Partial Enumeration and Curvature Regularization

Olsson, Carl; Ulén, Johannes; Boykov, Yuri; Kolmogorov, Vladimir

Published in:
Computer Vision (ICCV), 2013 IEEE International Conference on

DOI:
10.1109/ICCV.2013.365

2013

Partial Enumeration and Curvature Regularization

Carl Olsson Johannes Ulén
Centre for Mathematical Sciences
Lund University, Sweden
calle@maths.lth.se ulen@maths.lth.se

Yuri Boykov
Computer Science
UWO, Canada
yuri@csd.uwo.ca

Vladimir Kolmogorov
Inst. of Science & Technology
Austria
vnk@ist.ac.at

Abstract

Energies with high-order non-submodular interactions have been shown to be very useful in vision due to their high modeling power. Optimization of such energies, however, is generally NP-hard. A naive approach that works for small problem instances is exhaustive search, that is, enumeration of all possible labelings of the underlying graph. We propose a general minimization approach for large graphs based on enumeration of labelings of certain small patches. This partial enumeration *technique reduces complex high-order energy formulations to pairwise* Constraint Satisfaction Problems *with unary costs (uCSP), which can be efficiently solved using standard methods like TRW-S. Our approach outperforms a number of existing state-of-the-art algorithms on well known difficult problems (e.g. curvature regularization, stereo, deconvolution); it gives near global minimum and better speed.*

Our main application of interest is curvature regularization. In the context of segmentation, our partial enumeration technique allows to evaluate curvature directly on small patches using a novel integral geometry *approach.* [1]

1. Introduction

Optimization of curvature and higher-order regularizers, in general, has significant potential in segmentation, stereo, 3D reconstruction, image restoration, in-painting, and other applications. It is widely known as a challenging problem with a long history of research in computer vision. For example, when Geman and Geman introduced MRF models to computer vision [8] they proposed first- and second-order regularization based on *line process*. The popular *active contours* framework [12] uses elastic (first-order) and bending (second-order) energies for segmentation. Dynamic programming was used for curvature-based inpainting [18]. Curvature was also studied within PDE-based [5] and *level-sets* [6] approaches to image analysis.

Recently there has been a revival of interest in second-order smoothness for discrete MRF settings. Due to the success of global optimization methods for first-order MRF models [3, 10] researchers now focus on more difficult second-order functionals [34] including various discrete approximations of curvature [27, 7, 30]. Similarly, recent progress on global optimization techniques for first-order continuous geometric functionals [20, 23, 16, 35] has lead to extensions for curvature [4].

Our paper proposes new discrete MRF models for approximating curvature regularization terms like $\int_{\partial S} |\kappa| d\sigma$. We primarily focus on the absolute curvature. Unlike length or squared curvature regularization, this term does not add shrinking or ballooning bias.

Our technique evaluates curvature using small patches either on a grid or on a cell complex, as illustrated in Fig.1. In case of a grid, our patches use a novel *integral geometry* approach to evaluating curvature. In case of a complex, our patch-based approach can use standard geometry for evaluating curvature. The relationship to previous discrete MRF models for curvature is discussed in Section 2.

We also propose a very simple and efficient optimization technique, *partial enumeration*, directly applicable to our patch-based curvature model and some other high-order problems. Our approach reduces high-order discrete energy formulations to pair-wise *Constraint Satisfaction Problem* with unary costs (uCSP). The details of our patch-based optimization technique and related work are discussed in Section 3. Our contributions can be summarized as follows:

- simple patch-based models for curvature
- integral geometry technique for evaluating curvature
- easy-to-implement partial enumeration technique reducing patch-based MRF models to a pairwise *Constraint Satisfaction Problem* with unary costs directly addressable with many approximation algorithms
- our uCSP modification of TRWS outperforms several alternatives producing near-optimal solutions with smaller optimality gap and shorter running times

The experiments in Sections 3 and 4 show that our patch-based technique obtains state-of-the-art results not only for curvature-based segmentation, but also for high-order stereo and deconvolution problems.

[1] This work has been funded by the Swedish Research Council (grant 2012-4213), the Crafoord Foundation, the Canadian Foundation for Innovation (CFI 10318) and the Canadian NSERC Discovery Program (grant 298299-2012RGPIN).

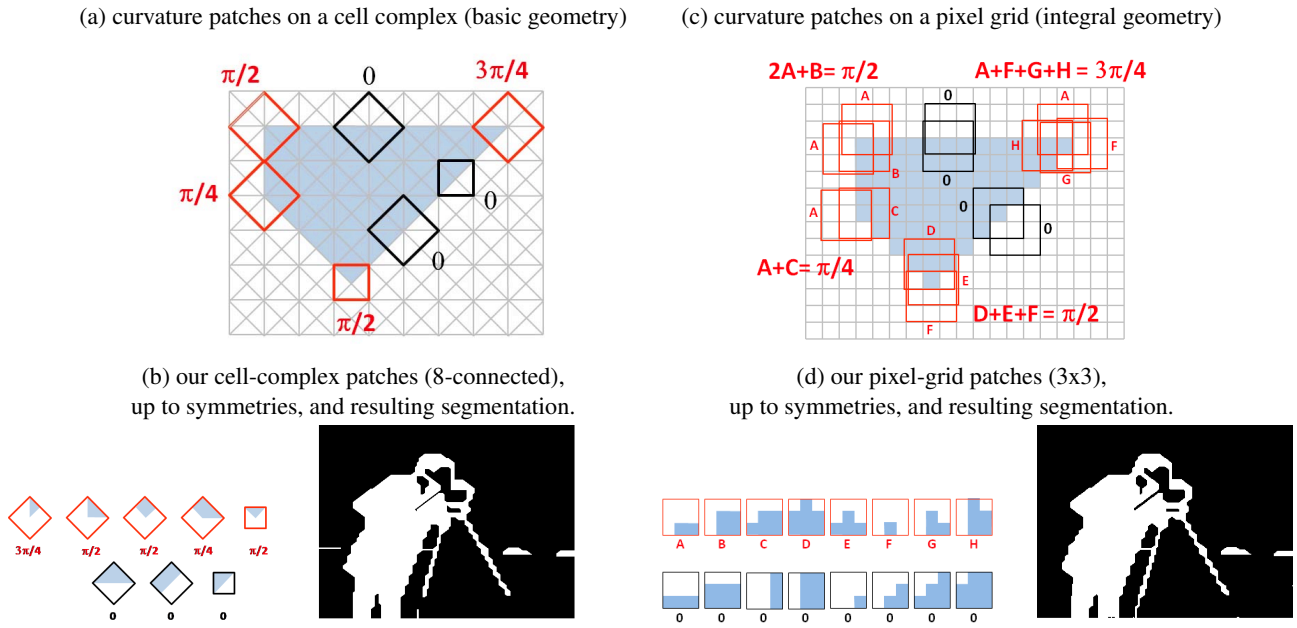


Figure 1: Evaluating curvature of a segment on a complex (a,b) and on a grid (c,d) using standard and integral geometry. At sufficiently high resolution, any segment C is a polygon. Thus, to minimize curvature functionals like $\int_C |\kappa| ds$ we need to evaluate all corners. We use (overlapping) patches created for each vertex on a complex (a) and for each pixel on a grid (c). A patch on a complex (a,b) consists of all cells adjacent to a vertex and a grid patch (c,d) is a square window centered at a pixel. For $\pi/4$ precision as on 8-complex (a), we use 3x3 windows on a grid (b). For finer $\pi/8$ precision as on 16-complexes, we use 5x5 windows. Note that each corner on a complex (a) can be directly evaluated from a configuration (labeling) of a single patch using standard geometry. However, each corner on a grid (c) should be evaluated using integral geometry by summing over multiple patches covering the corner. Patch configurations in black occur at straight boundaries and should contribute zero weights. Patch configurations in red correspond to curved boundaries. The weights $A, \ldots, H$ for all such configurations (d) can be pre-computed from a system of linear equations for all types of corners. The accuracy of integral geometry approach to curvature on a grid is comparable to the standard basic geometry used on complexes, see (b) and (d).

2. Curvature on patches and related work

We discuss approximation of curvature in the context of binary segmentation with regularization energy

$$E(S) = \int_{\text{int}(S)} f(x)\, dx + \int_{\partial S} \lambda |\kappa| d\sigma, \qquad (1)$$

where κ is curvature, λ is a weighting parameter, and unary potential $f(x)$ is a data term.

Our grid-patches in Fig.1(c) and our complex-patches in Fig.1(a) can be seen as "dual" methods for estimating curvature in exactly the same way as *geo-cuts* [2] and complex-based approach in [31] are "dual" methods for evaluating geometric length. Our grid-patch approach to curvature extends ideas in *geo-cuts* [2] that showed how discrete MRF-based regularization methods can use *integral geometry* to accurately approximate length via Cauchy-Crofton formula. We show how general integral geometry principles can also be used to evaluate curvature, see Fig.1(c). The complex-patch technique in Fig.1(a) uses an alternative method for approximating curvature based on standard geometry as in [27, 7, 30].

Our patch-based curvature models could be seen as extensions of *functional lifting* [4] or *label elevation* [21]. Analogously to the *line processes* in [8], these second-order regularization methods use variables describing both location and orientation of the boundary. Thus, their curvature is the first-order (pair-wise) energy. Our patch variables include enough information about the local boundary to reduce the curvature to unary terms.

Curvature is also reduced to unary terms in [27] using auxiliary variables for each pair of adjacent *line processes*. Their integer LP approach to curvature is formulated over a large number of binary variables defined on fine geometric primitives (vertexes, faces, edges, pairs of edges, etc), which are tied by constraints. In contrast, our unary representation of curvature uses larger scale geometric primitives (overlapping patches) tied by consistency constraints. The number of corresponding variables is significantly smaller, but they have a much larger label space. Unlike [27] and us, [7, 30] represent curvature via high-order factors.

Despite technical differences in the underlying formulations and optimization algorithms, our patch-based approach for complexes in Fig.1(a) and [27, 30] use geometrically equivalent models for approximating curvature. That is, all of these models would produce the same solution, if

there were exact global optimization algorithms for them. The optimization algorithms for these models do however vary, both in quality, memory, and run-time efficiency.

In practice, grid-patches are easier to implement than complex-patches because the grid's regularity and symmetry. While integral geometry estimates curvature on a pixel grid as accurately as the standard geometry on a cell complex, see Figs.1(b,d), in practice, our proposed optimization algorithm for the corresponding uCSP problems works better (with near-zero optimality gap) for the grid version of our method. To keep the paper focused, the rest of the paper primarily concentrates on grid-based patches.

Grid patches were also recently used for curvature evaluation in [28]. Unlike our integral geometry in Fig.1(c), their method computes a minimum response over a number of affine filters encoding some learned "soft" patterns. The response to each filter combines deviation from the pattern and the cost of the pattern. The mathematical justification of this approach to curvature estimation is not fully explained and several presented plots indicate its limited accuracy. As stated in [28], "the plots do also reveal the fact that we consistently overestimate the true curvature cost." The extreme "hard" case of this method may reduce to our technique if the cost of each pattern is assigned according to our integral geometry equations in Fig.1(c). However, this case makes redundant the filter response minimization and the pattern costs learning, which are the key technical ideas in [28].

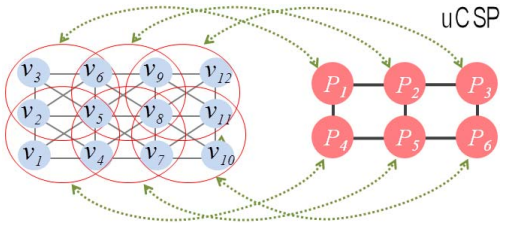
3. Simple Patch-based Optimization

One way to optimize our patch-based curvature model is to formulate the optimization problem on the original image pixel grid $\langle V, \mathcal{C} \rangle$ in Figure 2(a) using pixel variables $\mathbf{x} = \{x_i | i \in V\}$, high-order factors $\alpha \in \mathcal{C}$, and energy

$$E(\mathbf{x}) = \sum_{\alpha \in \mathcal{C}} E_\alpha(\mathbf{x}_\alpha) \tag{2}$$

where $\mathbf{x}_\alpha = \{x_i | i \in \alpha\}$ is the restriction of $\mathbf{x}$ to α. Optimization of such high-order energies is generally NP-hard, but a number of existing approximate algorithms for certain high-order MRF energies could be applied. Our experimental section includes the results of some generic methods [14, 11] that have publicly available code.

We propose a different approach for optimizing our high-order curvature models that equivalently reformulates the problem on a new graph, see Fig.2(b). The motivation is as follows. One naive approach applicable to NP-hard high-order energies on small images is exhaustive search that enumerates all possible labelings of the underlying pixel graph. On large problems one can use partial enumeration to simplify high-order problems. If some set of relatively small overlapping patches covers all high-order factors, we can build a new graph where nodes correspond to patches



(a) pixel-based graph (b) patch-based graph

Figure 2: Patch-based models in Fig.1, represented either on a pixel-graph with high-order interactions (a) or on a patch-graph where each patch corresponds to one node (b). In (b) the curvature reduces to unary terms, but the graph includes pairwise consistency constraints due to patch overlaps.

and their labels enumerate patch states, as in Fig.2(b). Note that high-order interactions reduce to unary potentials, but, due to patch overlap, hard pair-wise consistency constraints must be enforced.

Our general approach transforms a high-order optimization problem to a pair-wise Constraint Satisfaction Problem with unary costs (uCSP). Formally, the corresponding energy could be defined on graph $\langle \mathcal{V}, \mathcal{E} \rangle$ in Fig.2(b) where nodes correspond to a set of patches $\mathcal{V}$ with the following property: for every factor $\alpha \in \mathcal{C}$ there exists patch $\beta \in \mathcal{V}$ such that $\alpha \subseteq \beta$. For example, $\mathcal{V} = \mathcal{C}$ works, but, in general, patches in $\mathcal{V}$ can be bigger than factors in $\mathcal{C}$. We refer to nodes in $\mathcal{V}$ as super nodes. Clearly, (2) could be equivalently rewritten as an energy with unary and pairwise terms:

$$E_{\text{super}}(\mathbf{X}) = \sum_{\alpha \in \mathcal{V}} U_\alpha(X_\alpha) + \sum_{(\alpha,\beta) \in \mathcal{E}} P_{\alpha\beta}(X_\alpha, X_\beta) \tag{3}$$

The label X_α of a super node α corresponds to the state of all individual pixels $\mathbf{x}_\alpha$ within the patch. By enumerating all possible pixel states within the patch we can now encode the higher order factor $E_\alpha(x_\alpha)$ into the unary term $U_\alpha(X_\alpha)$ of (3). The pairwise consistency potential $P_{\alpha\beta}(X_\alpha, X_\beta) = 0$ if variables X_α and X_β agree on the overlap $\alpha \cap \beta$, and $P_{\alpha\beta}(X_\alpha, X_\beta) = +\infty$ otherwise. The set of edges $\mathcal{E}$ may contain all pairs $\{\alpha, \beta\} \subset \mathcal{V}$ such that $\alpha \cap \beta \neq \varnothing$, but a smaller could be enough. For example, the graph in Fig.2(b) does not need diagonal edges. A formal procedure for selecting the set of edges is given in our extended technical report [22].

Optimization of pairwise energy (3) can be addressed with standard methods like [13, 9] that can be modified for our specific consistency constraints to gain significant speed-up (see Sec.3.2).

LP relaxations When we apply method like TRW-S [13] to energy (3), we essentially solve a higher-order relaxation of the original energy (2). Many methods have been proposed in the literature for solving higher-order relaxations, e.g. [29, 15, 19, 33, 14] to name just a few. To understand

the relation to these methods, in [22] we analyze which specific relaxation is solved by our approach. We then argue that the complexity of message passing in our scheme roughly matches that of other techniques that solve a similar relaxation. [2] In practice, the choice of the optimization method is often motivated by the ease of implementation; we believe that our scheme has an advantage in this respect, and thus may be preferred by practitioners.

Other related work The closest analogue of our approach is perhaps the "hidden transformation" approach [1] that converts an arbitrary CSP into a pairwise CSP (also known as the "constraint satisfaction dual problem"). We provide a *weighted* version of this transformation; to our knowledge, this has not been studied yet, and the resulting relaxation has not been analyzed.

Our method bears some resemblance to the work [15] that also uses square patches. However, we believe that the relaxation solved in [15] is weaker than ours; details are discussed in [22].

Researchers also considered alternative techniques for converting a high-order energy of binary variables into a pairwise one. We will compare to one such technique, [11], which generalizes roof duality to factors of order 3 and 4.

3.1. Application to $\pi/2$-precision curvature

In this section we illustrate our approach on a very coarse approximation of curvature where we only allow boundary edges that are either horizontal or vertical. It is shown in [7] that the resulting energy can be formulated as in (2) where $\mathcal{C}$ contains the edges of an 8-connected neighborhood, see Fig. 6. In contrast we formulate the problem as (3) where $\mathcal{V}$ is the set of all 2×2 patches. Consider the patches in Figure 4 and their curvature estimates. The patches have

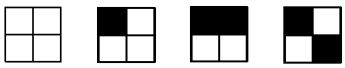


Figure 4: Four of the 16 patch states used to encode curvature with penalties 0, $\pi/2$, 0 and 2π respectively.

4 pixel boundaries that intersect in the middle of the patch. To compute the curvature contribution of a patch we need to determine which of the 4 pixel boundaries also belong to the segmentation boundary. If two neighboring pixels (sharing a boundary) have different assignments then their common boundary belongs to the segmentation boundary.

Figure 5 shows the approach. We start by forming

[2] Message passing techniques require the minimization of expressions of the form $E_\alpha(\mathbf{x}_\alpha) + \dots$ where dots denote lower-order factors. Here we assume that this expression is minimized by going through all possible labellings $\mathbf{x}_\alpha$. This would hold if, for example, $E_\alpha(\cdot)$ is represented by a table (which is the case with curvature). Some terms $E_\alpha(\cdot)$ used in practice have a special structure that allow more efficient computations; in this case other techniques may have a better complexity. One example is *cardinality-based potentials* [32] which can have a very high-order.

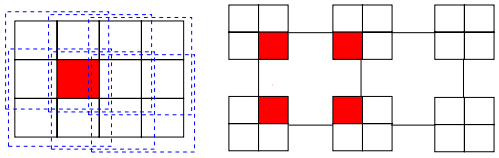


Figure 5: Super nodes formed in a sliding window fashion. The red pixel occurs in 4 super nodes. Pairwise interactions ensure that shared pixels are assigned the same value.

patches of size 2×2 into super nodes. This is done in a sliding window fashion, that is, super node (r, c) consists of the nodes (r, c), $(r, c+1)$, $(r+1, c)$ and $(r+1, c+1)$, where r and c are the row and column coordinates of the pixels.

Each super node label can take 16 values corresponding to states of the individual pixels. The curvature interaction and data terms of the original problem are now transformed to unary potentials. Note that since patches are overlapping pixels can be contained in up to four super nodes. In order not to change the energy we therefore weight the contribution from the original unary term, $f(x)$ in (1), to each patch such that the total contribution is 1. For simplicity we give pixels that occur k times the weight $1/k$ in each super node.

Finally to ensure that each pixel takes the same value in all the super nodes where it is contained we add the "consistency" edges $\mathcal{E}$ between neighboring super nodes (see Fig. 5). Note it is enough to use a 4-connected neighborhood.

Our two approaches from Figure 1 and [27, 7, 30] all assign the same curvature costs for the patches in Figure 4. Therefore, assuming that the global minimum can be found, they yield the same solution for $\pi/2$-precision curvature.

3.2. Efficient Message Passing

Since the number of labels can become very large when we have higher order factors it is essential to compute messages fast. The messages sent during optimization has the form

$$m^t_{\alpha\beta}(X_\beta) = \min_{X_\alpha}(P_{\alpha\beta}(X_\alpha, X_\beta) + h(X_\alpha)), \qquad (4)$$

where h is some function of super node label X_α.

To compute the message we order the labels of both node α and β into (disjoint) groups according to the assignments of the shared pixels. The message values $m^t_{\alpha\beta}(X_\beta)$ for all the X_β in the same group can now be found by searching for the smallest value of $h(X_\alpha)$ in the group consistent with the X_β. The label order depends on the direction of the edge between α and β, however it does not change during optimization and can therefore be precomputed at startup. The bottleneck is therefore searching the groups for the minimal value which can be done in linear time.

Note that this process does not require that all the possible patch assignments are allowed. For larger patches (see

λ	TRW-S Energy	TRW-S Lower bound	Unlabled by GRD(-heur)	TRW-S running time	GRD(-heur) running time
$2.5 \cdot 10^{-5}$	4677	4677	0% (0%)	0.934s	10737s (7.08s)
$2.5 \cdot 10^{-4}$	4680	4680	0% (0%)	0.961s	9287s (10.7s)
$2.5 \cdot 10^{-3}$	4707	4707	0.2% (0.2%)	2.43s	10731s (7.32s)
$2.5 \cdot 10^{-2}$	4910	4910	6.7% (6.8%)	3.98s	12552s (6.96s)
$2.5 \cdot 10^{-1}$	5833	5833	100% (100%)	14.3s	12337s (10.9s)
$2.5 \cdot 10^{0}$	7605	7605	100% (100%)	28.8s	7027s (22.2s)

Figure 3: Our results for 2×2 curvature with different regularization weight λ (top row of images). TRW-S with super nodes gives a tight lower bound. The figures within parenthesis are results when using the heuristics proposed for speedup in [11]).

Section 3.4) some of the patch states may not be of interest to us and the corresponding labels can simply be removed.

Figure 3 compares our approach for $\pi/2$-precision curvature to Generalized Roof Duality (GRD) [11]. We used TRW-S [13] with 2×2 patches and our efficient message passing scheme. Our approach gives no duality gap.

3.3. Lower Bounds using Trees

As observed in [7] the 2×2 curvature interaction reduces to pairwise interactions between all the pixels in the patch. In this discrete setting (1) reduces to (2) where $\mathcal{C}$ consists of the edges of the underlying (8-connected) graph, see Figure 6. Therefore it could in principle be solved using roof duality (RD) [25] or TRW-S [13]. (Note that this is only true for this particular neighborhood and the interaction penalty.) However, it may still be useful to form super nodes. Methods such as [13] work by decomposing the problem into subproblems on trees and combining the results into a lower bound on the optimal solution. Sub-trees with super nodes are in general stronger than regular trees.

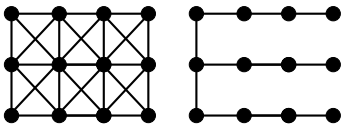


Figure 6: *Left*: 8-connected graph. *Right*: Sub-tree $\mathcal{T}$.

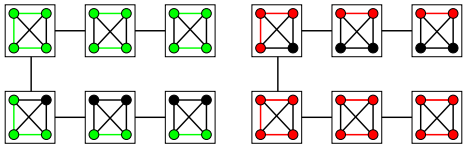


Figure 7: $\mathcal{T}_{2\times2}$ contains two copies (red and green) of $\mathcal{T}$.

Consider for example the sub-tree $\mathcal{T}$ in Figure (6). We can form a similar sub-tree $\mathcal{T}_{2\times2}$ using the super nodes, see Figure 7. Note that the edges that occur twice within the super nodes have half the weight of the corresponding edges in Figure 6. Looking in the super nodes and considering the consistency edges we see that we can find two instances of $\mathcal{T}$ within $\mathcal{T}_{2\times2}$ (see Figure 7) both with weights 1/2 (here the edges that have weight 1 are allowed to belong to both trees). Hence if we view these two instances as independent and optimize them we get the same energy as optimization over $\mathcal{T}$ would give. In addition there are other edges present in $\mathcal{T}_{2\times2}$, and therefore this tree gives a stronger bound.

In a similar way, we can construct even stronger trees by increasing the patch size further (event though the interactions might already be contained in the patches). If we group super nodes in a sliding window approach we obtain a graph with 3×3 patches, see Figure 9. (We refer to the new nodes as super-duper nodes.) If we keep repeating this process we will eventually end up enumerating the entire graph, so it is clear that the resulting lower bound will eventually approach the optimum.

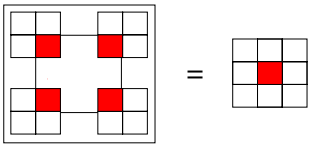


Figure 9: Super-duper nodes containing patches of size 3×3 are created by grouping super nodes of size 2×2 in groups of 2×2 in a sliding window fashion.

In Table 1 the same problem as in Figure 3 is solved using TRW-S without forming any super nodes.

3.4. Application to $\pi/4$ and $\pi/8$ precision curvature

For patches of size 2×2 it is only possible to encourage horizontal and vertical boundaries. Indeed, along a diagonal boundary edge all 2×2 patches look like the second patch in Figure 4. To make the model more accurate and include directions that are multiples of $\pi/4$ radians we will look at patches of a larger size, see Figure 1(c).

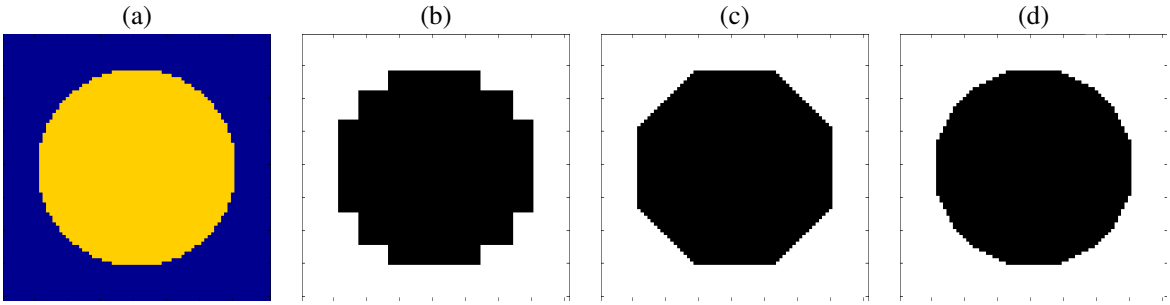


Figure 8: Segmentation results on a 81×81 pixel image using different patch sizes with same very high regularization weight ($\lambda = 20$). (a) - Data term, (b) - 2×2 patches clearly favors horizontal and vertical boundaries , (c) - 3×3 patches, favors directions that are multiples of $\pi/4$, (d) 5×5 patches, favors directions that are multiples of $\pi/8$.

λ	Energy	Lower bound
$2.5 \cdot 10^{-5}$	4677	4677
$2.5 \cdot 10^{-4}$	4680	4680
$2.5 \cdot 10^{-3}$	4709	4705
$2.5 \cdot 10^{-2}$	5441	4501
$2.5 \cdot 10^{-1}$	16090	-16039
$2.5 \cdot 10^{0}$	15940	-19990

Table 1: Same as in Figure 3 without super nodes.

For multiples of $\pi/4$ radians it is enough to have 3×3 patches and for $\pi/8$ radians we use patches of size 5×5. However, the number of distinct patch-labels needed to encode the interactions (transitions between directions) is quite high. It is not feasible to determine their costs by hand.

To compute the specific label costs we generate representative windows of size slightly larger than the patch (for 3×3 patches we use 5×5 windows) that contain either a straight line or a transition between two directions of known angle difference. From this window we can determine which super node assignments occur in the vicinity of different transitions. We extract all the assignments and constrain their sum, as shown in Figure 1, to be the known curvature of the window. Furthermore, we require that the cost of each label is positive. If a certain label is not present in any of the windows we do not allow this assignment. This gives us a set of linear equalities and inequalities for which we can find a solution (using linear programming). The procedure gives 122 and 2422 labels for the 3×3 and 5×5 cases respectively. More details are given in [22].

Figures 8 illustrates the properties of the different patch sizes. Here we took an image of a circle and segmented it using the 3 types of approximating patches. Note that there is no noise in the image, so simply truncating the data term would give the correct result. We segment this image using a very high regularization weight ($\lambda = 20$). In (b) horizontal and vertical boundaries are favored since these have zero regularization cost. In (b) and (c) the number of directions with zero cost is increased and therefore the approximation improved with the patch size. Figure 10 shows real segmentations with the different patch sizes (with $\lambda = 1$). Table 2 shows energies, lower bounds and execution times for a couple of methods. Note that all methods except GTRW-S use our super node construction, here we use the single separator implementation [14]. Thus, GTRW-S solves a weaker relaxation of the energy (this is confirmed by the numbers in Table 2). GTRW-S requires specifying all label combinations for each factor. For the patch assignments that we do not use to model curvature we specify a large cost (1000) to ensure that these are not selected. Furthermore, TRW-S and Loopy belief propagation (LBP) both use our linear time message computation. For comparison TRW-S (g) uses general message computation. All algorithms have an upper bound of 10,000 iterations. In addition, for TRW-S and GTRW-S the algorithm converges if the lower bound stops increasing. For MPLP [29] we allow 10,000 iterations of clustering and we stop running if the duality gap is less than 10^{-4}. Figure 11 shows convergence plots for the 2×2 case.

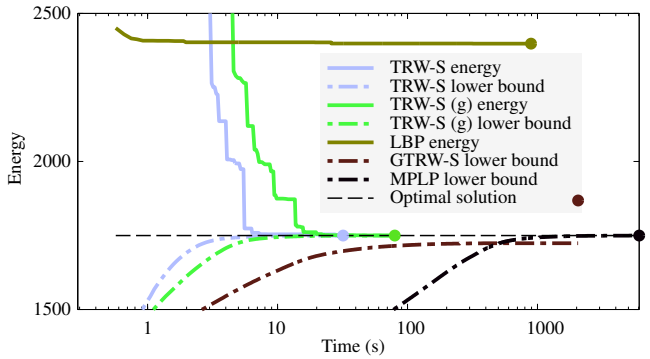


Figure 11: Logarithmic time plot for energy and lower bound over time for the 2×2 experiment in Table 2. For MPLP and GTRW-S we only show the final energy as a dot.

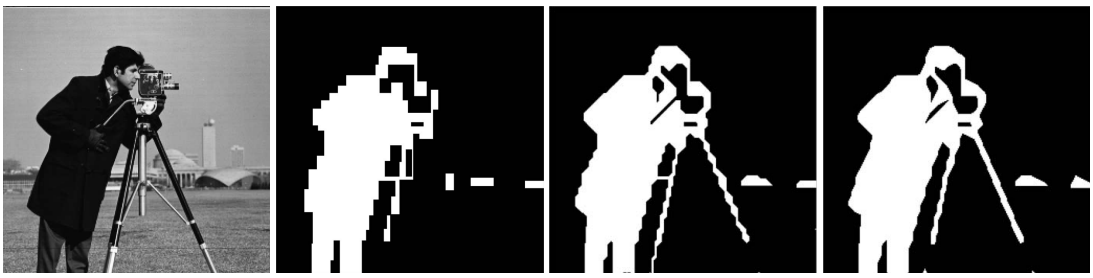


Figure 10: Segmentation of the camera man using (from left to right) 2×2, 3×3 and 5×5 patches with $\lambda = 1$.

	Energy	Lower bound	Time (s)
TRW-S	1749.4	1749.4	21
TRW-S (g)	1749.4	1749.4	1580
MPLP	1749.4	1749.4	6584
LBP	2397.7		1565
GTRW-S	1867.9	1723.8	2189

(a) 2×2 patches.

	Energy	Lower bound	Time (s)
TRW-S	1505.7	1505.7	355
TRW-S (g)	1505.7	1505.7	41503
MPLP	‡	‡	‡
LBP	∗		3148
GTRW-S	99840	1312.6	10785

(b) 3×3 patches.

	Energy	Lower bound	Time (s)
TRW-S	1417.0	1416.6	8829
TRW-S (g)	‡	‡	‡
MPLP	‡	‡	‡
LBP	∗		157532
GTRW-S	‡	‡	‡

(c) 5×5 patches.

Table 2: Cameraman (256×256 pixels) with $\lambda = 1$ run with with different path sizes. Resulting segmentation can be seen in Figure 10. (‡) Creating the problem instance not feasible due to excessive memory usage. ($*$) Inconsistent labeling.

4. Other Applications

Our framework does not only work for curvature but applies to a general class of problems. In this section we will test our partial enumeration approach for other problems than curvature regularization.

4.1. Binary Deconvolution

Figure 12 (a) shows an image convolved with a 3×3 mean value kernel with additional noise added to it. The goal is to recover the original (binary) image. We formulate the energy as outlined in [24]. The resulting interactions are pairwise and Figure 12 (b) shows the solution obtained using RD, here the gray pixels are unlabeled. For comparison we also plotted the solution obtained when solving the the same problem as RD but with [29] (c) and TRW-S (d). For these methods there are substantial duality gaps. In contrast (e) shows the solution obtained when forming super nodes with patches of size 3×3 and then applying TRW-S. Here there is no duality gap, so the obtained solution is optimal.

4.2. Stereo

In this section we optimize the energies occurring in Woodford *et al*. [34]. The regularization of this method penalizes second order derivatives of the disparity map, either using a truncated ℓ_1- or ℓ_2-penalty. The 2nd derivative is estimated from three consecutive disparity values (vertically or horizontally), thus resulting in triple interactions.

To solve this problem [34] uses fusion moves [17] where proposals are fused together to lower the energy. To compute the move [34] first reduces the interactions (using auxiliary nodes) and applies Roof duality (RD) [25]. In contrast we decompose the problem into patches of size 3×3, that contain entire triple interactions. Since the interactions will occur in as many as three super nodes we weight these so that the energy does not change.

Table 3 shows the results for the `Cones` dataset from [26] when fusing "SegPln" proposals [34]. Starting from a randomized disparity map we fuse all the proposals. To ensure that each subproblem is identical for the two approaches, we feed the solution from RD into our approach before running the next fusion. We also tested the "improve" heuristic [25] which gave a reduction in duality gap for RD. Running "probe" [25] instead of improve is not feasible due to the large number of unlabeled variables.

We also compared the final energies when we ran the methods independent of each other (not feeding solutions into our approach). For ℓ_1 regularization our solution has 0.82% lower energy than that of RD with "improve" and for ℓ_2 regularization our solution is 7.07% lower than RD with "improve".

References

[1] F. Bacchus, X. Chen, P. van Beek, and T. Walsh. Binary vs. non-binary constraints. *Artificial Intelligence*, 140(1/2):1–37, 2002. 4

[2] Y. Boykov and V. Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *International Conference on Computer Vision (ICCV)*, 2003. 2

[3] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transations on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001. 1

[4] K. Bredies, T. Pock, and B. Wirth. Convex relaxation of a class of vertex penalizing functionals. *J. Math. Imaging and Vision*, 47(3):278–302, 2013. 1, 2

(a) Noisy image (b) RD (379.44) (c) MPLP (324.07) (d) TRW-S (36.44) (e) TRW-S patches (12.11)

Figure 12: Binary deconvolution results (with energy). (a) Noisy image. (b) Gray pixels are unlabeled. Duality gap: 477.78 (unlabeled set to 0) (c) Duality gap: 422.40, maximum 10,000 clusterings. (d) Duality gap: 134.77. (e) Duality gap: 10^{-14}.

	Energy	Lower bound	Relative gap	Time (s)
Our	$1.4558 \cdot 10^{10}$	$1.4558 \cdot 10^{10}$	$1.4471 \cdot 10^{-14}$	315.3342
RD	$1.4637 \cdot 10^{10}$	$1.4518 \cdot 10^{10}$	$9.3694 \cdot 10^{-3}$	1.9216
Our/RD	0.9958	1.0019	$4.3825 \cdot 10^{-13}$	180.6859

(a) ℓ_1 regularization.

	Energy	Lower bound	Relative gap	Time (s)
Our	$1.3594 \cdot 10^{10}$	$1.3594 \cdot 10^{10}$	$2.0394 \cdot 10^{-14}$	428.2557
RD	$1.5165 \cdot 10^{10}$	$1.0484 \cdot 10^{10}$	0.5756	4.6479
Our/RD	0.9092	1.1652	$6.0910 \cdot 10^{-15}$	111.8597

(b) ℓ_2 regularization.

Table 3: Averaged stereo results on `Cones` sequence. Relative gap is defined as (Energy-Lower bound)/Lower bound. (a) For ℓ_1 regularization RD left 24% of the variables unlabeled. "Improve" lowered the average energy for RD to $1.4609 \cdot 10^{10}$. (b) For ℓ_2 regularization RD left 64% of the variables unlabeled. "Improve" lowered the average energy for RD to $1.4392 \cdot 10^{10}$.

[5] T. Chan and J. Shen. Nontexture inpainting by curvature driven diffusion (cdd). *Journal of Visual Communication and Image Representation*, 12:436–449, 2001. 1

[6] M. Droske and M. Rumpf. A level set formulation for Willmore flow. *Interfaces and Free Boundaries*, 6:361–378, 2004. 1

[7] N. El-Zehiry and L. Grady. Fast global optimization of curvature. In *Conf. Computer Vision and Pattern Recognition*, 2010. 1, 2, 4, 5

[8] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transations on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984. 1, 2

[9] A. Globerson and T. Jaakkola. Fixing max-product: Convergent message passing algorithms for MAP LP-relaxations. In *NIPS*, 2007. 3

[10] H. Ishikawa. Exact optimization for markov random fields with convex priors. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 25(10):1333 – 1336, 2003. 1

[11] F. Kahl and P. Strandmark. Generalized roof duality. *Discrete Applied Mathematics*, 160(16-17):2419–2434, 2012. 3, 4, 5

[12] M. Kass, A. Witkin, and D. Terzolpoulos. Snakes: Active contour models. *Int. Journal of Computer Vision*, 1(4):321–331, 1988. 1

[13] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Transanctions on Pattern Analysis and Machine. Intelligence.*, 28:1568–1583, October 2006. 3, 5

[14] V. Kolmogorov and T. Schoenemann. Generalized sequential tree-reweighted message passing. *arXiv:1205.6352*, 2012. 3, 6

[15] N. Komodakis and N. Paragios. Beyond pairwise energies: Efficient optimization for higher-order mrfs. In *Conf. on Computer Vision and Pattern Recognition*, 2009. 3, 4

[16] J. Lellmann and C. Schnorr. Continuous multiclass labeling approaches and algorithms. *SIAM Journal on Imaging Sciences*, 4:1049–1096, 2011. 1

[17] V. S. Lempitsky, C. Rother, S. Roth, and A. Blake. Fusion moves for markov random field optimization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(8):1392–1405, 2010. 7

[18] S. Masnou and J. Morel. Level-lines based disocclusion. In *International Conference on Image Processing (ICIP)*, 1998. 1

[19] T. Meltzer, A. Globerson, and Y. Weiss. Convergent message passing algorithms - a unifying view. In *Conf. on Uncertainty in Artificial Intelligence*, 2009. 3

[20] M. Nikolova, S. Esedoglu, and T. Chan. Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM Journal of Applied Mathematics*, 66:1632–1648, 2006. 1

[21] C. Olsson and Y. Boykov. Curvature-based regularization for surface approximation. In *Conf. Computer Vision and Pattern Recognition*, 2012. 2

[22] C. Olsson, J. Ulén, Y. Boykov, and V. Kolmogorov. Simplifying energy optimization using partial enumeration. *Technical report, arXiv:1303.1749v2*, 2013. 3, 4, 6

[23] T. Pock, D. Cremers, H. Bischof, and A. Chambolle. Global solutions of variational models with convex regularization. *SIAM Journal on Imaging Sciences*, 3:1122–1145, 2010. 1

[24] A. Raj and R. Zabih. A graph cut algorithm for generalized image deconvolution. In *International Conference of Computer vision (ICCV)*, 2005. 7

[25] C. Rother, V. Kolmogorov, V. S. Lempitsky, and M. Szummer. Optimizing binary mrfs via extended roof duality. In *Conf. Computer Vision and Pattern Recognition*, 2007. 5, 7

[26] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In *Conf. Computer Vision and Pattern Recognition*, 2003. 7

[27] T. Schoenemann, F. Kahl, S. Masnou, and D. Cremers. A linear framework for region-based image segmentation and inpainting involving curvature penalization. *Int. Journal of Computer Vision*, 2012. 1, 2, 4

[28] A. Shekhovtsov, P. Kohli, and C. Rother. Curvature prior for MRF-based segmentation and shape inpaint. In *arXiv: 1109.1480v1, 2011, also DAGM*, 2012. 3

[29] D. Sontag, T. Meltzer, A. Globerson, T. Jaakkola, and Y. Weiss. Tightening lp relaxations for map using message passing. In *UAI*, 2008. 3, 6, 7

[30] P. Strandmark and F. Kahl. Curvature regularization for curves and surfaces in a global optimization framework. In *EMMCVPR*, pages 205–218, 2011. 1, 2, 4

[31] J. Sullivan. A crystalline approximation theorem for hypersurfaces, phd thesis. 1992. 2

[32] D. Tarlow, I. Givoni, and R. Zemel. HOP-MAP: efficient message passing with higher order potentials. In *AISTATS*, 2010. 4

[33] T. Werner. Revisiting the linear programming relaxation approach to Gibbs energy minimization and weighted constraint satisfaction. *IEEE Transations on Pattern Analysis and Machine Intelligence*, 32(8):1474–1488, 2010. 3

[34] O. Woodford, P. Torr, I. Reid, and A. Fitzgibbon. Global stereo reconstruction under second order smoothness priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12):2115–2128, 2009. 1, 7

[35] 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)*, 2010. 1