

An Adaptive Subdivision Scheme for Quadratic Programming in Multi-Label Image Segmentation

Marko Rak
rak@isg.cs.ovgu.de

Tim König
tim.koenig@isg.cs.ovgu.de

Klaus-Dietz Tönnies
toennies@isg.cs.ovgu.de

Department of Simulation and Graphics
University of Magdeburg, Germany

Abstract

We address the problem of efficient, globally optimal multi-label image segmentation. Transforming the discrete labeling problem into a $[0, 1]$ -relaxed binary quadratic program we are able to solve arbitrary convex quadratic labeling tasks in polynomial time. Although this guarantees efficiency in a theoretical sense, large-scale quadratic programs that arise from relaxation can rarely be used for image segmentation directly. We treat this issue by an adaptive domain subdivision scheme, reducing the problem to a short sequence of spatially smoothed medium-scale programs, which subsequently better approximate the large-scale program. Our scheme is globally optimal in terms of the approximated problem. It allows for near-interactive multi-label segmentation and is highly accurate even in the presence of strong noise.

1 Introduction

Many image segmentation techniques are based on convex energy formulations over a discrete set of labels. In this context, a wide range of information sources, such as edge [2, 7], intensity [3, 19], texture [13, 18] and shape [4, 17] can be used for segmentation. Prominent optimization techniques are, for example, level sets and graph cuts in continuous and discrete domains, respectively. We concentrate on the latter kind of domain. Provided the energy formulation is submodular and we deal with the two-label case, i.e., a segmentation into foreground and background, graph cuts guarantee efficient estimation of the globally optimal solution.

The multi-label case is more challenging, even NP-hard in the general setting. Therefore, fast "solutions" are limited to local optimality, e.g., α -expansion by Boykov et al. [1] and the more recent FastPD by Komodakis et al. [9]. For a comprehensive introduction to the topic we refer to the recent survey by Nieuwenhuis et al. [12]. Circumventing the hardness, several approaches relax the labeling integrality condition, resulting in linear, e.g., in Kleinberg and Tardos [8] and in Werner [20], or quadratic programs, e.g., in Dalmau-Cedeno et al. [5] and in Rivera et al. [15, 16]. Although all relaxations can be solved in

polynomial time, quadratic programs seem to be preferable, because they involve less variables and tend to provide tighter relaxations, cf. Ravikumar and Lafferty [L4]. Therefore, we focus on quadratic relaxation; yet, our approach applies to linear relaxations as well.

Although being efficiently solvable from a theoretical point of view, a direct solution of the large-scale $[0, 1]$ -relaxed binary quadratic programs that arise from typical multi-label segmentation tasks is often impractical due to either time or space constraints, or both. We treat this issue by an adaptive domain subdivision scheme, (re-)constructing a sequence of spatially smoothed medium-scale quadratic programs which subsequently better approximate the large-scale program (Section 2). Though our work inherently applies to arbitrary convex quadratic energies, we restrict ourselves – for the sake of simplicity – to minimization of the piecewise constant Mumford-Shah functional. We demonstrate the quality and performance of our labeling technique on noisy synthetic image data as well as on standard images from the Berkeley Segmentation Dataset [B1] (Section 3).

2 Quadratic Programming

In image segmentation, we essentially try to find a labeling that most likely explains the image data. This task typically involves at least two competitive sources of information: a data and a regularization term, which describe the image fidelity and labeling likelihood, respectively. One popular expression of this kind is the piecewise constant Mumford-Shah functional [B1]

$$E = \sum_i \alpha_i \int_{\Omega_i} (u - u_i)^2 dx + \gamma |\Gamma|. \quad (1)$$

Here, we seek for a decomposition of the continuous image domain into mutually exclusive labeling segments Ω_i , which trade off the length of decomposition boundary Γ and the approximation of image intensities u by known reference intensities u_i for each label i . Non-negative weights α_i and γ express the individual importance of the competitive terms.

Let \mathcal{P} be the set of domain elements, e.g. pixel, voxel and so forth, and let \mathcal{N} be the set of all neighborhood relations. Let further $x_i^j \in \{0, 1\}$ be the binary variable that indicates whether element j belongs to label i or not. Provided mutual label exclusiveness, i.e., $\forall j: \sum_i x_i^j = 1$, an equivalent formulation of the continuous Mumford-Shah functional (1) for discrete domains is

$$E = \sum_{j \in \mathcal{P}} \sum_i \alpha_i (u - u_i)^2 x_i^j + \gamma \sum_{(j,k) \in \mathcal{N}} \sum_i (x_i^j - x_i^k)^2. \quad (2)$$

2.1 Problem Relaxation

The minimization of general convex quadratic labeling energies is hard due to the binarity constraint on the variables. Therefore, we relax the problem into $x_i^j \in [0, 1]$, while keeping the rest of the problem fixed. The relaxed variables may be interpreted as certainty measure for the labeling decision, where near-binary solutions express higher confidence. Suppose we are given a soft solution to the relaxed problem then the hard labeling can be obtained by "thresholding", i.e., taking the most certain label for each domain element.

A solution to the relaxed problem lies in the intersection of the $[0, 1]$ -hypercube with the affine hyperplanes $\sum_i x_i^j = 1$, thus in a convex set. For convenience, we assemble all variables x_i^j label-wise into a vector $\mathbf{x} = (x_1^1 x_1^2 \dots x_1^{p-1} x_1^p)^T$ with p and l denoting the number of

domain elements and labels, respectively. As with any quadratic labeling energy, the discrete Mumford-Shah functional (2) readily transforms into standard form

$$\mathbf{x}^T \mathbf{H} \mathbf{x} + \mathbf{f}^T \mathbf{x} \rightarrow \min! \quad (3)$$

$$\mathbf{A} \mathbf{x} = \mathbf{1} \quad (4)$$

$$\mathbf{0} \preceq \mathbf{x} \preceq \mathbf{1}. \quad (5)$$

Here, matrix $\mathbf{H} = \text{diag}(\mathbf{H}', \dots, \mathbf{H}')$ contains l symmetric submatrices $\mathbf{H}' \in \mathbb{R}^{p \times p}$ along its blockdiagonal, which subsume the neighborhood relations of Eq. (2) label by label. Vector $\mathbf{f} \in \mathbb{R}^{pl \times 1}$ does the same for the intensity approximation terms of Eq. (2). Element-wise label affinity is ensured by Eq. (4), where matrix $\mathbf{A} = [\mathbf{I} \dots \mathbf{I}]$ is a concatenation of l identities $\mathbf{I} \in \mathbb{R}^{p \times p}$ and $\mathbf{1} \in \mathbb{R}^{p \times 1}$ is an appropriately sized one-vector. $[0, 1]$ -bounds on variables are imposed in (5) by the pointwise operator " \preceq ", comparing \mathbf{x} with appropriately sized vectors $\mathbf{0}, \mathbf{1} \in \mathbb{R}^{pl \times 1}$.

This sparse quadratic program (QP) can be solved in polynomial time as long as \mathbf{H} is symmetric positive (semi-)definite and the feasibility region is convex, cf. Kozlov et al. [14]. Although this is efficient from a theoretical perspective, the large number of variables, i.e., $p \cdot l$, render its direct solution impractical for typical segmentation tasks due to time or space constraints, or both.

2.2 Affine Variable Reduction

Before addressing this issue via adaptive subdivision, we first apply a simple attack to the problem, reducing the number of variables by p . Note that this is not crucial to the subdivision scheme described hereafter; still, it improved the performance in all our experiments considerably.

Since one variable per domain element is determined by affinity, i.e., w.l.o.g. the l -th variable by $x_l^j = 1 - \sum_{i=1}^{l-1} x_i^j$, we can easily set up matrix $\mathbf{R} \in \mathbb{R}^{pl \times p(l-1)}$ and vector $\mathbf{r} \in \mathbb{R}^{pl \times 1}$ such that

$$\mathbf{x} = [\mathbf{R} \quad \mathbf{r}] \begin{pmatrix} \mathbf{y} \\ \mathbf{1} \end{pmatrix} \quad (6)$$

reduces the problem to variables $\mathbf{y} = (x_1^1 x_1^2 \dots x_{l-1}^{p-1} x_{l-1}^p)^T$ and the constant "1". Now, $\mathbf{A} \mathbf{x} = \mathbf{1}$ in Eq. (4), which enforces affinity in the initial QP, is implicitly satisfied if the remaining variables satisfy $\sum_{i=1}^{l-1} x_i^j \leq 1$ for each domain element j . Hence, we remove the last p columns of \mathbf{A} to create matrix $\mathbf{B} \in \mathbb{R}^{p \times p(l-1)}$, which by $\mathbf{B} \mathbf{y} \preceq \mathbf{1}$ reflects all such constraints.

Applying the affine variable reduction (6) to the initial QP (3 – 5) and replacing equality conditions (4) as mentioned above, we obtain an intermediate QP. This can be further simplified, eliminating the constant variable "1" by removal of the corresponding entries in quadratic and linear terms, cf. Gould and Toint [6]. Hence, we are left with $\mathbf{G} = \mathbf{R}^T \mathbf{H} \mathbf{R}$ and $\mathbf{e}^T = \mathbf{f}^T \mathbf{R}$, framed by the affinely reduced QP

$$\mathbf{y}^T \mathbf{G} \mathbf{y} + \mathbf{e}^T \mathbf{y} \rightarrow \min! \quad (7)$$

$$\mathbf{B} \mathbf{y} \preceq \mathbf{1} \quad (8)$$

$$\mathbf{0} \preceq \mathbf{y} \preceq \mathbf{1}. \quad (9)$$

For the special two-label case this actually halves the number of variables, merely leaving us with the x_1^j . Therefore, inequalities in (8) are redundant in this case and can be dropped, because \mathbf{B} is the identity matrix and just expresses $\mathbf{y} \preceq \mathbf{1}$.

2.3 Adaptive Subdivision

Although reduced, this is still a large-scale program in $p \cdot (l - 1)$ variables. To resolve this issue, we propose to approximate the QP (7 – 9) by a short sequence of medium-scale QPs, stemming from adaptive subdivision refinement steps. Thus, we gradually improve the approximation quality, while maintaining a comparably small problem.

QP Oracle Assume that an oracle tells us which domain elements are near a border of the optimal labeling and which are not, i.e., elements inside some labeling segment. Since in the latter case the QP consistently favors some particular label, it will do so for a locally smoothed version of our objective function (7) as well. Near to borders of the labeling, such a spatial smoothing should fade out to zero to keep fine details. A simple hierarchical subdivision via quadtree, octree and their generalization to higher dimensions can be used to implement this property.¹ To keep dimension-independent terminology, we refer to these as "subdivision trees" or simply as "trees".

Using the oracle information, we recursively subdivide the domain unless a region neither contains nor touches a border or itself is a domain element. We can express the labeling problem in terms of these subregions, i.e., leaves in the subdivision tree. Therefore, we define $l - 1$ label variables, termed "representatives", for each of the r regions. We assemble these representatives *label-wise* into a vector $\mathbf{z} \in \mathbb{R}^{r(l-1) \times 1}$ and relate both problem spaces, i.e., the large-scale element basis \mathbf{y} and the regional basis \mathbf{z} , by

$$\mathbf{y} = \mathbf{P}\mathbf{z}. \quad (10)$$

Here, matrix $\mathbf{P} = \text{diag}(\mathbf{P}', \dots, \mathbf{P}')$ contains $l - 1$ submatrices $\mathbf{P}' \in \mathbb{R}^{p \times r}$ along its block-diagonal, which project values of \mathbf{z} onto \mathbf{y} . More precisely, \mathbf{P}' superimposes the regional basis in \mathbf{z} , i.e., the representatives, to the element basis in \mathbf{y} label by label according to the subdivision. Applying the superimposition (10) to the large-scale QP (7 – 9) gives

$$\mathbf{z}^T \mathbf{P}^T \mathbf{G} \mathbf{P} \mathbf{z} + \mathbf{e}^T \mathbf{P} \mathbf{z} \rightarrow \min! \quad (11)$$

$$\mathbf{B} \mathbf{P} \mathbf{z} \preceq \mathbf{1}^2 \quad (12)$$

$$\mathbf{0} \preceq \mathbf{z} \preceq \mathbf{1}^3, \quad (13)$$

where the condensation by $\mathbf{P}^T \mathbf{G} \mathbf{P}$ and $\mathbf{e}^T \mathbf{P}$ renders an adaptively smoothed objective function in the regional basis \mathbf{z} . Unless the mild assumption of spatial labeling coherence is violated, e.g., if we deal with a random problem, this QP is of medium scale, because the number of regions is typically much smaller than the number of domain elements, i.e., $r \ll p$. This is especially true for higher-dimensional problems.

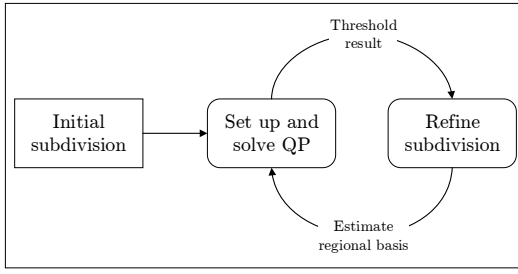
QP Sequence Dropping the aforementioned oracle assumption, we have to (re-)construct the subdivision tree alongside with the medium-scale QP (11 – 13). This can be achieved by short sequence $(0, 1, \dots, i, \dots, c)$ of representatives \mathbf{z}_i , stemming from $c + 1$ subdivision refinement steps. Similar to role of \mathbf{z} in the superimposition (10), these representatives superimpose their regional basis to the element basis \mathbf{y} by

$$\mathbf{y} = \mathbf{P}_i \mathbf{z}_i. \quad (14)$$

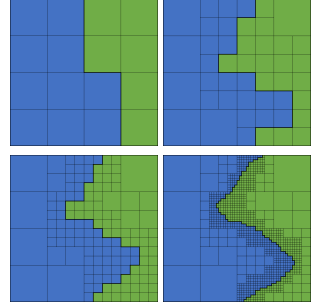
¹Note that quadtrees, octrees and so forth can be defined over arbitrarily sized domains.

²Matrix $\mathbf{B} \mathbf{P}$ contains duplicated conditions in its rows, which can be removed easily.

³Applying (10) to (9) formally gives $\mathbf{0} \preceq \mathbf{P} \mathbf{z} \preceq \mathbf{1}$, which due to \mathbf{P} expresses no more than $\mathbf{0} \preceq \mathbf{z} \preceq \mathbf{1}$.



(a) Refinement procedure outline



(b) Subsequent refinement steps

Figure 1: Overview of our adaptive subdivision scheme. (a) Starting from an initial domain subdivision, we alternately solve the current QP based on the regional subdivision basis and refine the subdivision near to labeling borders of the "thresholded" result afterwards. (b) "Thresholded" labelings for a two-label problem in subsequent refinement steps (row-wise).

Without knowing any labeling border, we initially decompose the image domain up to a preset tree level c above ground, i.e., above the 0-level where regions equal domain elements. As outlined in Figure 1(a), we then alternate $c + 1$ times between two steps. In the first step, we set up and solve the current approximate QP

$$\mathbf{z}_i^T \mathbf{P}_i^T \mathbf{G} \mathbf{P}_i \mathbf{z}_i + \mathbf{e}^T \mathbf{P}_i \mathbf{z}_i \rightarrow \min! \quad (15)$$

$$\mathbf{B} \mathbf{P}_i \mathbf{z}_i \preceq \mathbf{1}^4 \quad (16)$$

$$\mathbf{0} \preceq \mathbf{z}_i \preceq \mathbf{1}^5, \quad (17)$$

based on the tree leaves known so far. The solution is then transformed from regional to element basis via Eq. (14) and "thresholded" to a hard labeling, taking the most certain label for each domain element. In the second step, leaves of the tree are subdivided if they touch a border of the labeling. The refined subdivision tree is passed into the next alternation, allowing us to set up the refined regional basis. If $c + 1$ is reached, we omit this second step. Subsequent labelings of the adaptive refinement process are depicted in Figure 1(b).

As the alternation elapses, the number of regions r_i increases, which improves the problem approximation. Due to the adaptivity of the subdivision refinement, we keep the medium-scale property, i.e., $r_i \ll p$ for each QP (15 – 17) of the sequence, unless the mild underlying assumption of spatial labeling coherence is violated. Fully exploiting this property, we construct $\mathbf{P}_i^T \mathbf{G} \mathbf{P}_i$, $\mathbf{e}^T \mathbf{P}_i$ and $\mathbf{B} \mathbf{P}_i$ directly without ever setting up the large-scale quantities \mathbf{G} , \mathbf{e} or \mathbf{B} . This way, we are able to keep space requirements at its lowest while solving the labeling task.

The only parameter of our adaptive scheme is c , the initial tree level. Side lengths of regions in the initial subdivision are related to c by 2^c . Note that the special case $c = 0$ exactly defines the large-scale QP (7 – 9), because \mathbf{P}_0 is (up to permutation of rows) an identity matrix in this case. Labeling segments smaller than 2^c may not be recovered by our scheme, due to the objective function smoothing. For most tasks this is not an issue, because lower bounds on segment extents are known beforehand and c can be set accordingly. Often, this is even an advantage, removing spurious segments from the labeling.

⁴See Footnote 2.

⁵See Footnote 3.

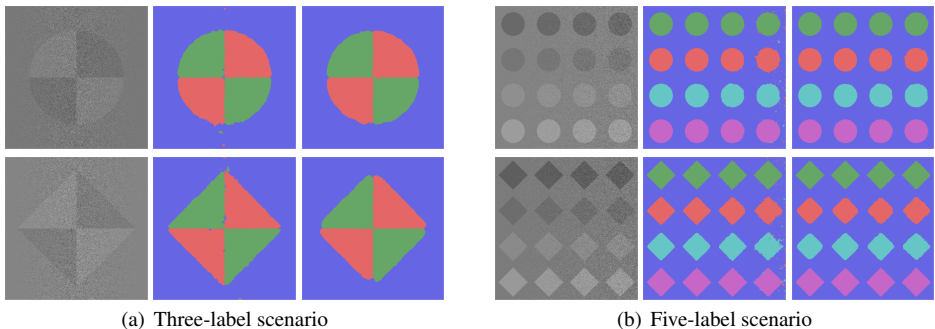


Figure 2: Comparison of our adaptive subdivision scheme to the large-scale quadratic program on synthetic image data. First columns: noisy input images, second columns: labeling of the large-scale QP, third columns: labeling of our adaptive subdivision scheme for $c = 7$

3 Experiments and Discussion

We implemented our approach in Matlab using SuiteSparse for handling of sparse matrices. We tested several solver packages for sparse QPs and found Matlab’s Optimization Toolbox to perform best. IBM’s CPLEX package showed similar performance in most cases. All experiments were realized on a standard 64-bit system with an Intel Core i5 4×3.30 GHz and 12 GB of main memory.

3.1 Synthetic Scenarios

To demonstrate the quality and performance of our approach we implemented two scenarios, which both correspond to the piecewise constant Mumford-Shah assumption. We used these scenarios for demonstration only as our approach is not limited to this simple labeling functional. As shown in Figure 2, each scenario comprises two noisy 512×512 instances with known ground truth.

The first scenario is a three-label problem with one background and two foreground labels, having intensities of 0.5, 0 and 1, respectively. In both instances, zero mean Gaussian noise linearly increases from vertical image borders to the center, having a peak variance of 0.4. This results in a signal-to-noise ratio near to one at central spots. As second scenario, we implemented a five-label problem with one background and four foreground labels. Background intensity is 0.5 and foreground labels follow intensities of 0, 0.25, 0.75 and 1, respectively. Akin to the former scenario, zero mean Gaussian noise varies spatially for both instances. This time noise increases linearly from left to right with a peak variance of 0.2, resulting in a signal-to-noise ratio similar to the former scenario.

We parameterized reference intensities of the Mumford-Shah functional optimally, i.e., according to the label intensities that were used to set up the problem instances. Regarding relative importance of intensity approximation and boundary length, we found weights $\forall i : \alpha_i = 0.995$ and $\gamma = 0.005$ to perform well independent of the scenario or problem instance. We benchmarked our approach for subdivision levels ranging from $c = 7$ down to $c = 0$, whereby the latter means solving the large-scale QP (7 – 9) directly without adaptive subdivision. Performance was assessed by run time measurements, while quality was measured by \mathcal{L}_1 error w.r.t. the ground truth.

3.2 Benchmark Results

Regarding the overall labeling quality, our approach performed reasonably well even under such noisy conditions. As shown in Table 1, \mathcal{L}_1 errors are below 1.5 % in every problem instance independent of the initial subdivision level. Although intra-scenario differences are small, we still observe that circular problem instances are recognized slightly better than those containing diamonds, which is most likely due to the oversmoothing of corners. Increasing subdivision levels, i.e., solving for $c > 0$, reduced the labeling error in the five-label scenario. The reason is that the adaptive smoothing of the objective function helped to reduce spurious segments, as can be seen comparing results for $c = 0$ and $c = 7$ in Figure 2(b). No such trend can be identified in the three-label scenario. Although a removal of spurious segments occurs here as well, the effect is outweighed by the oversmoothing of corners for $c > 1$. Results for $c = 0$ and $c = 7$ are shown in Figure 2(a). The results imply that our approach is stable in the presence of strong noise and even introduces valuable regularization for $c > 0$.

Table 1: \mathcal{L}_1 error for synthetic image data with respect to ground truth in percent.

| Scenario | Instance | Initial subdivision level c | | | | | | | |
|-------------|----------|-------------------------------|-------------|-------------|------|------|------|------|------|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| three-label | circle | 0.88 | 0.71 | 0.73 | 0.76 | 0.76 | 0.76 | 0.76 | 0.76 |
| | diamond | 1.03 | 0.89 | 1.12 | 1.34 | 1.41 | 1.41 | 1.41 | 1.41 |
| five-label | circles | 1.11 | 0.54 | 0.53 | 0.54 | 0.54 | 0.54 | 0.54 | 0.54 |
| | diamonds | 1.37 | 0.80 | 0.80 | 0.82 | 0.83 | 0.83 | 0.99 | 0.99 |

Concerning performance measurements, we observe a significant decrease in run time as c increases. As shown in Table 2, the peak speed-up factor w.r.t. the large-scale QP is between eight and nine for instances of the three-label scenario, while it is between four and five for the five-label scenario. As could be expected, if the initial subdivision level is increased too far, i.e., $c > 5$ for the three-label scenario, the computational cost slightly rises again. Still, even for $c = 7$ speed-up factors lay between four and five for all problem instances. This implies that starting with an initial subdivision level that is "too coarse" is not crucial to the performance of our approach. Another expectation is confirmed when comparing inter-scenario performance. Here, run time measurements for the five-label scenario are significantly higher. The difference in performance is due to the increased number of labels and due to the difference in lengths of labeling borders, which are substantially longer in the five-label scenario. Still, the performance measurements show suitability of our approach for near-interactive image segmentation, even in this demanding scenario.

Table 2: Run time measurements on synthetic image data in seconds.

| Scenario | Instance | Initial subdivision level c | | | | | | | |
|-------------|----------|-------------------------------|------|------|------|------|-------------|------|-------------|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| three-label | circle | 8.37 | 8.93 | 7.08 | 2.78 | 1.06 | 0.92 | 1.67 | 1.74 |
| | diamond | 8.51 | 9.04 | 7.09 | 2.92 | 1.25 | 1.05 | 1.98 | 2.10 |
| five-label | circles | 28.7 | 24.3 | 14.0 | 5.58 | 5.87 | 6.39 | 5.95 | 5.49 |
| | diamonds | 28.9 | 23.8 | 13.7 | 6.47 | 6.04 | 5.78 | 5.82 | 5.88 |

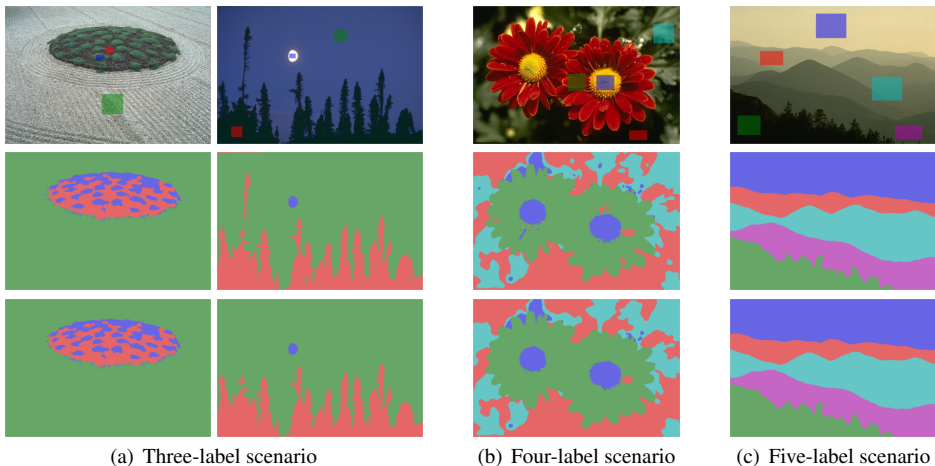


Figure 3: Comparison of our adaptive subdivision scheme to the large-scale quadratic program on standard image data. First row: input images, second row: labeling of the large-scale QP, third row: labeling of our adaptive subdivision scheme for $c = 5$

3.3 Standard Images

To illustrate its effectiveness on real-world problems, we applied our approach to a number of images from the Berkeley Segmentation Dataset [14], each sized 481×321 pixels. As shown in Figure 3, our first two images can be interpreted as three-label scenario instances, while the other two form four and five-label problems, respectively. As with the synthetic image data, we restricted ourselves to images that roughly correspond to the Mumford-Shah functional (1), which we extended from intensity to color using (squared) Euclidean distances in RGB color space. Reference color vectors were obtained directly from the image data by averaging over the rectangular regions marked in Figure 3.

Reviewing the results, we observe semantically reasonable labelings in all cases, independent of the initial subdivision level c . Comparing results among subdivision levels, only the "moonlight" instance in Figure 3(a) shows a semantically meaningful alteration of the labeling for $c > 0$. Therein, thin parts of the large tree gradually smooth out as the initial subdivision level increases, leading to the complete removal of the treetop for $c = 5$. Besides that, the "flower" instance in Figure 3(b) shows minor differences with little semantic alteration. For example, spurious segments as the less lit petal parts, which were labeled incorrectly by the large-scale QP, smooth out when adaptive subdivision is applied. The other problem instances, namely the "rock bed" in Figure 3(a) and the "mountain" instance in Figure 3(b), do not show any noticeable differences among subdivision levels. The results imply valuable regularization w.r.t. spurious segments for $c > 0$. Certainly, this has the downside of smoothing out small, yet meaningful segments too. As mentioned earlier, this is not a problem in most cases, since lower bounds on segment extents are typically known beforehand and c can be set accordingly.

Run time measurements for all problem instances are given in Table 3. Akin to the case of synthetic image data, we notice that our approach significantly speeds up the computation throughout all instances. For the "rock bed" instance we observe a peak speed-up of about seven, while it is only around two for the "flower" instance. The difference in complexity

(border length) of both labelings is most likely the cause of this gap. The other instances lie midway between the extrema, having peak factors between three and four. If we increase the initial subdivision level beyond its optimum, e.g., $c > 3$ for the four and five-label scenario, the computational cost rises again. Still, the increase in run time is minor, reducing the "rock bed" speed-up to about five, while factors for the other instances remain almost unchanged. This again suggests that choosing a level that is "too coarse" is not crucial to the performance of our approach.

Table 3: Run time measurements on standard image data in seconds.

| Scenario | Instance | Initial subdivision level c | | | | | |
|-------------|-----------|-------------------------------|-------|------|-------------|-------------|------|
| | | 0 | 1 | 2 | 3 | 4 | 5 |
| three-label | rock bed | 6.29 | 3.75 | 3.29 | 1.59 | 0.91 | 1.31 |
| | moonlight | 5.76 | 3.76 | 2.94 | 1.71 | 1.62 | 1.78 |
| four-label | flowers | 13.31 | 7.99 | 8.11 | 6.68 | 7.26 | 7.16 |
| five-label | mountains | 16.65 | 11.12 | 6.40 | 4.47 | 5.01 | 5.20 |

4 Conclusion

In this work, we focussed on multi-label image segmentation by convex quadratic energy formulations over discrete domains. At the example of the piecewise constant Mumford-Shah functional, we transformed the labeling problem into its binary quadratic programming equivalent. Since solving the problem is NP-hard in the general setting, we relaxed the binary assumption, transforming the problem into a $[0, 1]$ -relaxed binary quadratic program. Although the relaxed problem can be solved in polynomial time, the large number of variables renders its direct solution impractical due to either space or time constraints, or both.

Addressing this issue, we showed how to approximate the large-scale quadratic program by a short sequence of medium-scale programs, subsequently increasing the approximation quality. Each program originates in a simple, yet powerful refinement step that adaptively subdivides the image domain according to the labeling border obtained from the preceding program. The only parameter of our scheme is the initial subdivision level, which directly corresponds to the length of the sequence. Looking at it from another perspective, our refinement scheme subsequently (re-)constructs a medium-scale quadratic program that is an adaptively smoothed approximation to the actual large-scale program. The initial subdivision level hence defines the amount of smoothing that is initially applied to the problem.

Experiments on synthetic image data with strong noise show that our scheme yields high quality image labelings widely independent of the initial smoothing. The smoothing even introduces additional regularization, preventing the emergence of spurious segments. Comparing performance to the large-scale program, our scheme consistently speeds up the computation by a factor of, at least, four to five, while keeping space requirements at its lowest. Essentially the same holds true for standard image data. Depending on the complexity of the multi-label problem, our scheme outperformed the large-scale program by factors ranging between two and seven, with the labeling being widely independent of the initial smoothing. The gain in performance even rendered relaxed binary quadratic programming near-interactive in our experiments on medium-sized, multi-label segmentation problems.

In summary, we proposed a spatially adaptive multi-label segmentation scheme that is efficient and globally optimal in terms of the approximated problem. Our approach applies

to problems of arbitrary dimension and allows for general convex quadratic energy terms, e.g., incorporating priors for edge, intensity, texture and shape information. Boundary conditions, such as the label affiliation of domain subregions, which are often used in interactive segmentation, can be implemented easily by supplying additional constraints to the program.

We plan to migrate our approach from general purpose quadratic programming solvers to schemes that are explicitly tailored to multi-label image segmentation. Exploiting the repetitive structure of our system matrices and the simple nature of our constraints, we should be able to provide highly efficient adaptive solvers for large-scale, multi-label image segmentation tasks. Furthermore, we seek to transition from discrete to continuous problem domains, making adaptive relaxed binary quadratic programming a relevant option there as well.

Acknowledgments

This research was partially funded by the German Research Foundation (TO 166/13-2) and the research initiative STIMULATE of the German Ministry of Education and Research.

References

- [1] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:1222–1239, 2001.
- [2] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22:61–79, 1997.
- [3] T. F. Chan and L. A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10:266–277, 2001.
- [4] D. Cremers, S. J. Osher, and S. Soatto. Kernel density estimation and intrinsic alignment for shape priors in level set segmentation. *International Journal of Computer Vision*, 69:335–351, 2006.
- [5] O. Dalmau-Cedeno, M. Rivera, and P. P. Mayorga. Computing the alpha-channel with probabilistic segmentation for image colorization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1–7, 2007.
- [6] N. Gould and P. L. Toint. Preprocessing for quadratic programming. *Mathematical Programming*, 100:95–132, 2004.
- [7] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: active contour models. *International Journal of Computer Vision*, 1:321–331, 1988.
- [8] J. Kleinberg and É. Tardos. Approximation algorithms for classification problems with pairwise relationships: metric labeling and Markov random fields. *Journal of the ACM*, 49:616–639, 2002.
- [9] N. Komodakis, G. Tziritas, and N. Paragios. Fast, approximately optimal solutions for single and dynamic MRFs. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

- [10] M. K. Kozlov, S. P. Tarasov, and L. G. Khachiyan. The polynomial solvability of convex quadratic programming. *USSR Computational Mathematics and Mathematical Physics*, 20:223–228, 1980.
- [11] 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 *Proceedings of the IEEE International Conference on Computer Vision*, pages 416–423, 2001.
- [12] C. Nieuwenhuis, E. Töppe, and D. Cremers. A survey and comparison of discrete and continuous multi-label optimization approaches for the Potts model. *International Journal of Computer Vision*, pages 1–18, 2013.
- [13] N. Paragios and R. Deriche. Geodesic active regions and level set methods for supervised texture segmentation. *International Journal of Computer Vision*, 46:223–247, 2002.
- [14] P. Ravikumar and J. Lafferty. Quadratic programming relaxations for metric labeling and Markov random field MAP estimation. In *Proceedings of the International Conference on Machine Learning*, pages 737–744, 2006.
- [15] M. Rivera, O. Ocegueda, and J. L. Marroquin. Entropy-controlled quadratic Markov measure field models for efficient image segmentation. *IEEE Transactions on Image Processing*, 16:3047–3057, 2007.
- [16] M. Rivera, O. Dalmau-Cedeno, and J. Tago. Image segmentation by convex quadratic programming. In *Proceedings of the International Conference on Pattern Recognition*, pages 1–5, 2008.
- [17] M. Rousson and N. Paragios. Shape priors for level set representations. In *Proceedings of the European Conference on Computer Vision*, pages 78–92, 2002.
- [18] M. Rousson, T. Brox, and R. Deriche. Active unsupervised texture segmentation on a diffusion based feature space. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 699–704, 2003.
- [19] A. Tsai, A. Jr. Yezzi, and A. S. Willsky. Curve evolution implementation of the Mumford-Shah functional for image segmentation, denoising, interpolation, and magnification. *IEEE Transactions on Image Processing*, 10:1169–1186, 2001.
- [20] T. Werner. A linear programming approach to max-sum problem: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:1165–1179, 2007.