

EMMCVPR 2011, St. Petersburg

Curvature Regularity for Multi-Label Problems - Standard and Customized Linear Programming

Thomas Schoenemann, Yubin Kuang and Fredrik Kahl

Centre for Mathematical Sciences
Lund University, Sweden

Abstract. We follow recent work by Schoenemann et al. [25] for expressing curvature regularity as a linear program. While the original formulation focused on binary segmentation, we address several multi-label problems, including segmentation, denoising and inpainting, all cast as a single linear program.

Our multi-label segmentation introduces a “curvature Potts model” and combines a well-known Potts model relaxation [14] with the above work. For inpainting, we improve on [25] by grouping intensities into bins. Finally, we address the problem of denoising with absolute differences in the data term.

Furthermore, we explore alternative solving strategies, including higher order Markov Random Fields, min-sum diffusion and a combination of augmented Lagrangians and an accelerated first order scheme to solve the linear programs.

1 Introduction

Labeling problems¹ with length-based regularity terms for computer vision have received an enormous amount of attention over the past two decades. Many of the associated optimization problems are solved globally today, e.g. [12,18] using the length of segmentation boundaries as a regularizer. If, more generally, one considers the length of level lines the result is the well-known total variation which is very wide-spread in image denoising [23,7] and again can be optimized globally.

In many problems, one also wants to include second order regularity terms based on curvature. This has turned out to be much more challenging. Here, in a region-based context only the inpainting approach of Masnou and Morel [15] finds global optima.

For a long time, region-based problems were addressed by local curve evolution, e.g. for segmentation in [19,11], for inpainting in [8,29,5] and very recently for denoising in [33,6]. In particular for segmentation problems this requires a good initialization and more generally, it can lead to numerical instabilities.

¹ This work was funded by the European Research Council (GlobalVision grant no. 209480).

For purely boundary based problems (without any regional terms), solutions were available already in 1990: Amini et al. [1] gave a solution that computes global optima as long as self-intersecting curves are allowed. More recently, this has been generalized to ratio functionals [24], which in practice is even more efficient. Curvature has also been explored for the problem of trace inference [20].

In a recent work, Schoenemann et al. [25,26] proposed to solve region-based problems with curvature regularity as linear programs. This approach is independent of initialization and has the advantage over the boundary-based methods that region-terms can be included, self-intersections be excluded and that the computed regions need not be (singly-) connected. The authors also show applications to inpainting. Subsequently El-Zehiry and Grady [10] proposed to solve binary segmentation problems with curvature regularity via the QPBO-method [13] with a subsequent probing stage. This works correctly for a 4-connectivity only, but an approximation for an 8-connectivity was included.

Recently, it has also been tried to solve region-based curvature problems via non-convex constrained optimization [28]. While this is different from curve evolution, it is still dependent on initialization.

In this work we consider the *multi*-label problems of multi-region segmentation, inpainting and denoising. For segmentation we derive a “curvature Potts model”, i.e. show that the linear program in [25] can be generalized to more than two regions. Here, we make use of prior work [14,32] on writing the (length-based) Potts model as a very compact integer linear program. Approaches derived from a continuous formulation of the Potts model include [21,31].

For inpainting, we show that the parameterization of [25] may degenerate (and does so quite often in practice) since “level lines” are allowed to cross between different levels. We discuss what is needed to prevent this and then introduce a new multi-label parameterization to remedy the situation that is still practicable (in terms of memory and run-time). Subsequently we address the problem of image denoising with absolute differences in the data term and a level-line based curvature regularizer.

Finally, we discuss alternative techniques to model curvature as a linear program. Starting from [10] we derive a linear program that corresponds to a higher order MRF of a form considered in [30]. We then compare the two formulations and try suboptimal solving strategies such as min-sum diffusion (see [30]) and a (prematurely terminated) convex augmented Lagrangian solver. Previously, customized interior point methods have been investigated [4] for *inequality constrained* linear programs. However, in our setting we have equality constraints and found that the method does not work well on such problems.

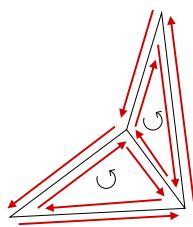
2 Curvature and Linear Programming

In this section, we review the formulation in [25] for region-based binary segmentation problems with curvature regularity (but differ slightly in notation). Given an image $I : \Omega \rightarrow \mathbb{R}$, the problem is to find a division of $\Omega \subset \mathbb{R}^2$

into two regions, foreground and background, where two given functions $g_F(\mathbf{x})$ and $g_B(\mathbf{x})$ (derived from the image I) define the cost for a point \mathbf{x} belonging to foreground and background respectively. It will be convenient to define $g(\mathbf{x}) = g_F(\mathbf{x}) - g_B(\mathbf{x})$. Together with length and curvature regularity, up to a constant the resulting problem is to minimize

$$\min_{R \subseteq \Omega} \int_R g(\hat{\mathbf{x}}) d\hat{\mathbf{x}} + \nu|C| + \lambda \int_C |\kappa_C(\hat{\mathbf{x}})|^2 d\mathcal{H}^1(\hat{\mathbf{x}}) , \quad (1)$$

where $C = \partial R$ is the segmentation boundary of length $|C|$, weighted with $\nu > 0$, κ_C denotes the curvature of the boundary and $\lambda > 0$ is a weighting factor for curvature regularity. The one-dimensional Hausdorff measure $d\mathcal{H}^1$ signifies an integration over a set of closed lines and that the integral is independent of the parameterization of the lines.



One can reformulate (1) as a linear program (LP) by first subdividing Ω into a cell complex, i.e. a subdivision into N non-overlapping *basic regions* whose union gives Ω . Where two regions meet there is an *edge*, and for every edge the approach considers two *line segments*, one for every possible direction of traversal of the edge – see the Figure on the left. We denote the set of basic regions \mathcal{F} , the set of edges \mathcal{E} and the set of line segments \mathcal{E}^o .

In the arising discrete setting, the region integral in (1) is expressed as $\mathbf{g}^T \mathbf{x}$, where the vector $\mathbf{x} \in \{0, 1\}^N$ contains an entry x_f for every basic region $f \in \mathcal{F}$ and with 1 indicating foreground, 0 background. The entries of the vector \mathbf{g} reflect the integral of $g(\cdot)$ over each basic region.

In principle the region variables \mathbf{x} already define a boundary and hence the cost of the regularity terms. However, since this mapping is not linear [25] introduces an additional variable-vector \mathbf{y} with an entry $y_{jj'} \in \{0, 1\}$ for every pair $j, j' \in \mathcal{E}^o$ of consistently oriented adjacent line segments. The variable $y_{jj'}$ is meant to be 1 if and only if both line pairs are part of the boundary.

The regularity term is now expressed as $\mathbf{w}^T \mathbf{y}$, where \mathbf{w} is a vector defining the length and curvature weights for every pair j, j' of consistent line segments. Here curvature is computed in terms of the angle θ between the line segments:

$$w_{jj'} = \frac{1}{2}\nu(l_j + l_{j'}) + \lambda\theta^2 \quad ,$$

where l_j and $l_{j'}$ are the lengths of the two line segments.

To ensure the consistency between boundary and region variables, constraints are needed, and in [26] there are three sets. The first ensures that wherever the foreground region ends there is an appropriate boundary variable. Hence there is one constraint for every edge. To formalize these constraints, all basic regions are assigned a consistent orientation (e.g. clockwise traversal everywhere). The oriented boundary line to be fitted must have the opposite orientation. Finally one needs to assign each edge an (arbitrary but fixed) orientation.

The constraints are now based on the notion of *incidence* of basic regions to edges and of line pairs to edges. The incidence of region f and edge e is denoted

m_e^f (for “match”) and defined to be 0 unless the basic region f contains the edge e in its boundary. Otherwise, m_e^f is 1 if the orientation of edge e agrees with the orientation of the region f and -1 if the orientations disagree.

For edge pairs jj' , the incidence operator $m_e^{jj'}$ is 0 unless line segment j is an orientation of the edge e . In this case it is 1 if e and j agree in orientation, -1 if they disagree. The first set of constraints, called *surface continuation* is now:

$$\sum_f m_e^f x_f + \sum_{j,j'} m_e^{jj'} y_{jj'} = 0 \quad \forall e \in \mathcal{E} . \quad (2)$$

This set alone is not sufficient to ensure consistency of region and boundary variables. As a consequence, [25] introduces the *boundary continuation constraints*, with the aim to ensure that whenever $y_{jj'} = 1$ both affected edges are actually part of the region boundary induced by \mathbf{x} : `vspace-.8mm`

$$\sum_{j'} y_{j'j} = \sum_{j''} y_{jj''} \quad \forall j \in \mathcal{E}^o , \quad (3)$$

i.e. whenever a line segment forms the first part of a pair in the boundary, it must also be the second part for some such pair. Finally, to prevent touching lines one adds the constraint set

$$\sum_{jj'} |m_e^{jj'}| y_{jj'} \leq 1 \quad \forall e \in \mathcal{E} , \quad (4)$$

which is called *boundary consistency*. They express that for every edge e only one line pair in the boundary may contain an orientation of e as its first element. Simultaneous to our work appears the work [27] that proposes an alternative constraint system to prevent touching lines. We leave this for future work.

The resulting integer linear program is

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{y}} \quad & \mathbf{g}^T \mathbf{x} + \mathbf{w}^T \mathbf{y} \\ \text{s.t.} \quad & (2), (3), (4), \quad x_f \in \{0, 1\} \quad \forall f, \quad y_{jj'} \in \{0, 1\} \quad \forall j, j' . \end{aligned} \quad (5)$$

As in [25], we will solve the linear programming relaxation, i.e. the non-convex variable domains $\{0, 1\}$ are relaxed to the convex sets $[0, 1]$ for all variables.

3 Formulations for Multi-Label Problems

Above we have described how to handle curvature in binary segmentation via linear programs, and later on we will review how this can be extended to inpainting. In all cases there is one region variable for each basic region in the mesh. We now introduce formulations for multi-label problems where we have several variables associated to each region.

3.1 Multi-Region Segmentation

The first problem we address is multi-region segmentation, i.e. a generalization of (1) to the region set $\mathcal{I} = \{1, \dots, K\}$ where K is a pre-defined constant.

In the continuous formulation, we are now looking for a segmentation function $s : \Omega \rightarrow \mathcal{I}$ and want to minimize the model

$$\int_{\Omega} g(\hat{\mathbf{x}}, s(\hat{\mathbf{x}})) d\hat{\mathbf{x}} + \frac{1}{2} \sum_{i=1}^K [\nu |C_i| + \lambda \int_{C_i} |\kappa_{C_i}(\hat{\mathbf{x}})|^2 d\mathcal{H}^1(\hat{\mathbf{x}})] \quad (6)$$

where $g(\hat{\mathbf{x}}, i)$ defines a data term for point $\hat{\mathbf{x}}$ belonging to segment i , $C_i = \partial\{\hat{\mathbf{x}} \mid s(\hat{\mathbf{x}}) = i\}$ is the set of points where $s(\cdot)$ is discontinuous and on an adjacent side takes the value i and $|C_i|$ denotes the one-dimensional measure of this set of points. Note that (6) exploits the known observation [14,32] that every point in the discontinuity set of $s(\cdot)$ belongs to *two* region boundaries, hence the factor $1/2$ for the regularity term.

Here we are assuming that the region boundary of every region is a smooth curve, i.e. we do not consider cusps that may arise when three regions meet in a point. In the discrete setting we consider below this is not an issue. Furthermore, we mention that the data term $g(\hat{\mathbf{x}}, i)$ can be chosen freely and that more general regularity terms can be included, e.g. weighted curvature.

Again, we consider a discretized version of the continuous problem, where we note that for $\lambda = 0$, i.e. without curvature term, this is exactly the well-known Potts model [22]. This is why we call our discrete formulation “curvature Potts model”, the extension of the Potts model to a discrete curvature term.

Curvature Potts Model. We now show how to extend the integer linear program (5) to the curvature Potts model for multi-region segmentation. Like [14,32] we rely on indicator variables $x_f^i \in \{0, 1\}$ for each basic region where $x_f^i = 1$ signifies that the basic region f belongs to the segmentation region i (i.e. $s_f = i$). It must be assigned to exactly one such region, so

$$\sum_{i \in \mathcal{I}} x_f^i = 1 \quad \forall f \in \mathcal{F} \quad . \quad (7)$$

From the region variables x_f^i , the boundary of any region can be computed exactly as for the binary case: there are now separate boundary variables $y_{jj'}^i$, for each $i \in \{1, \dots, K\}$. Each of these “layers” has separate surface continuation, boundary continuation and boundary consistency constraints. This amounts to

the integer linear program

$$\begin{aligned}
& \min_{\mathbf{x}, \mathbf{y}} \mathbf{g}^T \mathbf{x} + \mathbf{w}^T \mathbf{y} & (8) \\
& \text{s.t.} \quad \sum_i x_f^i = 1 \quad \forall f \in \mathcal{F} \\
& \quad \sum_f m_e^f x_f^i + \sum_{j, j'} m_e^{jj'} y_{jj'}^i = 0 \quad \forall e \in \mathcal{E}, i \in \mathcal{I} \\
& \quad \sum_{j'} y_{j', j}^i = \sum_{j''} y_{j j''}^i \quad \forall j \in \mathcal{E}^o, i \in \mathcal{I} \\
& \quad \sum_{jj'} |m_{jj'}^e| y_{jj'}^i \leq 1 \quad \forall e \in \mathcal{E}, i \in \mathcal{I} \\
& \quad x_f^i \in \{0, 1\} \quad \forall f, i, \quad y_{jj'}^i \in \{0, 1\} \quad \forall i, j, j' \quad ,
\end{aligned}$$

where the only coupling constraint is now (7). The associated linear programming relaxation is again obtained by relaxing the sets $\{0, 1\}$ to $[0, 1]$ everywhere.

3.2 Inpainting

Given is now an image $I : \Omega \rightarrow \mathbb{R}$ together with a damaged region $\Omega_d \subset \Omega$ which is to be filled with proper values. We first review the continuous formulation for inpainting in [25]. The model minimizes over the functions $u : \Omega \rightarrow \mathbb{R}$,

$$\begin{aligned}
& \min_{u: \Omega \rightarrow \mathbb{R}} \nu \int_{\Omega} |\nabla u(\hat{\mathbf{x}})| d\hat{\mathbf{x}} + \lambda \int_{I_l}^{I_u} \int_{C_{u,t}} |\kappa_{C_{u,t}}(\hat{\mathbf{x}})|^2 d\mathcal{H}^1(\hat{\mathbf{x}}) dt \\
& \text{s.t.} \quad u(\hat{\mathbf{x}}) = I(\hat{\mathbf{x}}) \quad \forall \hat{\mathbf{x}} \in \Omega \setminus \Omega_d \quad , & (9)
\end{aligned}$$

where $C_{u,t} = \{\hat{\mathbf{x}} \mid u(\hat{\mathbf{x}}) = t\}$ is the set of level lines of $u(\cdot)$ for level t . I_u and I_l are the maximal and minimal intensity at the boundary of the damaged region respectively. Note that the first term here is the well-known *total variation* that is closely related to the length of level lines.

In principle there are infinitely many levels to consider, but one commonly [15] deals only with those levels that actually occur on the image boundary, i.e. a subset of $\mathcal{L} = \{0, \dots, 255\}$. In [25] this problem was addressed by an integer program similar to (5), with two differences: firstly, the variables are no longer binary – they can now take any value in $[I_l, I_u]$. Likewise, the boundary variables can take any value in $[0, I_u]$ where the maximal range is needed at the image border. In practice it is sensible to first subtract the constant I_l from the image so that the range is $[0, I_u - I_l]$ for all variables. Secondly, all region variables for the known region $\Omega \setminus \Omega_d$ are fixed to the corresponding values of $I(\cdot)$. In practice one only considers region variables that are inside Ω_d or very close to its border.

As noted in [25] and visualized in Figure 1 the resulting boundary variables do usually *not* reflect level lines: In this formulation lines are allowed to switch between different intensity levels. One way to circumvent this for the discrete

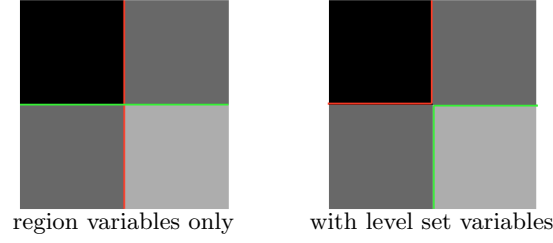


Fig. 1. Best viewed in color. When fixing the intensities, the effect of including level set variables becomes apparent: lines with same colors indicate an active pair (not all are shown). Without level sets one of the pairs connects a jump from black to dark-grey and one from dark-gray to light-grey. This is not a level line. When introducing level sets the correct level lines are found.

intensity range \mathcal{L} is to introduce variables that reflect the actual *level sets* of the intensity profile. Then, there is no longer *one* variable x_f for every basic region, there are now 255 *binary* variables $x_f^i \in \{0, 1\}, \forall i \in \mathcal{L} \setminus \{0\}$ with the intention that $x_f^i = 1$ if and only if $I_f \geq i$. This naturally induces the constraints $x_f^i \geq x_f^{i+1} \quad \forall f \in \mathcal{F}$.

Level lines are exactly the boundary lines of level sets, so using the procedure for binary segmentation to fit boundary lines to the binary variables x_f^i gives indeed level lines. There are then $|\mathcal{L}| - 1$ layers of variables x_f^i and $y_{j,j'}^i$, that are coupled by constraints on the level variables and further constrained by the boundary conditions, i.e. by setting the variables x_f^i for fixed regions so that they represent the level sets correctly.

A Compromise. Introducing 255 layers is not practicable for any but the smallest domains - the memory demands are too high. Hence, we consider an intermediate strategy: we group the intensity interval $[0, I_u - I_l]$ into B equally large bins, where B is set by the user and the bin sizes are all integral (this implies that the uppermost bin may only partially be used).

That is, we now have variables $x_f^b \in [0, I_B], \forall b \in \{1, \dots, B\}$, where $I_B = \lceil (I_u - I_l)/B \rceil$ is the size of each bin. We want $x_f^b = I_B$ if $I_f \geq b \cdot I_B$, else $x_f^b = \max\{0, I_f - (b - 1) \cdot I_B\}$. The result is the integer linear program

$$\begin{aligned}
 & \min_{\mathbf{x}, \mathbf{y}} \mathbf{w}^T \mathbf{y} & (10) \\
 & \text{s.t. } x_f^b \geq x_f^{b+1} \quad \forall f \in \mathcal{F}, b < B \\
 & \sum_f m_e^f x_f^b + \sum_{j,j'} m_e^{jj'} y_{jj'}^b = 0 \quad \forall e \in \mathcal{E}, b \leq B \\
 & \sum_{j'} y_{j',j}^b = \sum_{j''} y_{jj''}^b \quad \forall j \in \mathcal{E}^o, 1 \leq b \leq B \\
 & \sum_{jj'} |m_{jj'}^e| y_{jj'}^b \leq I_B \quad \forall e \in \mathcal{E}, 1 \leq b \leq B \\
 & x_f^b \in \{0, \dots, I_B\} \quad \forall f, b, \quad y_{jj'}^b \in \{0, \dots, I_B\} \quad \forall b, j, j'.
 \end{aligned}$$

plus some additional constraints for variables outside Ω_d .

3.3 Denoising

The formulation (10) for inpainting can be easily generalized to the problem of image denoising with absolute differences in the data term and a curvature prior on level lines. That is, the continuous model takes the form

$$\begin{aligned} \min_{u: \Omega \rightarrow \mathbb{R}} & \int_{\Omega} |I(\hat{\mathbf{x}}) - u(\hat{\mathbf{x}})| d\hat{\mathbf{x}} + \nu \int_{\Omega} |\nabla u(\hat{\mathbf{x}})| d\hat{\mathbf{x}} \\ & + \lambda \int_{\mathbb{R}} \int_{C_{u,t}} |\kappa_{C_{u,t}}(\hat{\mathbf{x}})|^2 d\mathcal{H}^1(\hat{\mathbf{x}}) dt \quad , \end{aligned}$$

where compared to (9) there is now a data term and u is to be optimized over all of Ω . The function $u : \Omega \rightarrow \mathbb{R}$ is an inherently continuous-valued function, so the region variables are no longer integral. However, to estimate the level lines one again needs to include variables reflecting level sets in the formulation. These variables, as well as the associated boundary variables, are still integral.

Since there is no longer an a priori known set of levels to consider, we can only approximate the desired behavior. To do this, we use the same trick as above: levels are grouped into bins and bins have to be filled from the bottom up.

The associated integer program is very similar to (10), so we only explain how to integrate the data term. It is well-known, e.g. [9], that an expression of the form

$$\min_{\mathbf{x} \geq \mathbf{0}} \mathbf{c}^T \mathbf{x} + |\mathbf{a}^T \mathbf{x} - b| \quad ,$$

where \mathbf{c} and \mathbf{a} are arbitrary vectors, and b some scalar, can be transformed to the auxiliary linear program

$$\begin{aligned} \min_{\mathbf{x} \geq \mathbf{0}, z_+ \geq 0, z_- \geq 0} & \mathbf{c}^T \mathbf{x} + z_+ + z_- \\ \text{s.t.} & \mathbf{a}^T \mathbf{x} - b = z_+ - z_- \quad . \end{aligned}$$

This construction is easily generalized to multiple absolutes, where each will introduce two auxiliary variables.

4 Alternative Strategies

In this section, we explore alternative strategies, both in the formulation of the integer linear program and in the way the linear programming relaxations are solved.

4.1 Linear Programming and Higher Order MRFs

El-Zehiry and Grady [10] proposed to model curvature on cell complexes as a higher order Markov Random Field with ternary cliques. This only allows a 4-connectivity, but as we will see below a related formulation with fourth-order

cliques can handle all connectivities. In this paper we explore the associated integer linear program and its linear programming relaxation.

The nodes in the MRF correspond to the basic regions in the mesh, so region cost are unary terms. The ternary cliques are induced by the edge pairs: for a 4-connectivity, whenever a pair of adjacent edges represents a direction change it will always have a common adjacent region, so in total there are only three regions to consider. When using QPBO the associated length and curvature cost can be decomposed into two-cliques. For our integer program they are simply pre-computed into a lookup-table for all possible constellations of the three regions.

When considering arbitrary meshes each edge pair can have up to four adjacent regions, so fourth order cliques are necessary. The (relevant) cliques in the arising graph can now be of order two (at the image border), three or four. The MRF can be associated a “standard” integer linear program, growing exponentially with the order of the maximal clique. Problems of this form were considered in [30].

Here we give a formulation for the binary segmentation problem. As above, there is a variable x_f for every basic region f . For any clique $\mathcal{C} = \{f_1, \dots, f_{k_{\mathcal{C}}}\}$ of order $k_{\mathcal{C}}$ the integer program contains $2^{k_{\mathcal{C}}}$ auxiliary variables representing all possible value assignments, i.e. for any $v \in \{0, 1\}^{k_{\mathcal{C}}}$ there is a variable $y_{\mathcal{C}}^v$ with cost defined by the above mentioned lookup tables.

The constraint system expresses consistency of the region variables and the clique variables, i.e. the ILP reads

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{y}} \quad & \mathbf{c}_{\mathbf{x}}^T \mathbf{x} + \mathbf{c}_{\mathbf{y}}^T \mathbf{y} \\ \text{s.t.} \quad & x_f = \sum_{v: v(f)=1} y_{\mathcal{C}}^v \quad \forall f, \mathcal{C} : f \in \mathcal{C} \\ & 1 - x_f = \sum_{v: v(f)=0} y_{\mathcal{C}}^v \quad \forall f, \mathcal{C} : f \in \mathcal{C} \quad , \end{aligned}$$

where $v(f)$ means the entry of v that corresponds to the basic region f for the currently considered (ordering of the) clique \mathcal{C} . Note that this integer program is not completely equivalent to the one given in section 2: wherever two regions meet in a point the cost is overestimated as all possible boundary configurations are penalized. In contrast, the formulation (5) will choose the best configuration here.

4.2 Do We Need Standard Solvers?

Standard linear programming solvers have only sparsely been used in Computer Vision, due to the high demands on memory and the long run-times. Such solvers have proven to work fine for curvature problems [25], but the image sizes that can be handled in practice are rather small - processing a 128×128 image with a 16-connectivity and the simplex method requires 4 GB. Clearly, less memory intensive techniques are desirable and one may even want to accept a loss in numerical precision of the solution. We examine two strategies.

Min-sum Diffusion. The first technique is called *min-sum diffusion* and applicable only to the linear program derived from the higher order MRF. It was used for the higher order terms in [30], where it is described in detail. Here we only mention that the method is doing a reparameterization of the cost function, i.e. it shifts cost from the higher order terms to the unary terms without affecting the represented energy. Finding the optimal reparameterization is equivalent to solving the dual linear program. However, not all reparameterizations are considered, so in general the method is suboptimal – see the experiments below.

The main advantages of the method are its simplicity and the savings in memory consumption it offers.

Augmented Lagrangians. The augmented Lagrangian method [3, chap. 4.2] is a very general technique to handle a large class of constrained optimization problems. Here we detail it on the example of the linear program

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} \quad & \mathbf{A} \mathbf{x} = \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0}, \end{aligned} \tag{11}$$

assumed to be bounded and feasible. The augmented Lagrangian method introduces the convex relaxation

$$\min_{\mathbf{x} \geq \mathbf{0}} \Psi_{\gamma, \boldsymbol{\lambda}}(\mathbf{x}) = \mathbf{c}^T \mathbf{x} + \boldsymbol{\lambda}^T (\mathbf{A} \mathbf{x} - \mathbf{b}) + \gamma \|\mathbf{A} \mathbf{x} - \mathbf{b}\|^2,$$

and since $\mathbf{A} \mathbf{x} = \mathbf{b}$ for any feasible solution the cost of such solutions remain untouched - for any $\gamma > 0$ and $\boldsymbol{\lambda} \in \mathbb{R}^n$. The choice of γ and $\boldsymbol{\lambda}$ significantly influences the quality of the relaxation, but also the computational difficulty of minimizing $\Psi_{\gamma, \boldsymbol{\lambda}}(\mathbf{x})$. In practice one proceeds in iterations, starting with some γ_0 and $\boldsymbol{\lambda} = \mathbf{0}$. Ideally, each iteration would minimize $\Psi_{\gamma, \boldsymbol{\lambda}}(\mathbf{x})$ with a guaranteed precision of some specified ϵ . In our case this is too computationally costly, so we only run a fixed number of (inner) iterations. After each outer iteration, the multipliers are updated according to [3, chap. 4.2] $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} + \gamma(\mathbf{A} \mathbf{x} - \mathbf{b})$ and γ is multiplied with a constant factor (we take 1.1).

To minimize $\Psi_{\gamma, \boldsymbol{\lambda}}(\mathbf{x})$ for given γ and $\boldsymbol{\lambda}$, we exploit that the gradient of the function is Lipschitz-continuous and apply an accelerated first order scheme [17] with a projection on the set $\mathbf{x} \geq \mathbf{0}$. In the case of multi-region segmentation some constraints can be included directly into this projection [16]. This scheme is parallelizable, so we consider a GPU-implementation. The computed solutions are very nearly feasible but, due to the limit on the inner iterations, their cost are frequently above the optimal ones.

5 Experiments

We now give results for all discussed applications, where the experiments were run on a 3 GHz Core2 Duo with 8 GB memory and an NVidia Tesla 2050 card. We found that our problems are best solved via interior point methods, which frequently requires the entire 8 GB. We use the commercial solvers FICO Xpress and Gurobi.

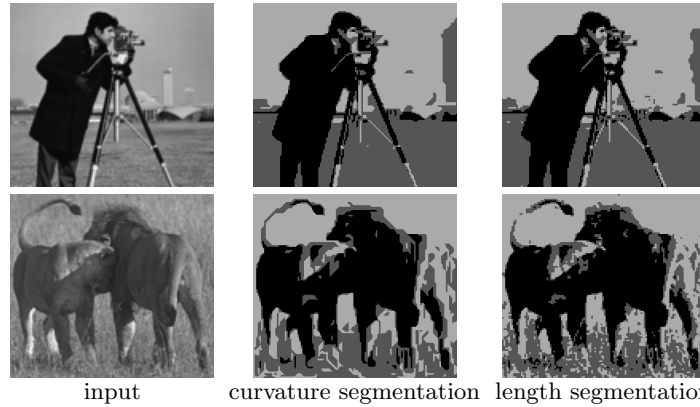


Fig. 2. Comparison of purely curvature-based and purely length-based segmentation (all 8-connected).



Fig. 3. Curvature-inpaintings with a 16-connectivity: already two bins have a significant effect on the results.

5.1 Image Segmentation

For image segmentation we choose an 8-connectivity and the data term $g(\hat{\mathbf{x}}, i) = (I(\hat{\mathbf{x}}) - \mu_i)^2$, where we select the mean values as the 16%, 50% and 83%-percentiles of the image.

Two results for multi-region segmentation on image sizes of 128×116 are given in Figure 2. Note in particular that without curvature, even a low length weight splits the tail of the lion into several parts, whereas the proposed curvature-based method closes the gap. For the length-based results we re-ran our method with a curvature weight of 0. Despite the NP-hardness of the Potts model they are globally optimal. The computed curvature-based solutions (roughly 5 hours) are 15% and 37% away from the respective proven lower bounds.

Running the Augmented Lagrangian solver (Sec. 4.2) for the lions and 30×2500 iterations gave a nearly feasible solution 0.01% higher than the optimum. The thresholded solution looks visually very similar. With 1.5 hours (18x on the CPU) this is faster than the exact solver and needs less memory.

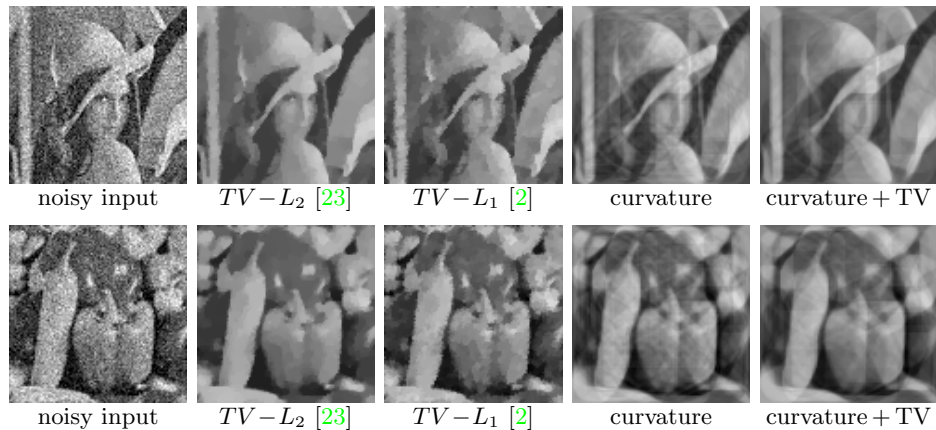


Fig. 4. Image denoising with first and second order methods (our implementations) on a resolution of 96×96 . With curvature regularity (16-connected) there are less block artifacts.

5.2 Inpainting

For inpainting, Figure 3 demonstrates that indeed the introduction of bins has a significant influence on the results. We prefer a 16-connectivity over a large number of bins.

At the same time, it can be seen that the method can handle non-singly connected domains. To estimate the direction of the level lines at the border of the inpainting domains we use a structure tensor based method similar to [5]. The inpainted image is obtained by summing all bin variables associated to each face.

5.3 Denoising

With a 16-connectivity we can handle denoising tasks for resolutions of 96×96 , using the entire 8 GB and roughly 3 hours computing time. As shown in Figure 4 the results are less blocky than the state of the art. However, they contain new kinds of artifacts. When adding a length penalty these are partially removed and the results look more natural than with a TV term alone.

Further, we tested different numbers of bins for an 8-connectivity, but at least 6 bins were needed to affect the solutions and this already takes 3 GB.

5.4 Comparison of ILPs

When using standard solvers, we found that the clique-lp requires roughly five times more memory. As a consequence, for the comparison of the two presented ILPs we use an image resolution of 64×64 and only consider binary segmentation. As shown in Figure 5 there is no clear winner: both methods can produce tighter lower bounds and better integral solutions than the respective other.

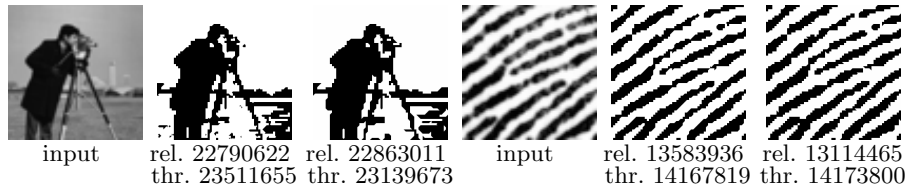


Fig. 5. Comparison of the different strategies for an 8-connectivity. Next to the input images: Higher Order ILP. Left to that: ILP from [26]. In all cases the lower bounds and the energies of the thresholded solutions are given.

The problem with the higher order ILP is its huge memory consumption, though, and the run-times are also clearly inferior to the other ILP. Hence, we ran the min-sum diffusion of section 4.2 (until near convergence), but with lp-energies of 22770853 (for the cameraman) and 13558816 (for the fingerprints) it was far away from the actual optima (see Fig. 5). Still, the thresholded solutions are useful and the memory consumption is now a factor of two below the standard solving of [25] (plus boundary consistency).

Conclusion

This paper has shown that a discrete curvature regularizer can be applied to multi-label region-based problems. It has addressed three different tasks: segmentation, denoising and inpainting. All are modeled as linear programs with several layers of variables coupled by relatively few constraints. Hence, in future work we intend to consider dual decomposition strategies.

At the same time, we have shown in this work that linear programs need not necessarily be solved by standard solvers if one is willing to sacrifice a bit of precision. Here, we have considered two different linear programs and two strategies of solving them.

To facilitate further research in this area, the source code associated to this paper will be made publicly available.

References

1. A. Amini, T. Weymouth, and R. Jain. Using dynamic programming for solving variational problems in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 12(9):855–867, Sept. 1990. 2
2. J.-F. Aujol, G. Gilboa, T. Chan, and S. Osher. Structure-texture image decomposition modeling, algorithms, and parameter selection. *International Journal on Computer Vision (IJCV)*, 67(1):111–136, 2006. 12
3. D. Bertsekas. *Nonlinear Programming, 2nd edition*. Athena Scientific, 1999. 10
4. A. Bhusnurmath and C. Taylor. Graph cuts via l_1 norm minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 30(10):1866–1871, Oct. 2008. 2
5. F. Bornemann and T. März. Fast image inpainting based on coherence transport. *Journal on Mathematical Imaging and Vision*, 28(3):259–278, July 2007. 1, 12

6. C. Brito-Loeza and K. Chen. Multigrid algorithm for higher order denoising. *SIAM Journal for Imaging Science*, 3(3):363–389, 2010. [2](#)
7. T. Chan and S. Esedoglu. Aspects of total variation regularized l^1 function approximation. *SIAM Journal of Applied Mathematics*, 65(5):1817–1837, 2004. [1](#)
8. T. Chan, S. Kang, and J. Shen. Euler’s elastica and curvature based inpaintings. *SIAM Journal of Applied Mathematics*, 2:564–592, 2002. [1](#)
9. G. Dantzig and M. Thapa. *Linear Programming 1: Introduction*. Springer Series in Operations Research. Springer, 1997. [8](#)
10. N. El-Zehiry and L. Grady. Fast global optimization of curvature. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, California, June 2010. [2](#), [8](#)
11. S. Esedoglu and R. March. Segmentation with depth but without detecting junctions. *Journal on Mathematical Imaging and Vision*, 18:7–15, 2003. [1](#)
12. D. Greig, B. Porteous, and A. Seheult. Exact maximum *a posteriori* estimation for binary images. *Journal of the Royal Statistical Society, Series B*, 51(2):271–279, 1989. [1](#)
13. P. Hammer, P. Hansen, and B. Simeone. Roof duality, complementation and persistency in quadratic 0-1 optimization. *Mathematical Programming*, 28(2):121–155, Feb. 1984. [2](#)
14. J. Kleinberg and E. Tardos. Approximation algorithms for classification problems with pairwise relationships: metric labeling and Markov Random Fields. In *Symposium on Foundations of Computer Science*, 1999. [1](#), [2](#), [5](#)
15. S. Masnou. Disocclusion: A variational approach using level lines. *IEEE Transactions on Image Processing (TIP)*, 11:68–76, 2002. [1](#), [6](#)
16. C. Michelot. A finite algorithm for finding the projection of a point onto the canonical simplex of \mathbb{R}^n . *Journal on Optimization Theory and Applications*, 50(1), July 1986. [10](#)
17. Y. Nesterov. *Introductory lectures on convex optimization*. Applied Optimization. Kluwer Academic Publishers, 2004. [10](#)
18. M. Nikolova, S. Esedoglu, and T. Chan. Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM Journal of Applied Mathematics*, 66(5):1632–1648, 2006. [1](#)
19. M. Nitzberg, D. Mumford, and T. Shiota. Filtering, segmentation and depth. In *LNCS*, volume 662. Springer Verlag, 1993. [1](#)
20. P. Parent and S. Zucker. Trace inference, curvature consistency, and curve detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 11(8):823–839, 1989. [2](#)
21. T. Pock, A. Chambolle, H. Bischof, and D. Cremers. A convex relaxation approach for computing minimal partitions. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Miami, Florida, June 2009. [2](#)
22. R. Potts. Some generalized order-disorder transformation. *Proceedings of the Cambridge Philosophical Society*, 48:106–109, 1952. [5](#)
23. L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992. [1](#), [12](#)
24. T. Schoenemann and D. Cremers. Introducing curvature into globally optimal image segmentation: Minimum ratio cycles on product graphs. In *IEEE International Conference on Computer Vision (ICCV)*, Rio de Janeiro, Brazil, Oct. 2007. [2](#)
25. 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 (ICCV)*, Kyoto, Japan, Sept. 2009. [1](#), [2](#), [3](#), [4](#), [6](#), [9](#), [13](#)

26. T. Schoenemann, F. Kahl, S. Masnou, and D. Cremers. A linear framework for region-based image segmentation and inpainting involving curvature regularity. Technical report, ArXiv report, Feb. 2011. [2](#), [3](#), [13](#)
27. P. Strandmark and F. Kahl. Curvature regularization for curves and surfaces in a global optimization framework. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, St. Petersburg, Russia, July 2011. [4](#)
28. X.-C. Tai, J. Hahn, and G. Chung. A fast algorithm for Euler's elastica model using augmented Lagrangian method. Technical report, UCLA CAM report, July 2010. [2](#)
29. D. Tschumperlé. Fast anisotropic smoothing of multi-valued images using curvature-preserving PDE's. *International Journal on Computer Vision (IJCV)*, 68(1):65–82, June 2006. [1](#)
30. T. Werner. High-arity interactions, polyhedral relaxations, and cutting plane algorithm for soft constraint optimisation (MAP-MRF). In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Anchorage, Alaska, June 2008. [2](#), [9](#), [10](#)
31. J. Yuan, E. Bae, X.-C. Tai, and Y. Boykov. A continuous max-flow approach to Potts model. In *European Conference on Computer Vision (ECCV)*, Iraklion, Greece, Sept. 2010. [2](#)
32. C. Zach, D. Gallup, J.-M. Frahm, and M. Niethammer. Fast global labeling for real-time stereo using multiple plane sweeps. In *Vision, Modeling and Visualization Workshop (VMV)*, Konstanz, Germany, Oct. 2008. [2](#), [5](#)
33. W. Zhu and T. Chan. Image denoising and mean curvature. Technical report, Courant Institute, New York, Nov. 2006. Preprint, available online http://www.cims.nyu.edu/~wzhu/meancurvature_06Nov30.pdf. [2](#)