

To this end, we consider a hierarchical framework for the binary partitioning approach presented in [8] to obtain a segmentation into multiple clusters (Section 2). To reduce the problem size by several orders of magnitude (to less than 0.01% of the all-pixel-based graph), we discuss an over-segmentation technique [10] which forms coherent “superpixels” [11] in a preprocessing step (Section 3). Section 4 treats various aspects concerning the development of a fully automatic unsupervised segmentation scheme. Experimental results based on a benchmark dataset of real world scenes [9] and comparisons with the normalized cut criterion illustrate the encouraging performance of our approach (Section 5).

2 Image Segmentation via Graph Cuts

The problem of image segmentation based on pairwise affinities can be formulated as a graph partitioning problem in the following way: consider the weighted graph $G(V, E)$ with locally extracted image features as vertices V and pairwise similarity values $w_{ij} \in \mathbb{R}_0^+$ as edge-weights. Segmenting the image into two parts then corresponds to partitioning the nodes of the graph into disjoint groups S and $\bar{S} = V \setminus S$. Representing such a partition by an indicator vector $x \in \{-1, +1\}^n$ (where $n = |V|$), the quality of a binary segmentation can be measured by the weight of the corresponding cut in the graph: $\text{cut}(S, \bar{S}) = \sum_{i \in S, j \in \bar{S}} w_{ij} = \frac{1}{4} x^\top L x$, where $L = D - W$ denotes the graph Laplacian matrix, and D is the diagonal degree matrix with $D_{ii} = \sum_{j \in V} w_{ij}$.

As directly minimizing the cut favors unbalanced segmentations, several methods for defining more suitable measures have been suggested in the literature. One of the most popular is the normalized cut criterion [7], which tries to avoid unbalanced partitions by appropriately scaling the cut-value. Since the corresponding cost function yields an NP-hard minimization problem, a spectral relaxation method is used to compute an approximate solution which is based on calculating minimal eigenvectors of the normalized Laplacian $L' = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$.

To get a binary solution of the original problem, these eigenvectors are then thresholded appropriately.

SDP relaxation. In this paper, we employ an alternative technique to find balanced partitions which originates from spectral graph theory [6]. As a starting point consider the following combinatorial problem formulation:

$$\begin{aligned} \min_{x \in \{-1, +1\}^n} \quad & x^\top Lx \\ \text{s.t.} \quad & c^\top x = b. \end{aligned} \quad (1)$$

Thus, instead of *normalizing* the cut-value as in [7], in this case an *additional balancing constraint* $c^\top x = b$ is used to compute favorable partitions. A classical approach to find a balanced segmentation uses $c = (1, \dots, 1)^\top$ and $b = 0$, which is reasonable for graphs where each vertex is equally important. However, this may not be the case for the preprocessed images considered here; we will therefore discuss alternative settings for c and b in Section 4.

In order to find an approximate solution for the NP-hard problem (1), an advanced method is proposed in [8] which in contrast to spectral relaxation is not only able to handle the general linear constraint, but also takes into account the integer constraint on x in a better way. Observing that the cut-weight can be rewritten as $x^\top Lx = \text{tr}(Lxx^\top)$, the problem variables are lifted into a higher dimensional space by introducing the matrix variable $X = xx^\top$. Dropping the rank one constraint on X and using arbitrary positive semidefinite matrices $X \succeq 0$ instead, we obtain the following relaxation of (1):

$$\begin{aligned} \min_{X \succeq 0} \quad & \text{tr}(LX) \\ \text{s.t.} \quad & \text{tr}(cc^\top X) = b^2 \\ & \text{tr}(e_i e_i^\top X) = 1 \quad \text{for } i = 1, \dots, n, \end{aligned} \quad (2)$$

where $e_i \in \mathbb{R}^n$ denotes the i -th unit vector (see [8] for details).

The important point is that (2) belongs to the class of *semidefinite programs* (SDP), which can be solved in polynomial time to arbitrary precision, without needing to adjust any additional tuning parameters (see, e.g., [12]). To finally recover an integer solution x from the computed solution matrix X of (2), we use a randomized approximation technique [13]. Since this method does not enforce the balancing constraint from (1), it rather serves as a strong bias to guide the search instead of a strict requirement (cf. [8]).

Hierarchical clustering. In order to find segmentations of the image into multiple parts, we employ a hierarchical framework (e.g. [14]). In contrast to direct multiclass techniques (cf. [15,16]), the original cost function is used throughout the segmentation process, but for different (and usually smaller) problems in each step. As a consequence, the number k of segments does not need to be defined in advance, but can be chosen during the computation (which is more feasible for *unsupervised* segmentation tasks). Moreover, the subsequent splitting of segments yields a hierarchy of segmentations, so that changing k leads to similar segmentations. However, as no global cost function is optimized, additional decision criteria are needed concerning the selection of the next partitioning step

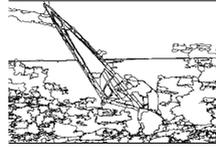


Fig. 2. 304 image patches are obtained for the image from Fig. 1 by over-segmenting it with mean shift. Note that in accordance with the homogeneous regions of the image, the patches differ in size. In this way, the splitting of such regions during the hierarchical graph cut segmentation is efficiently prevented.

and when to stop the hierarchical process. We will consider such criteria in Section 4.

3 Reducing the Problem Size

One important issue for segmentation methods based on graph representations is the size of the corresponding similarity matrix. If the vertex set V contains the pixels of an image, the size of the similarity matrix is equal to the squared number of pixels, and therefore generally too large to fit into computer memory completely (e.g. for an image of 481×321 pixels — the size of the images from the Berkeley segmentation dataset [9] — the similarity matrix contains $154401^2 \approx 23.8$ billion entries). As reverting to sparse matrices (which works efficiently for spectral methods) is of no avail for the SDP relaxation approach, we suggest to reduce the problem size in a preprocessing step. While in this context, approaches based on probabilistic sampling have recently been applied successfully to image segmentation problems [17,18], we propose a different technique.

Over-segmentation with mean shift. Our method is based on the straightforward idea to abandon pixels as graph vertices and to use small image patches (or “superpixels” [11]) of coherent structure instead. In fact, it can be argued that this is even a more natural image representation than pixels as those are merely the result of the digital image discretization. The real world does not consist of pixels!

In principle, any unsupervised clustering technique could be used as a preprocessing step to obtain such image patches of coherent structure. We apply the mean shift procedure [10], as it does not smooth over clear edges and results in patches of varying size (see Fig. 2 for an example). In this way, the important structures of the image are maintained, while on the other hand the number of image features for the graph representation is greatly reduced.

In summary, the mean shift uses gradient estimation to iteratively seek modes of a density distribution in some Euclidean feature space. In our case, the feature vectors comprise the pixel positions along with their color in the perceptually uniform $L^*u^*v^*$ space. The number and size of the image patches is controlled by scaling the entries of the feature vectors with the spatial and the range bandwidth parameters h_s and h_r , respectively (see [10] for details).

In order to get an adequate problem size for the SDP relaxation approach, we determine these parameters semi-automatically: while the spatial bandwidth

h_s is set to a fixed fraction of the image size, we calculate the range bandwidth h_r by randomly picking a certain number of pixels from the image, computing their maximum distance d_{\max} in the $L^*u^*v^*$ color space, and setting h_r to a fraction of d_{\max} . Moreover, we fix the minimum size of a region to $M = 50$ pixels. For the images from the Berkeley dataset [9], experiments showed that setting $h_s = 5.0$ and $h_r = \frac{d_{\max}}{15}$ results in an appropriate number of 100–700 image patches (corresponding to less than 0.01% of the number of pixels).

Constructing the graph. Using the image patches obtained with mean shift as graph vertices, the corresponding affinities are defined by representing each patch i with its mean color y_i in $L^*u^*v^*$ space, and calculating the similarity weights w_{ij} between neighboring patches as $w_{ij} = l_{ij} \exp\left(-\frac{\|y_i - y_j\|_2}{h_r}\right)$, where l_{ij} denotes the length of the edge in the image between the patches i and j . Hence, the problem is represented by a locally connected graph.

Assuming that each pixel originally is connected to its four neighbors, the multiplication with l_{ij} simulates a standard coarsening technique for graph partitioning [14]: the weight between two patches is calculated as the sum of the weights between the pixels contained within these patches. As each patch is of largely homogeneous color, using the mean color y_i instead of exact pixel colors does not change the resulting weights considerably.

Note that additional cues like texture or intervening contours can be incorporated into the classification process by computing corresponding similarity values based on the image patches, and combining them appropriately (see e.g. [14,19]). However, we do not consider modified similarities here.

4 Towards a Fully Automatic Segmentation

Based on the image patches obtained with mean shift the SDP relaxation approach is applied hierarchically to successively find binary segmentations. While solving the relaxation itself does not require tuning any parameters, the hierarchical application necessitates to discuss strategies for building up the segmentation tree, which is the subject of this section.

Segmentation constraints. Concerning the balancing constraint $c^\top x = b$ in (1), the graph vertices represented by the entries of c now correspond to image patches of varying size. For this reason, we calculate the number of pixels m_i contained in each patch i and set $c_i = m_i$ instead of $c_i = 1$, while retaining $b = 0$. In this way, the SDP relaxation seeks for two coherent parts with each containing approximately the same number of pixels.

However, if the part of the image under consideration in the current step contains a dominating patch k with $c_k = \max_i c_i \gg c_j$ for all $j \neq k$, segmentation into equally sized parts may not be possible. Nevertheless, we can still produce a feasible instance of the SDP relaxation in this case by adjusting the value of b in (1) appropriately, e.g. to $b = c_k - \frac{1}{2} \sum_{i \neq k} c_i$. Note that such an adjustment is not possible for spectral relaxation methods!

Which segment to split next? This question arises after each binary partitioning step. As the goal of unsupervised image segmentation mainly consists

in capturing the global impression of the scene, large parts of coherent structure should always be preferred to finer details. For this reason, we generally select the largest existing segment as the next candidate to be split.

However, we allow for two exceptions to this general rule: (1) If the candidate segment contains less than a certain number of patches (which we set to 8 in our experiments), it is not split any further. This prevents dividing the image into too much detail. (2) If the cut-value obtained for the candidate segment is too large, this split is rejected, since this indicates that the structure of this segment is already quite coherent. To decide when a cut-value z is too large, we compare it against the sum of all edge-weights w' (which is an upper bound on z): only if z is smaller than 2% of w' , the corresponding split is accepted.

Stopping criteria. The probably most difficult question in connection to unsupervised image segmentation concerns the number of parts the image consists of, or equivalently, when to stop the hierarchical segmentation process. As every human is likely to answer this question differently, one could even argue that without defining the desired granularity, image segmentation becomes an ill-posed problem. The hierarchical SDP relaxation approach offers two possible stopping criteria based on the desired granularity: the first one directly defines the maximum number of parts for the final segmentation. The second one is based on the fact that adding the cut-values results in an increasing function depending on the step number, which is bounded above by w' . Therefore, we can introduce the additional criterion to stop the hierarchical segmentation process when the complete cut value becomes larger than a certain percentage of w' .

5 Experimental Results

To evaluate the performance of our hierarchical segmentation algorithm, we apply it to images from the Berkeley segmentation dataset [9], which contains images of a wide variety of natural scenes. Moreover, this dataset also provides “ground-truth” data in the form of segmentations produced by humans (cf. Fig. 1), which allows to measure the performance of our algorithm quantitatively. Some exemplary results are depicted in Fig. 3. These encouraging segmentations are computed in less than 5 minutes on a Pentium 2 GHz processor.

As a quantitative measure of the segmentation quality, we use the precision-recall framework presented in [19]. In this context, the so-called F -measure is a valuable statistical performance indicator of a segmentation that captures the trade-off between accuracy and noise by giving values between 0 (bad segmentation) and 1 (good segmentation). For the results shown in Fig. 3, the corresponding F -measures confirm the positive visual impression.

For comparison, we also apply the normalized cut approach within the same hierarchical framework with identical parameter settings. While the results indicate the superiority of the SDP relaxation approach, this one-to-one comparison should be judged with care: as the normalized cut relaxation cannot appropriately take into account the varying patch sizes, the over-segmentation produced with mean shift may not be an adequate starting point for this method.

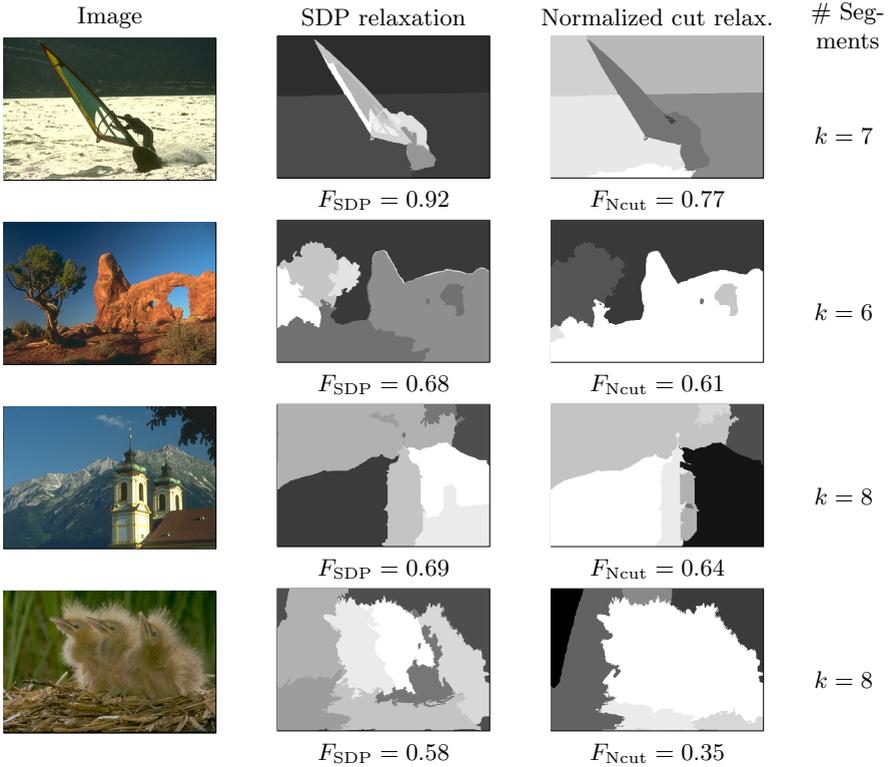


Fig. 3. Segmentation results for four color images (481×321 pixels) from the Berkeley segmentation dataset [9]. Note the superior quality of the segmentations obtained with the SDP relaxation approach in comparison to the normalized cut relaxation, which are approved by the higher F -measures.



Fig. 4. Evolution of the hierarchical segmentation for the image from Fig. 1. Note the coarse-to-fine nature of the evolution: First the broad parts of the image (water and sky) are segmented, while the finer details of the surfer arise later.

Finally, Fig. 4 gives an example of how the segmentation based on the SDP relaxation evolves hierarchically. In this context, note that although the water contains many patches (cf. Fig. 2), it is not split into more segments since the corresponding cut-values are too large.

6 Conclusion

We presented a hierarchical approach to unsupervised image segmentation which is based on a semidefinite relaxation of a constrained binary graph cut problem.

To prevent large homogeneous regions from being split (a common problem of balanced graph cut methods) we computed an over-segmentation of the image in a preprocessing step using the mean shift technique. Besides yielding better segmentations, this also reduced the problem size by several orders of magnitude.

The results illustrate an important advantage of the SDP relaxation in comparison to other segmentation methods based on graph cuts: As the balancing constraint can be adjusted to the current problem, we can appropriately take into account the different size of image patches. Moreover, it is easy to include additional constraints to model other conditions on the image patches, like connections to enforce the membership of certain patches to the same segment. We will investigate this aspect of semi-supervised segmentation in our future work.

References

1. S.-C. Zhu. Statistical modeling and conceptualization of visual patterns. *IEEE Trans. Patt. Anal. Mach. Intell.*, 25(6):691–712, 2003.
2. B. van Cutsem, editor. *Classification and Dissimilarity Analysis*, volume 93 of *Lecture Notes in Statistics*. Springer, 1994.
3. U. Hahn and M. Ramscar, editors. *Similarity and Categorization*. Oxford Univ. Press, 2001.
4. T. Hofmann and J. Buhmann. Pairwise data clustering by deterministic annealing. *IEEE Trans. Patt. Anal. Mach. Intell.*, 19(1):1–14, 1997.
5. J. Puzicha and J. M. Buhmann. Multiscale annealing for unsupervised image segmentation. *Comp. Vision and Image Underst.*, 76(3):213–230, 1999.
6. B. Mohar and S. Poljak. Eigenvalues in combinatorial optimization. In *Combinatorial and Graph-Theoretical Problems in Linear Algebra*, volume 50 of *IMA Vol. Math. Appl.*, pages 107–151. Springer, 1993.
7. J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Patt. Anal. Mach. Intell.*, 22(8):888–905, 2000.
8. J. Keuchel, C. Schnörr, C. Schellewald, and D. Cremers. Binary partitioning, perceptual grouping, and restoration with semidefinite programming. *IEEE Trans. Patt. Anal. Mach. Intell.*, 25(11):1364–1379, 2003.
9. D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int. Conf. Computer Vision (ICCV)*, volume 2, pages 416–423. IEEE Comp. Soc., 2001.
10. D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Patt. Anal. Mach. Intell.*, 24(5):603–619, 2002.
11. X. Ren and J. Malik. Learning a classification model for segmentation. In *Proc. 9th Int. Conf. Computer Vision (ICCV)*, pages 10–17. IEEE Comp. Soc., 2003.
12. H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors. *Handbook of Semidefinite Programming*, volume 27 of *International series in operations research & management science*. Kluwer Acad. Publ., Boston, 2000.
13. M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, 1995.
14. J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. *Int. J. Comp. Vision*, 43(1):7–27, 2001.

15. C. J. Alpert, A. B. Kahng, and S.-Z. Yao. Spectral partitioning with multiple eigenvectors. *Discrete Applied Math.*, 90:3–26, 1999.
16. S. X. Yu and J. Shi. Multiclass spectral clustering. In *Proc. 9th Int. Conf. Computer Vision (ICCV)*, pages 313–319. IEEE Comp. Soc., 2003.
17. Charless Fowlkes, Serge Belongie, Fan Chung, and Jitendra Malik. Spectral grouping using the Nyström method. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(2):214–225, 2004.
18. J. Keuchel and C. Schnörr. Efficient graph cuts for unsupervised image segmentation using probabilistic sampling and SVD-based approximation. In *3rd Internat. Workshop on Statist. and Comput. Theories of Vision*, Nice, France, 2003.
19. D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. Patt. Anal. Mach. Intell.*, 26(5):530–549, 2004.