

Fast Global Optimization of Curvature

Noha Youssry El-Zehiry
Siemens Corporate Research
Princeton, NJ

noha.el-zehiry.ext@siemens.com

Leo Grady
Siemens Corporate Research
Princeton, NJ

leo.grady@siemens.com

Abstract

*Two challenges in computer vision are to accommodate noisy data and missing data. Many problems in computer vision, such as segmentation, filtering, stereo, reconstruction, inpainting and optical flow seek solutions that match the data while satisfying an additional regularization, such as total variation or boundary length. A regularization which has received less attention is to minimize the curvature of the solution. One reason why this regularization has received less attention is due to the difficulty in finding an optimal solution to this image model, since many existing methods are complicated, slow and/or provide a suboptimal solution. Following the recent progress of Schoenemann *et al.* [28], we provide a simple formulation of curvature regularization which admits a fast optimization which gives globally optimal solutions in practice. We demonstrate the effectiveness of this method by applying this curvature regularization to image segmentation.*

1. Introduction

Recent progress in continuous and discrete optimization techniques has yielded techniques for finding optimal or nearly optimal solutions for several of the classical models in computer vision. These models include boundary length regularization, total variation and the Mumford-Shah functional. An important consequence of this success in optimization is that any inadequacies of the models can no longer plausibly be attributed to suboptimal solutions but must rather lie with inadequacies of the models themselves. One inadequacy observed in many of the existing models is a failure for the models to enforce *continuity* of an object boundary in the presence of noise, occlusion or missing data [29]. Curve continuity is a feature of the Gestalt description of human visual perception and is therefore likely to be an important feature of any successful computer vision model. This lack of curve continuity in existing models has a negative effect on the use of these models in the context of image segmentation, inpainting and filtering [29].

One common proposal for enforcing curve continuity is to use a regularization term that penalizes the *curvature* of a solution. Mumford described the minimum curvature regularization by the Euler elastica model [22] which he showed to embody the Bayes optimal solution of a model of curve continuity. Although Mumford and other authors have convincingly justified the elastica model for enforcing curve continuity, the optimization of the elastica regularization term has proven difficult, leading to algorithms that provide suboptimal solutions, require implementation parameters and are generally computationally inefficient.

A feature of many optimization successes in recent years is the use of combinatorial optimization techniques to solve a discretized form of models which are written using classical continuous mathematics. This discretization approach has also been applied to the elastica model. Bruckstein *et al.* [7] gave a discretized form of the elastica model which described the object boundary by a polygon for which the curvature could be estimated using the exterior angles of the successive line segments of the polygon. The outstanding contribution of Schoenemann *et al.* [28] showed how the Bruckstein discretization could be incorporated into a combinatorial optimization framework by viewing the *dual* graph as a space of polygonal boundaries over which the optimization could be performed. However, this parameterization of the Bruckstein discretization required a relaxation of the indicator function of the boundary segments, often leading to a suboptimal solution for the curvature. Although this solution for the curvature regularizer was suboptimal, it still had strong advantages over previous optimization techniques in the sense that the optimization was parameter-free and did not depend on initialization.

Our work builds directly on the Bruckstein *et al.* and Schoenemann *et al.* formalism to produce a method for optimizing curvature. As with Schoenemann *et al.*, we adopt the Bruckstein discretization of the elastica model, but instead parameterize the space of curves based on the *primal* graph in which each edge cut in the primal graph represents a normal vector of the curve in the dual graph. By reformulating Bruckstein's discretization on the angles between

edges of the primal graph, we give a formulation of the elastica model in which the value of each graph cut corresponds to the sum of squared curvature of the cut boundary.

Although this formulation equates graph cuts with boundary curvature, a difficulty of the formulation is the presence of *nonsubmodular* terms in the graph cut (i.e., negative edge weights). Fortunately, the optimization of graph cuts with nonsubmodular terms has been a focus of recent activity in computer vision and has led to a powerful set of methods which are provably optimal [18, 25]. Specifically, the Quadratic Pseudo-Boolean Optimization (QPBO) method [2, 3, 18] has been shown to provide an optimal solution for some variables and to leave other variables with unknown solution. However, by using QPBO iteratively (with *probing*) it was shown that a drastic reduction of the number of unknown variables is feasible [25]. Furthermore, the QPBO method with probing (QPBO-P) preserves global optimality of the solution, although some problems yield a fast convergence and other problems yield a slow convergence. Empirically, our discrete formulation of the elastica model always yields a fast convergence using QPBO-P and in our experiments all (or nearly all) of the variables are labeled, meaning that the solution we obtain is globally optimal.

1.1. Previous work

The elastica model has previously been employed by several research groups for the perceptual completion and image inpainting problems [10, 29, 13, 20], in which the model was optimized in a PDE framework. These implementations were often difficult or computationally inefficient and an appropriate initialization is required. Curvature regularization has also appeared in methods for perceptual grouping and capturing illusory contours in which curvature regularization was demonstrated to provide results closer to human perception than any other model [31, 24].

Curvature regularization has appeared in several guises in the computer vision literature, of which Mumford’s elastica model is just one example. Other approaches use cycle ratios to provide curvature dependent image segmentation [27, 17]. Schoenemann *et al.* presented globally optimal image segmentation by minimizing the ratio of the flux over the weighted sum of length and curvature of the object of interest. However, the memory requirements and the computational time of this approach makes it impractical for many vision applications.

Several groups have also applied advanced optimization techniques to solve traditional models in computer vision. For example, the total variation model [26] has been well optimized using both continuous [8, 1, 30] and discrete methods [11]. Similarly, progress in the Mumford-Shah model has been also appeared from improved optimization of the continuous [6, 14] and discrete forms [12, 15]. The

QPBO optimization that we depend on in this work was developed originally by Boros and Hammer [2, 3], but later brought to computer vision and further extended by Rother and Kolmogorov [18, 25].

2. Methods

We begin this section by reviewing the elastica model and the Bruckstein discretization. We then demonstrate how to formulate this model on the primal graph, which leads to an identification of cut cost with boundary curvature. The optimization of this model is then discussed.

Mumford’s elastica curve model is defined as [22]:

$$E(C) = \int_C (a + b\kappa^2) ds \quad a, b > 0, \quad (1)$$

where κ denotes scalar curvature and ds represents the arc length element. When $a = 0$ (the boundary length is ignored), the elastica model reduces to the integral of the boundary squared curvature

$$e(C) = \int_C b\kappa^2 ds \quad b > 0. \quad (2)$$

Bruckstein *et al.* [7] introduced a discrete formulation for the curvature by measuring the curvature of a polygon and showing that this discretization approaches the continuous formulation as the length of each line segment approaches zero. Given a boundary (polygonal curve) consisting of n segments $\vec{l}_1, \vec{l}_2, \dots, \vec{l}_n$ of length l_1, l_2, \dots, l_n , respectively, Bruckstein *et al.* estimated the integral curvature (raised to a power p) over the boundary as [7]

$$\sum_{i=1}^{n-1} \frac{\phi_{i,i+1}^p}{(\min(l_i, l_{i+1}))^{p-1}}, \quad (3)$$

where $\phi_{i,i+1}$ is the exterior angle formed by the intersection of \vec{l}_i and \vec{l}_{i+1} as depicted in Figure 1. Notice that the Bruckstein’s formula is general for any exponent p , but throughout this paper, we will use $p = 2$ to match the elastica model as defined by Mumford.

Our strategy for employing Bruckstein’s discretization consists of two steps. In the first step, we replace Bruckstein’s polygonal line segments by the edges cut in a graph representing the image pixels. In the second step we map the space of polygons to edge cuts such that the measure of curvature energy for each polygon, as measured by Bruckstein’s formula (3), is equal to the cost of a cut. In order to present our formulation of Bruckstein’s discrete elastica formulation, we introduce a graph specific notation. A graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ consists of a set of vertices $v \in \mathcal{V}$ and a set of edges $e \in \mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. An edge incident to vertices v_i and v_j is denoted e_{ij} . In our formulation, each pixel is identified with a node, v_i . A weighted graph is a graph in

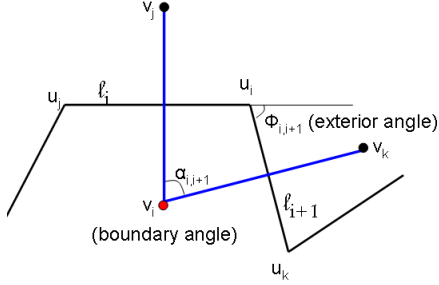


Figure 1. The exterior angle ϕ formed by the adjacent line segments in a polygonal curve and the interior angle α formed by the intersection of the normals to the line segments. Note that $\phi = \alpha$. which every edge e_{ij} is assigned a weight w_{ij} . An edge cut is any collection of edges that separates the graph into two sets, $\mathcal{S} \subseteq \mathcal{V}$ and $\bar{\mathcal{S}}$, which may be represented by a binary indicator vector x ,

$$x_i = \begin{cases} 1 & \text{if } v_i \in \mathcal{S}, \\ 0 & \text{else.} \end{cases} \quad (4)$$

The cost of the cut represented by any x is defined as

$$\text{Cut}(x) = \sum_{e_{ij}} w_{ij} |x_i - x_j|. \quad (5)$$

To formulate the curvature optimization problem on the primal graph, we first adopt the Schoenemann *et al.* formalism of the boundary polygonal as existing on the *dual* graph. In this case, exterior angles of the line segments (edges) of the dual graph correspond to the interior angles ($\alpha_{i,i+1}$) formed by the intersection of the dual boundary edges with the primal cut edges, as shown in Figure 1. To have a well-defined dual, we begin by considering our pixel lattice as a 4-connected graph, in which the dual lattice is also a 4-connected graph [28]. It is obvious that $\phi_{i,i+1} = \alpha_{i,i+1}$ and, for image lattices, it is straightforward to show that $\triangle u_j u_i u_k$ and $\triangle v_j v_i v_k$ are similar. Hence we may adapt Bruckstein's formula in (3) to our primal formulation by penalizing the edges with angle $\alpha = \angle v_j v_i v_k$ by the weight w_{ijk} given by

$$w_{ijk} = \frac{\alpha^p}{\min(\|\vec{v_i v_j}\|, \|\vec{v_i v_k}\|)^{p-1}}. \quad (6)$$

Every exterior angle that contributes to the calculation of the integral curvature proposed by Bruckstein *et al.* has a corresponding interior angle in which edges should be cut to partition the domain in two disjoint sets. This concept is illustrated in Figure 2.

When the successive line segments of every polygon in the dual graph are either parallel to each other or dual to edges incident on a single node, v_i , then the angle between two dual line segments can be represented as a function of three nodes in the primal graph, v_i, v_j and v_k . Specifically, we can write the contribution of the angle of two successive line segments in the dual graph in terms of the indicator

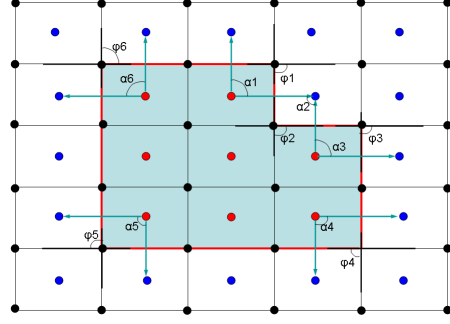


Figure 2. The correspondence between calculating the curvature using exterior angles between the boundary line segments and using the interior angles between the cut edges. Blue and red vertices are the vertices of the primal graph. Black vertices are the vertices of the dual graph.

function x as

x_i	x_j	x_k	w
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	w_{ijk}
1	0	0	w_{ijk}
1	0	1	0
1	1	0	0
1	1	1	0

Since this formulation for the curvature is described in terms of three variables, we define a 3-clique with these penalties as the *curvature clique*. Remarkably, the curvature clique can be decomposed into three 2-cliques (edges) having weights

$$\begin{aligned} w_{ij} &= \frac{1}{2} w_{ijk}, \\ w_{ik} &= \frac{1}{2} w_{ijk}, \\ w_{jk} &= -\frac{1}{2} w_{ijk}. \end{aligned} \quad (8)$$

Therefore

$$E(x_i, x_j, x_k) = w_{ij} |x_i - x_j| + w_{ik} |x_i - x_k| + w_{jk} |x_j - x_k|, \quad (9)$$

where $E(x_i, x_j, x_k)$ represents the energy defined in (7). Even though the curvature clique was designed to penalize the cut of both edges e_{ij} and e_{ik} , the decomposition in (8) effectively adds an edge e_{jk} having negative weight. We denote this new set of effective edges which have nonzero weights as \mathcal{E}^* . Note that $\mathcal{E} \subseteq \mathcal{E}^*$ where \mathcal{E} .

Therefore, the weights for every curvature clique may be computed for each successive pair of edges incident on every node (taken clockwise). If each curvature clique is further decomposed into pairwise edge weights via (8), then





				
Boundary				
Curvature calculated using [21]	340	404	414	630
Cut cost using our formulation	305.9577	365.1754	381.2135	592.1763

Figure 3. A comparison of our cut formulation of Bruckstein’s curvature discretization with an independent estimation of curvature given by Meyer [21] which is widely used in computer graphics. For each shape, we used Meyer’s formula to calculate the integral of squared boundary curvature and compared it with the cost of the cut we used to calculate the integral of the squared boundary curvature. In each case, our estimates of boundary curvature closely match each other, demonstrating that Bruckstein’s formulation of polygonal curvature (and our cut-based reformulation) is very similar to a different method for calculating boundary curvature proposed by Meyer.

the cost of the cut in (5) equals the sum of the curvature weights computed for angles in the dual polygon, as given by Bruckstein *et al.* In this sense, our formulation represents the *primal* form of the Schoenemann *et. al* formalism.

As an example of the correctness of our formulation, we compared the cost of the cut in our formulation (i.e., estimated curvature of the boundary) with an independent method for calculating the integral of the curvature of the boundary of an object given by Meyer [21] and widely used in computer graphics. Meyer provided a simple discrete approximation for the integral mean curvature at a particular vertex v_i as

$$\frac{1}{2} \sum_{j \in N_1(v_i)} (\cot \gamma_{ij} + \cot \beta_{ij}) \|v_i v_j\|, \quad (10)$$

where $N_1(v_i)$ is the set of 1-ring neighbor vertices of vertex i , and γ_{ij} and β_{ij} are the two angles opposite to the edge $v_i v_j$ in the two triangles sharing this edge and $\|v_i v_j\|$ is the Euclidean length of the line segment connecting v_i and v_j .

We simulated several objects by combining circular, elliptic, parabolic and hyperbolic functions. For each object, we have calculated the integral curvature on the boundary of these objects using Meyer’s formula (10) and compared this calculation with the cost of the cut using the proposed graph construction. Figure 3 depicts the simulated objects that we used in the validation, Meyer’s calculation of the integral squared curvature of the object boundary and our calculation of the integral of the squared curvature obtained by the cut cost. This example demonstrates that our cut formulation of Bruckstein’s discretization of curvature closely matches an independent measure of boundary curvature commonly used in the computer graphics literature.

2.1. Optimization

In our primal formulation of Bruckstein’s curvature discretization, we provided a graph weighting such that the integral of squared curvature of the boundary curve equals

the cost of the cut. Therefore, in order to find the boundary with minimal curvature (subject to some constraints and/or data observation) it is necessary to find the minimum cut in our weighted graph. Efficient algorithms exist for finding a minimum cut when the edge weights have nonnegative values. However, the presence of negative weights causes our minimum cut problem to be *nonsubmodular* [19] which means that a straightforward max-flow computation will not yield a minimum cut.

Recently, minimizing nonsubmodular functions has been the center of attention of several research groups [18, 3, 25]. Progress in the optimization of nonsubmodular functions by Boros and Hammer led to the development of a technique called Quadratic Pseudo Boolean Optimization (QPBO) [16, 2]. Optimization with QPBO was later brought to computer vision by Kolmogorov and Rother [18] who demonstrated applications in texture restoration and image stitching. The QPBO technique has the amazing ability to provide a partial labeling of the variables which is *optimal* for all labeled variables. The output of QPBO is

$$x_i = \begin{cases} 1 & \text{if } v_i \in \mathcal{S}, \\ 0 & \text{if } v_i \in \overline{\mathcal{S}}, \\ \emptyset & \text{otherwise.} \end{cases} \quad (11)$$

Recall that \mathcal{S} represents the set of nodes for which we are computing the curvature of the boundary.

The utility of QPBO is determined by the number of vertices that the approach fails to label. Experimentally, it has been verified [25] that if the number of non submodular terms in the energy function is large, the output of QPBO contains many unlabeled vertices (Rother *et al.* [25] reported that up to 99.9% of variables could be undetermined). A promising approach to resolve this problem is the extended roof duality presented in [3]. Rother *et al.* [25] also reviewed the extended roof duality approach, first introduced by Boros *et al.* [3], and presented a more efficient implementation than Boros’. Their work extends

QPBO by a probing operation that aims at calculating the global minimum for the vertices that have not been assigned a label by QPBO. The approach is referred to as QPBOP (Quadratic Pseudo Boolean Optimization with Probing). By applying QPBO multiple times to our energy function, the QPBOP eventually labels almost all variables, providing a globally optimal solution for the energy minimum. When QPBOP did fail to label some pixels, we found that these pixels were isolated in a few small connected components. In these situations, we adopted the method of assigning to each connected component the label that best lowers the total energy. Therefore, we are able to apply the QPBOP method to effectively find a minimum cut (minimum curvature boundary) even though our construction contains negative weights. The only potential concern with this approach is the number of iterations required for QPBOP to converge (computation time). In practice, we find that QPBOP converges quickly for our problem.

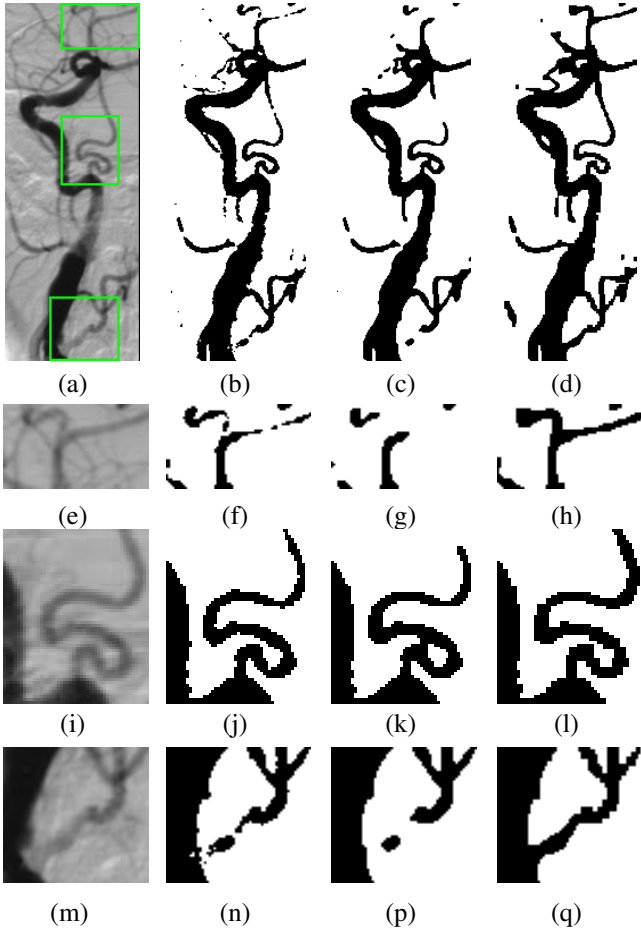


Figure 4. Comparison of a segmentation obtained by length regularization and curvature regularization. (a) An angiogram, (b) Segmentation obtained by data alone, (c) Segmentation obtained by data and length regularization, (d) Segmentation obtained by data and curvature regularization. The second, third and fourth rows are detailed views of highlighted parts of the image. Results were obtained with $\mu = 1000$.

3. Image Segmentation with Curvature Regularity

Image segmentation is an important application that can benefit from curvature regularity. Schoenemann *et al.* [28] showed that curvature regularization can particularly enhance the segmentation of elongated or thin structures. The most natural method for applying a curvature regularization is by including a region-based data model. Specifically, a region-based image segmentation model consists of a data term which penalizes the smoothness of the data points and a regularizer that enforces boundary smoothness. The most common boundary regularizer is to minimize the boundary length [23, 9, 1, 30]. For simplicity, we adopt the piecewise constant data model [23, 9].

Given a 2D image with image values associated with each pixel (node), $g : \mathcal{V} \rightarrow \mathcal{R}$, our goal in a segmentation application is to find the boundary, represented by indicator function x , that subdivides the image into a foreground (the object of interest) and a background. This goal can be achieved by minimizing the discrete Mumford-Shah-Euler segmentation energy given by

$$\begin{aligned}
 F_{\text{MSE}}(x) &= E_{\text{data}}(x) + \mu E_{\text{curvature}}(x) \\
 &= \sum_{v_i \in \mathcal{V}} x_i (g_i - \mu_F)^2 + \sum_{v_i \in \mathcal{V}} (1 - x_i) (g_i - \mu_B)^2 \\
 &\quad + \mu \sum_{e_{ij} \in \mathcal{E}^*} w_{ij} |x_i - x_j|, \quad (12)
 \end{aligned}$$

where μ is a coefficient that control the relative importance of each term. The values of μ_F and μ_B represent the mean intensity values inside and outside the set \mathcal{S} represented by x . The first two terms represent the Mumford-Shah energy functional [23], which reflect the data smoothness by minimizing the intensity variations in each region. The third term imposes a regularization of the boundary by minimizing the integral curvature along the boundary. Note that the data term can take many other forms (see [4, 28] for more possibilities) and boundary length may additionally incorporated into our optimization by adding the edge weights formulated to encode Euclidean boundary length presented in [5]. In fact, adding the positive edge weights that encode boundary length serve to make the optimization easier by reducing the number of edges with negative weight. Finally, interactively placed foreground/background seeds can easily be incorporated into our optimization by treating foreground seeds as source nodes and background seeds as sink nodes, in exactly the same way as traditional graph cuts [4].

3.1. Summary of the algorithm

The segmentation algorithm can be summarized as follows:



Figure 5. Effect of edge connectivity. Top row: Input image, segmentation by data alone, segmentation with 4-connected curvature regularization, segmentation with 8-connected curvature regularization. Bottom rows: Detailed views of the segmentation. Results were obtained with $\mu = 500$.

1. For every node, compute the weights of the curvature clique for each successive pair of edges incident on the node (taken clockwise). Each curvature clique is further decomposed into pairwise edge weights via (8) to form the new set of edges \mathcal{E}^* .
2. Apply Quadratic Pseudo Boolean Optimization with Probing [25] to find the solution x producing an optimum of (12). The min cut partitions the graph into two components \mathcal{S} and $\bar{\mathcal{S}}$.

4. Experimental Results

This section presents the experimental results for the image segmentation. Following Schoenemann *et al.* [28], we fixed the values of μ_F and μ_B in equation (12) based on the minimal and maximal intensities in the image. All of the experiments were performed on a 2.8 GHz Intel machine with 1GB RAM. In all of our experiments, we endeavored to choose parameters to display the best segmentations we could obtain by the data term alone or the segmentation obtained by the data term regularized with boundary length.

We begin by demonstrating the usefulness of curvature regularization by examining the segmentation of an angiography image in which the blood vessels have weak contrast at locations. Figure 4 displays the segmentation obtained by data modeling only, the traditional boundary length regularization and the curvature regularization. The segmentation obtained by data modeling alone leads to a fragmented, disconnected segmentation. In this example, the boundary length regularization has the effect of removing small pieces

of vessel or shortcutting torturous regions of the vessel in an effort reduce boundary length. In contrast, the curvature regularization connects small, weak, co-aligned fragments and avoids shortcutting the boundary.

A common feature of graph-based algorithms formulated on a 4-connected graph is a bias toward axis-aligned structures. However, the introduction of a larger neighborhood connectivity usually has the effect of ameliorating this bias [5, 28]. Unfortunately, the 8-connected lattice is non-planar. Therefore, there exists no dual graph which may be equated to Bruckstein's formulation (and [28]). However, for any lattice, we can still order the edges incident to a particular node and use (3) to penalize cutting these edge pairs. We can then use the pairwise decomposition of the resulting curvature clique in (8) to establish a pairwise-weighted graph that serves as the curvature regularization. In this manner, we applied our formulation of curvature in terms of the angles of edge pairs around a node to lattices with 8-connectivity. Empirically, the 8-connected lattice improves the metrication artifacts sometimes observed when using a 4-connected construction. Meanwhile, an 8-connected construction allows us to capture diagonal curve continuity for cases in which the 4-connected lattice has difficulty. Figure 5 illustrates this effect.

Figure 6 shows a collection of results that illustrate the ability of curvature regularization to complete co-linear structures even in the presence of substantial noise or data weakness. Two examples are also given of interactive segmentation with seeds using curvature regularization. The segmentation of blood vessels in the presence of steno-

sis (occlusion) is a major difficulty in medical imaging for which curvature regularization can be very successful. In our experiments, QPBO succeeded to label all pixels when an 8-connected lattice was used. In the experiments using a 4-connected lattice, there were occasions when some small connected components were unlabeled and we therefore adopted the procedure of labeling each connected component with the label that best lowers the total energy. Generally, all of our images took less than 10 seconds to produce a segmentation with curvature regularization. This CPU time, to the best of our knowledge, is faster than any time recorded for curvature based segmentation in the literature. For example, the Schoenemann *et al.* paper [28] reported time varying from 10 minutes to 3.5 hours for the segmentation of a single image. For example, Schoenemann *et al.* segmented the fly image shown in Figure 6 in 3.5 hours while our formulation allowed us to segment the same image in 6.22 seconds, an improvement by more than four orders of magnitude.

5. Conclusion

Mumford promoted the elastica image model as paradigm for obtaining curve continuity. Later work with this model in image inpainting and segmentation demonstrated that the elastica model does promote solutions that exhibit curve continuity. However, traditional work on the elastica model produced PDE-based solutions that could be difficult to implement, slow and dependent on initialization. The recent work of Schoenemann *et al.* [28] broke the initialization dependence by using a combinatorial optimization of the discretization given by Bruckstein *et al.* [7]. However, the solution provided by Schoenemann *et al.* [28] was computationally intensive and often did not provide a global optimum of the desired objective. We built on all of this work to give a fast solution to the elastica model that far exceeds the computational requirements of previous approaches by employing recent advances in combinatorial optimization [18, 25]. Empirically, QPBO provides a solution which labels all (or almost all) pixels, providing a globally optimal solution to the elastica model.

The elastica model has held promise since Mumford introduced it to segment challenging objects by making use of curve continuity. Our optimization of this objective has demonstrated that the model does in fact allow us to meet these segmentation challenges. Future work will focus on a 3D formulation and the application of our regularization method to other problems in computer vision.

References

[1] B. Appleton and H. Talbot. Globally optimal surfaces by continuous maximal flows. *IEEE PAMI*, 28(1):106–118, Jan. 2006.

[2] E. Boros, P. L. Hammer, and X. Sun. Network flows and minimization of quadratic pseudo-boolean functions. Technical Report RRR 17-1991, RUTCOR, May 1991.

[3] E. Boros, P. L. Hammer, and G. Tavares. Preprocessing of unconstrained quadratic binary optimization. Technical Report RRR 10-2006, RUTCOR, April 2006.

[4] Y. Boykov and M.-P. Jolly. *Interactive graph cuts* for optimal boundary & region segmentation of objects in N-D images. In *Proc. of ICCV*, pages 105–112, 2001.

[5] Y. Boykov and V. Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *Proc. of ICCV*, volume 1, October 2003.

[6] X. Bresson, S. Esedoglu, P. Vanderghenst, J. Thiran, and S. Osher. Fast global minimization of the active contour/snake model. *JMIV*, 28(2):151–167, 2007.

[7] A. Bruckstein, A. Netravali, and Richardson. Epicongvergence of discrete elastica. *Applicable Analysis*, 79(1-2):137–171, 2001.

[8] A. Chambolle. An algorithm for total variation minimization and applications. *JMIV*, 20(1-2):89–97, 2004.

[9] T. F. Chan and L. A. Vese. Active contours without edges. *IEEE Trans. Imag. Proc.*, 10(2):266–277, 2001.

[10] G. Citti and A. Sarti. A cortical based model of perceptual completion in the roto-translation space. *JMIV*, 24(3):307–326, 2006.

[11] J. Darbon and M. Sigelle. A fast and exact algorithm for total variation minimization. In *IbPRIA (1)*, pages 351–359, 2005.

[12] N. El-Zehiry, S. Xu, P. Sahoo, and A. Elmaghraby. Graph cut optimization for the Mumford Shah model. In *VIIIP*, pages 182–187, 2007.

[13] S. Esedoglu and J. Shen. Digital inpainting based on the mumfordshaheuler image model. *European Journal of Applied Mathematics*, 13(4):353–370, 2002.

[14] T. Goldstein, X. Bresson, and S. Osher. Geometric applications of the split bregman method: Segmentation and surface reconstruction. Technical Report 09-06, UCLA, 2009.

[15] L. Grady and C. Alvin. The piecewise smooth Mumford-Shah functional on an arbitrary graph. *IEEE Trans. on Image Proc.*, 18(11):2547–2561, Nov. 2009.

[16] P. L. Hammer, P. Hansen, and B. Simeone. Roof duality, complementation and persistency in quadratic 0 1 optimization. *Mathematical Programming*, 28(2):121–155, 1984.

[17] I. H. Jermyn and H. Ishikawa. Globally optimal regions and boundaries as minimum ratio weight cycles. *IEEE PAMI*, pages 1075–1088, 2001.

[18] V. Kolmogorov and C. Rother. Minimizing nonsubmodular functions with graph cuts—a review. *IEEE PAMI*, 29(7):1274–1279, 2007.

[19] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE PAMI*, 26(2):147–159, 2004.

[20] S. Masnou and J.-M. Morel. Level lines based. In *ICIP (3)*, pages 259–263, 1998.

[21] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and Mathematics III*, pages 35–57. Springer-Verlag, Heidelberg, 2003.

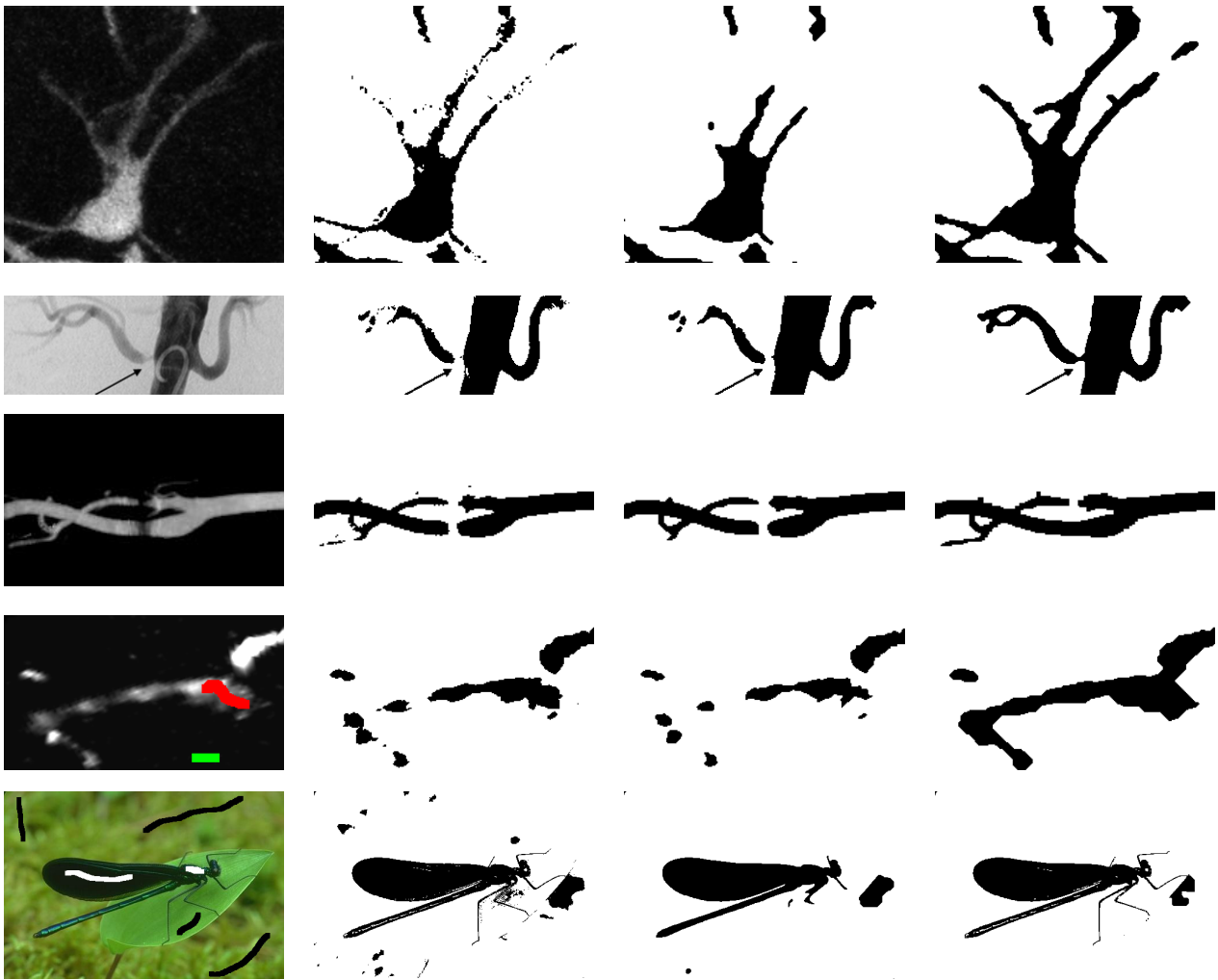


Figure 6. An illustration of the performance of our curvature regularization method as compared to traditional boundary regularization and data modeling alone. First column: Input images. Second column: Segmentation using the data term only. Third column: Segmentation with boundary regularization. Fourth column: Segmentation with curvature regularization. The bottom two images (the blood vessel in ultrasound and the insect) demonstrate how our curvature regularization method can be used for interactive segmentation by specifying foreground/background seeds.

- [22] D. Mumford. Elastica and computer vision. *Algebraic Geometry and Its Applications*, pages 491–506, 1994.
- [23] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. on Pure and App. Math.*, 42(5):577–685, 1989.
- [24] P. Parent and S. W. Zucker. Trace inference, curvature consistency, and curve detection. *IEEE PAMI*, 11(8):823–839, 1989.
- [25] C. Rother, V. Kolmogorov, V. S. Lempitsky, and M. Szummer. Optimizing binary MRFs via extended roof duality. In *CVPR*, 2007.
- [26] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Phy. D*, 60:259–268, 1992.
- [27] T. Schoenemann and D. Cremers. Introducing curvature into globally optimal image segmentation: Minimum ratio cycles on product graphs. In *ICCV*, pages 1–6, 2007.
- [28] T. Schoenemann, F. Kahl, and D. Cremers. Curvature regularity for region-based image segmentation and inpainting: A linear programming relaxation. In *ICCV*, Kyoto, Japan, 2009.
- [29] J. Shen, S. H. Kang, and T. F. Chan. Euler’s elastica and curvature-based inpainting. *SIAM Journal of Applied Mathematics*, 63(2):564–592, 2003.
- [30] M. Unger, T. Pock, W. Trobin, D. Cremers, and H. Bischof. TVSeg - Interactive total variation based image segmentation. In *BMVC*, 2008.
- [31] W. Zhu and T. Chan. A variational model for capturing illusory contours using curvature. *J. Math. Imaging Vis.*, 27(1):29–40, 2007.