

## Thin Structure Estimation with Curvature Regularization

Dmitrii Marin\*, Yuchen Zhong\*, Maria Drangova\*<sup>†</sup>, Yuri Boykov\*

\*University of Western Ontario, Canada

<sup>†</sup>Robarts Research Institute, Canada

dmitrii.a.marin@gmail.com yzhong.cs@gmail.com mdrangova@robarts.ca yuri@csd.uwo.ca

### Abstract

Many applications in vision require estimation of thin structures such as boundary edges, surfaces, roads, blood vessels, neurons, etc. Unlike most previous approaches, we simultaneously detect and delineate thin structures with sub-pixel localization and real-valued orientation estimation. This is an ill-posed problem that requires regularization. We propose an objective function combining detection likelihoods with a prior minimizing curvature of the center-lines or surfaces. Unlike simple block-coordinate descent, we develop a novel algorithm that is able to perform joint optimization of location and detection variables more effectively. Our lower bound optimization algorithm applies to quadratic or absolute curvature. The proposed early vision framework is sufficiently general and it can be used in many higher-level applications. We illustrate the advantage of our approach on a range of 2D and 3D examples.

### 1. Introduction

A large amount of work in computer vision is devoted to estimation of structures like edges, center-lines, or surfaces for fitting thin objects such as intensity boundaries, blood vessels, neural axons, roads, or point clouds. This paper is focused on the general concept of a *center-line*, which could be defined in different ways. For example, Canny approach to edge detection implicitly defines a center-line as a “ridge” of intensity gradients [6]. Standard methods for shape skeletons define medial axis as singularities of a distance map from a given object boundary [35, 34]. In the context of thin objects like edges, vessels, etc, we consider a center-line to be a smooth curve minimizing orthogonal projection errors for the points of the thin structure.

We study curvature of the center-line as a regularization criteria for its inference. In general, curvature is actively discussed in the context of thin structures. For example, it is well known that curvature of the object boundary has significant effect on the medial axis [17, 35]. In contrast, we are directly concerned with curvature of the center-line, not

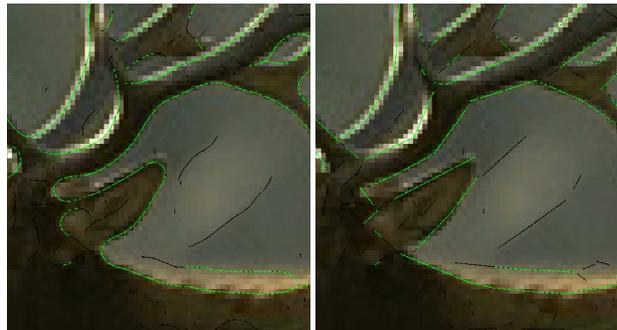


Figure 1. Edge detection. The result of our algorithm for squared (on the left) and absolute (on the right) curvature approximations. Green and black lines correspond to edges with high and medium confidence measure correspondingly. Note the strong bias to straight lines on the right: the energy prefers a small number of sharp corners rather than many smooth corners like on the left.

the curvature of the object boundary. Moreover, we do not assume that the boundary of a thin structure (e.g. vessel or road) is given. Detection variables are estimated simultaneously with the center-line. This paper proposes a general energy formulation and an optimization algorithm for detection and subpixel delineation of thin structures based on curvature regularization.

Curvature is a natural regularizer for thin structures and it has been widely explored in the past. In the context of image segmentation with second-order smoothness it was studied by [31, 37, 32, 5, 14, 28, 25]. It is also a popular second-order prior in stereo or multi-view-reconstruction [20, 27, 40]. Curvature has been used inside connectivity measures for analysis of diffusion MRI [24]. Curvature is also widely used for *inpainting* [3, 7] and edge completion [13, 39, 2]. For example, *stochastic completion field* technique in [39, 24] estimates probability that a completed/extrapolated curve passes any given point assuming it is a random walk with bias to straight paths. Note that common edge completion methods use existing edge detectors as an input for the algorithm.

In contrast to these prior works, this paper proposes a

general low-level regularization framework for detecting thin structures with accurate estimation of location and orientation. In contrast to [39, 13, 24] we explicitly minimize the integral of curvature along the estimated thin structure. Unlike [12] we do not use curvature for grouping pre-detected thin structures, we use curvature as a regularizer during the detection stage.

**Related work:** Our regularization framework is based on the curvature estimation formula proposed by Olsson et al. [26, 27] in the context of surface fitting to point clouds for multi-view reconstruction, see Fig.2(a). One assumption in [26, 27] is that the data points are noisy readings of the surface. While the method allows outliers, their formulation is focused on estimation of local surface patches. Our work can be seen as a generalization to detection problems where majority of the data points, e.g. image pixels in Fig.2(c), are not within a thin structure. In addition to local tangents, our method estimates probability that the point is a part of the thin structure. Section 2 discusses in details this and other significant differences from the formulation in [26, 27].

Assuming  $p_i$  and  $p_j$  are neighboring points on a thin structure, e.g. a curve, Olsson et al. [26] evaluate local curvature as follows. Let  $l_i$  and  $l_j$  be the tangents to the curve at points  $p_i$  and  $p_j$ . Then the authors propose the following approximation for the absolute curvature

$$|\kappa(l_i, l_j)| = \frac{\|l_i - p_j\| + \|l_j - p_i\|}{\|p_i - p_j\|}$$

and for the squared curvature

$$\kappa^2(l_i, l_j) = \frac{\|l_i - p_j\|^2 + \|l_j - p_i\|^2}{\|p_i - p_j\|^2}$$

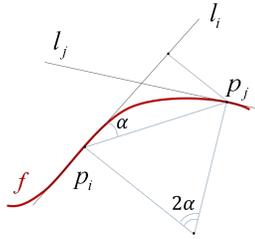
where  $\|l_i - p_j\|$  is the distance between point  $p_j$  and line  $l_i$ .

Assume that the curve  $\vec{r} = f(\tau)$  is parameterized by arc-length  $\tau$  such that  $\tau_1 \leq \tau \leq \tau_M$ . If  $(\tau_1, \tau_2, \dots, \tau_M)$  is an increasing parameter sequence then the curvature of  $f$  can be approximated by

$$\int |\kappa|^\alpha d\tau \approx \sum_{(i,j) \in N} |\kappa(l_i, l_j)|^\alpha$$

where  $N = \{(i, i+1) \mid i = 1, 2, \dots, M-1\}$  is a neighborhood system for curve points  $p_i = f(\tau_i)$  and  $l_i = \dot{f}(\tau_i)$  are their tangent lines.

Olsson et al. [26] use regularization for fitting a surface (or curve) to a cloud of points in 3D (or 2D) space. Every observed point  $\tilde{p}_i$  is treated as a noisy measurement of some unknown point  $p_i$  that is the closest point on the estimated surface, see Fig.2(a). Each  $\tilde{p}_i$  is associated with unknown



local surface patch  $l_i$  that is a tangent plane for the surface at  $p_i$ . The proposed surface fitting energy combines curvature-based regularization with the first order data fidelity term

$$E(L) = \sum_{(i,j) \in N} |\kappa(l_i, l_j)|^\alpha w_{ij} + \sum_i \frac{1}{\sigma^2} \|l_i - \tilde{p}_i\|^2 \quad (1)$$

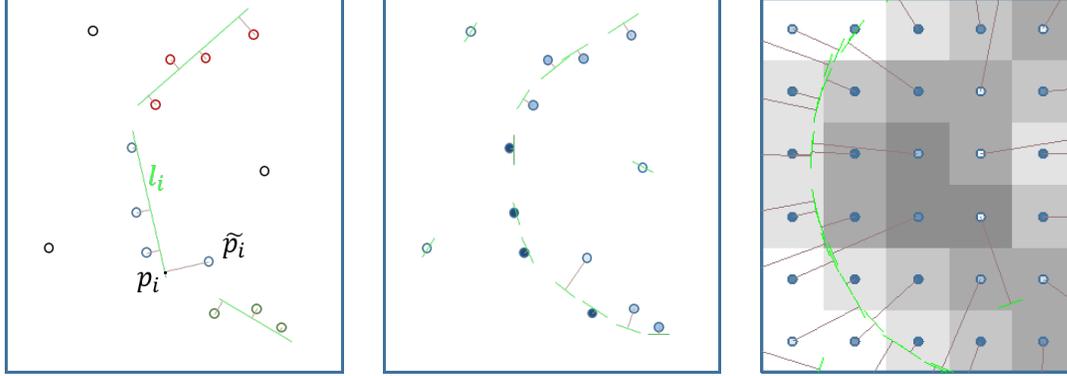
where  $L = \{l_i\}$  is the set of tangents,  $N$  is a neighborhood system,  $\sigma$  is non-negative constant,  $w_{ij}$  is a positive constant such that  $\sum_{j \in N_i} w_{ij} = 1$ . To minimize (1), the algorithm in [26] iteratively optimizes the assignment variables for a limited number of tangent proposals, and then re-estimates tangent plane parameters, see Fig.2(a).

In contrast to [26], our method estimates thin structures in the image grid where, *a priori*, it is unknown which pixels belong to the thin structure, see Fig.2(c). We introduce set  $X = \{x_i\}$  of indicator variables  $x_i \in \{0, 1\}$  where  $x_i = 1$  iff pixel  $\tilde{p}_i$  belongs to the thin structure. Our basic energy (2) and its extensions combine unary detection potentials with curvature regularization. Due to the regularity of our grid neighborhood, we use constant weights  $w_{ij}$ , which are omitted from now on. We propose a different optimization technique estimating a posteriori distribution of  $x_i$  and separate tangents  $l_i$  at each point. As illustrated in Fig.2(b), our framework is also applicable to energy (1) and multi-view reconstruction problem as in [26, 27].

Parent&Zucker [29] formulate a closely related *trace inference* problem for detecting curves in 2D grid. Similarly to us, they estimate indicator variables  $x_i$  and tangents  $l_i$ . However, they estimate  $x_i$  and  $l_i$  by enforcing a *co-circularity* constraint assuming given local *curvature information*, which they estimate in advance. In contrast, we simultaneously estimate  $x_i$  and  $l_i$  by optimizing objective (2) that directly regularizes curvature of the underlying thin structure. Moreover, [29] quantizes curvature information and tangents while our model uses real valued curvature and tangents. The extension of [29] to 3D is not trivial.

Similarly to [26, 29] we estimate tangents only at a *finite* set of points. Additional regularization is required if continuous center-line between these points is needed [16].

**Contributions:** It is known that curvature of an object boundary is an important shape descriptor [33] with a significant effect on *medial axis* [17, 35], which is not robust even to minor perturbations of the boundary. In the context of thin objects (e.g. edges, vessels) we study a concept of a center-line (a smooth 1D curve minimizing the sum of projection errors), which is different from *medial axis*. We regularize the curvature of the center-line. Unlike many standard methods for center-lines, we do not assume that the shape of the object is given and propose a general low-level vision framework for thin structure detection combined with sub-pixel localization and real-valued orientation of its center-line. Therefore, we propose an approach



(a) Olsson's model [26] (b) Our model for cloud of points (c) Our model for grid points

Figure 2. Comparison with [26]. An empty circle in (b) and (c) denotes low confidence and a dark blue circle means high confidence.

that takes into account all possible configurations of the indicator variables while estimating the tangents. This significantly improves stability with respect to local minima. Our optimization method uses variational inference and trust region frameworks adapted to absolute and quadratic curvature regularization.

Our proof-of-the-concept experiments demonstrate encouraging results in the context of edge and vessel detection in 2D and 3D images. In particular, we obtain promising results for estimating highly detailed vessels structure on high-resolution microscopy CT volumes. We also show examples of sub-pixel edge detection regularizing curvature. While there are no databases for comparing edge detectors with real-valued location and orientation estimation, we obtained competitive results on a pixel-level edge detection benchmark [11]. Our general early vision methodology can be integrated into higher semantic level boundary detection techniques, e.g. [22], but this is outside the scope of this work. Our current sequential implementation is not tuned to optimize performance. Its running time for edges in 2D image of Fig.1 is 20 seconds and for vessels in 3D volume of Fig.10 is one day. However, our method is highly-parallelizable on GPU and fast real-time performance on 2D images can be achieved.

In Section 2 we describe the proposed model and discuss a simple block-coordinate descent optimization algorithm and its drawbacks. In Section 3 we propose a new optimization method for our energy based on variational inference framework. In Section 4 we describe the details of the proposed method and discuss the difference between squared and absolute curvatures (Subsection 4.1). We describe several applications of the proposed framework in Section 5 and conclude in Section 6.

## 2. Energy formulation

In the introduction we informally defined the center-line of a thin structure as a smooth curve minimizing orthogonal

projection errors. Here we present the energy formalizing this criterion. First we note that in our model the curve is not defined explicitly but through points  $p_i$  it passes and tangent lines  $l_i$  at these points. The energy is given by

$$E(L, X) = \sum_{(i,j) \in N} \kappa^2(l_i, l_j) x_i x_j + \sum_i \frac{1}{\sigma^2} \|l_i - \tilde{p}_i\|_+^2 x_i + \sum_i \lambda_i x_i \quad (2)$$

where  $N$  is a neighborhood system,  $X = \{x_i\}$  is a set of indicator variables  $x_i \in \{0, 1\}$  where  $x_i = 1$  iff pixel  $\tilde{p}_i$  belongs to the thin structure,  $\lambda_i$  define unary potentials penalizing/rewarding presence of the structure at  $\tilde{p}_i$ . In contrast to (1), potentials  $\lambda_i$  define the data term while  $\frac{1}{\sigma^2} \|l_i - \tilde{p}_i\|_+^2$  is a soft constraint.

We explore two choices of the soft constraint  $\|l_i - \tilde{p}_i\|_+$ . The first one uses Euclidean distance. In that case it models normally distributed errors. Although it is appropriate for many applications, e.g. surface estimation in multi-view reconstruction [26, 27], the normal errors assumption is no longer valid for the image grid because the discretization errors are not Gaussian. In fact, using Euclidean distance may make the soft constraint term proportional to the length of the center-line, see illustration on the right.

Thus, we also propose a truncated form of Euclidean distance:

$$\|l_i - \tilde{p}_i\|_+ = \max(0, \|l_i - \tilde{p}_i\| - 1). \quad (3)$$

This does not penalize tangent lines  $l_i$  that are within one pixel from points  $\tilde{p}_i$ . Different applications may require a different choice of no-penalty threshold.

**Extensions.** We can extend the energy (2) by adding other terms that encourage various other useful properties. For

example, energy

$$E'(L, X) = E(L, X) - \gamma \sum_{(i,j) \in \mathcal{N}} x_i x_j \quad (4)$$

for  $\gamma > 0$  will reward well aligned tangents. The effect of this term is shown in Fig.6. This term is similar to edge “repulsion” in MRF-based segmentation. The overall pairwise potential  $(\kappa(l_i, l_j) - \gamma)x_i x_j$  encourages edge continuity.

Another extension is to use prior knowledge about the center-line direction  $g_i$  at pixel  $\tilde{p}_i$ :

$$E'(L, X) = E(L, X) + \beta \sum_i m(l_i, g_i)^2 x_i. \quad (5)$$

The term  $m(l_i, g_i)$  measures how well tangent line  $l_i$  is aligned with prior  $g_i$ :

$$m(l_i, g_i) = \|g_i\| \sin \angle(l_i, g_i).$$

The magnitude of  $g_i$  constitutes the confidence measure. For example, vectors  $g_i$  could be obtained from the image gradients or the eigenvectors in the vesseness measure [9].

### 2.1. Block-coordinate descent optimization

To motivate our optimization approach for energy (2) described in Section 3, first we describe a simpler optimization algorithm and discuss its drawbacks.

The most obvious way to optimize energy (2) is a block-coordinate descent. The optimization alternates two steps described in Alg.1. The auxiliary energy optimized on line 4 is a non-linear least square problem and can be optimized by a trust-region approach, see Section 4. The auxiliary function on line 5 is a non-submodular binary pairwise energy that can be optimized with TRWS[18].

---

#### Algorithm 1 Block-coordinate descent

---

- 1: Initialize  $L^0$  and  $X^0$
  - 2:  $k \leftarrow 0$
  - 3: **while** not converged **do**
  - 4:   Optimize  $L^{k+1} \leftarrow \arg \min_L E(L, X^k)$
  - 5:   Optimize  $X^{k+1} \leftarrow \arg \min_X E(L^{k+1}, X)$
  - 6:    $k \leftarrow k + 1$
  - 7: **end while**
- 

We found that Alg.1 is extremely sensitive to local minima, see Fig.3. The reason is that tangents  $l_i$  for points with indicator variables  $x_i^k = 0$  do not participate in optimization on line 4. To improve performance of block-coordinate descent, we tried heuristics to extrapolate tangents into such regions. We found that good heuristics should have the following properties:

1. Since integral of curvature is sensitive to small local errors (see the figure on the right), the extrapolating procedure should yield close tangents for

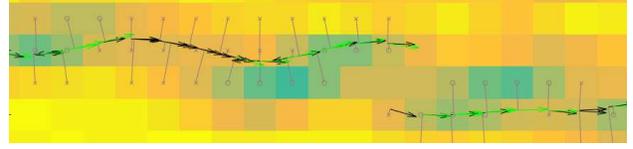
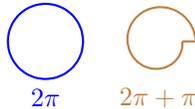


Figure 3. An example of local minima for Alg.1. The more “blue” is a pixel, the more likely it is to lie on an edge. Green arrows correspond to pixels that were initialized as edges. Black arrows correspond to the edges detected by Alg.1. This local minimum consists of two disconnected center-lines. The globally minimum solution smoothly connects the two pieces into a single center-line.

neighbors. Otherwise step 5 of the algorithm is ineffective. This issue could be partially solved by using energy (4). In this case it can be beneficial to connect two tangents even if there is some misalignment error.

2. The heuristic should envision that some currently disconnected curves may lie on the same center-line, see Fig.3.

The first property was easy to incorporate, while the second would require sophisticated edge continuation methods, e.g. a stochastic completion field [39, 24]. Instead we develop a new optimization procedure (Section 3) based on variational inference. The advantage of our new procedure is that it is closer to joint optimization of  $L$  and  $X$ .

### 3. Variational Inference

Ideally, we wish to jointly optimize (2) with respect to all variables. This is a mixed integer non-linear problem with an enormous number of variables. Thus, it is intractable. However, we can introduce elements of joint optimization based on stochastic *variational inference* framework. The proposed approach takes into account all possible configurations of indicator variables  $x_i$  while estimating tangents  $l_i$ . This significantly improves stability w.r.t. local minima.

Energy (2) corresponds to a Gibbs distribution:

$$P(I, X, L') = \frac{1}{Z} \exp(-E(L', X))$$

where  $Z$  is a normalization constant and the image is given by data fidelity terms  $I = \{\lambda_i\}$ . Here  $I$  are visible variables, indicator variables  $X$  and tangents  $L' = \{l'_i\}$  are hidden ones. We add a prime sign for tangent notation to distinguish values of random variables and parameters of the distribution. Our goal is to approximate the posterior distribution  $P(X, L'|I)$  of unobserved (hidden) indicators  $X$  and tangents  $L'$  given image  $I$ . The problem of approximating the posterior distribution has been extensively studied and is known as *variational inference* [4].

Variational inference is based on the decomposition

$$\ln P(I) = \mathcal{L}(q) + \text{KL}(q||p) \quad (6)$$

where  $\ln P(I)$  is the *evidence*,  $q(X, L')$  is a distribution over the hidden variables,  $p(X, L') = P(X, L'|I)$  is the

posterior distribution, and

$$\mathcal{L}(q) = \sum_X \int q(X, L') \ln \left( \frac{P(I, X, L')}{q(X, L')} \right) dL', \quad (7)$$

$$\text{KL}(q||p) = - \sum_X \int q(X, L') \ln \left( \frac{P(X, L'|I)}{q(X, L')} \right) dL'. \quad (8)$$

Since KL (Kullback–Leibler divergence) is always non-negative, the functional  $\mathcal{L}(q)$  is a lower bound for the evidence  $\ln P(I)$ . One of the nice properties of this decomposition is that the global maximum of lower bound  $\mathcal{L}$  coincides with the global minimum of  $\text{KL}(q||p)$  and optimal  $q^*(X, L') = \arg \max_q \mathcal{L}(q)$  is equal to the true posterior  $P(X, L'|I)$  [4].

Unfortunately (7) cannot be optimized exactly. To make optimization tractable, in variational inference framework one assumes that  $q$  belongs to a family of suitable distributions. In this work we will assume that  $q$  is a factorized distribution (*mean field theory* [30]):

$$q(X, L') = q(X)q(L'), \quad (9)$$

$$q(X) = \prod_i q_i(x_i) = \prod_i q_i^{x_i} (1 - q_i)^{1-x_i}, \quad (10)$$

$$q(L') = \prod_i \delta(l'_i - l_i) \quad (11)$$

where  $\delta(l'_i - l_i)$  is a deterministic (degenerate) distribution with parameter  $l_i$ . Under this assumption lower bound functional  $\mathcal{L}$  becomes a function of parameters  $q_i$  and  $l_i$ . We denote this function  $\mathcal{L}(Q, L)$  where  $Q = \{q_i\}$  and  $L = \{l_i\}$ .

The proposed algorithm is defined by Alg.2. It optimizes lower bound  $\mathcal{L}(Q, L)$  in block-coordinate fashion. The algorithm returns optimal tangents  $l_i^*$ , see Fig.5(b), and optimal probabilities  $q_i^*$ , see Fig.5(c).

---

**Algorithm 2** Block-Coordinate Descend for Variational Inference

---

- 1: Initialize  $L^0$  and  $Q^0$
  - 2:  $k \leftarrow 0$
  - 3: **while** not converged **do**
  - 4:   Optimize  $L^{k+1} \leftarrow \arg \max_L \mathcal{L}(Q^k, L)$
  - 5:   Optimize  $Q^{k+1} \leftarrow \arg \max_Q \mathcal{L}(Q, L^{k+1})$
  - 6:    $k \leftarrow k + 1$
  - 7: **end while**
  - 8: **return**  $L^k, Q^k$
- 

Now we consider optimization of  $\mathcal{L}$  over  $L$ . Taking into account (11), (2) and (7) we can derive

$$\begin{aligned} \arg \max_L \mathcal{L}(Q^k, L) &= \arg \min_L \sum_X q^k(X) E(X, L) = \\ &= \arg \min_L \sum_{(i,j) \in N} \psi_{ij} q_i^k q_j^k + \sum_i \psi_i q_i^k. \end{aligned} \quad (12)$$

where

$$\begin{aligned} \psi_{ij} &\equiv \kappa^2(l_i, l_j), \\ \psi_i &\equiv \frac{1}{\sigma^2} \|l_i - \tilde{p}_i\|_+^2 + \lambda_i. \end{aligned}$$

In case of (4) we redefine  $\psi_{ij} \equiv \kappa^2(l_i, l_j) - \gamma$ , and in case of (5) we redefine  $\psi_i \equiv \frac{1}{\sigma^2} \|l_i - \tilde{p}_i\|_+^2 + \lambda_i + \beta m(l_i, g_i)$ .

We see that optimization of  $\mathcal{L}(Q^k, L)$  with respect to  $L$  is a non-linear least square problem. For optimization details please refer to Section 4.

The optimization w.r.t.  $Q$  can be done by coordinate descent. The update equation is [4]

$$\begin{aligned} \ln q_i^*(x_i) &= E_{j \neq i} [\ln P(I, X|L)] + const = \\ &= -x_i \left( \sum_{j:(i,j) \in N} \psi_{ij} q_j + \psi_i \right) + const. \end{aligned} \quad (13)$$

The constant in expression (13) does not depend on  $x_i$  and thus can be determined from the normalization equation  $q_i(1) + q_i(0) = 1$ . We initialize  $q^0(x_i) = \exp(-x_i \psi_i) / (1 + \exp(-\psi_i))$  on line 1 of Alg.2. We iterate over all pixels update step (13) on line 5 until convergence, which is guaranteed by convexity of  $\mathcal{L}$  with respect to each  $q_i$  [4].

Note that if we further restrict  $q$  to be a degenerate distribution (meaning  $q(x_i) \in \{0, 1\}$ ) we will get the block-coordinate descend Alg.1.

The initialization of  $L^0$  is application dependent. In many cases some information about direction of a thin structure is available. Concrete initialization examples are described in Section 5.

**Alternative interpretations.** The goal of Alg.1 is to find  $\min_{L, X} E(L, X)$ , which is equivalent to

$$\max_L \max_X (-E(L, X)). \quad (14)$$

As shown in Section 2 optimization of (14) in a block-coordinate fashion requires optimization of tangents  $L$  with fixed indicator variables  $X$ . This necessitates extrapolation of tangents. Instead we propose to optimize  $L$  taking into account all possible configurations of  $X$ . That is we propose to replace maximum with smooth maximum:

$$\max_L \sum_X \exp(-E(L, X)).$$

Then we can write down a decomposition similar to (6), which provides a lower bound yielding the same optimization procedure.

The proposed procedure is closely related to the EM algorithm[8] where we treat tangents  $L$  as the parameters of the distribution. However, in this case the normalization constant of the distribution depends on  $L$  and optimization problem is intractable. One possible way to fix this issue is to use a *pseudo likelihood* [21].

## 4. Trust region for tangent estimation

Optimization of the auxiliary functions on line 4 of Algorithms 1 and 2 as well as energy (1) is a non-linear least square problem. In [26, 27] energy (1) is optimized using discrete multi-label approach in the context of surface approximation. In our work we adopt the inexact Levenberg-Marquardt method in [41], which is a trust region second order continuous iterative optimization method.

Each iteration consists of several steps. First, the method linearizes:

$$\begin{aligned} \mathcal{L}(q^k, L + \delta L) &\approx \mathcal{L}(\delta L) \equiv \\ &\equiv \sum_{(i,j) \in N} \left( |\kappa(l_i, l_j)| + \frac{\partial \kappa}{\partial l_i} \delta l_i + \frac{\partial \kappa}{\partial l_j} \delta l_j \right)^2 q_i^k q_j^k + \\ &+ \sum_i \frac{1}{\sigma^2} \left( \|l_i - \tilde{p}_i\|_+ + \frac{\partial d}{\partial l_i} \delta l_i \right)^2 q_i^k \end{aligned}$$

where for compact notation we define  $\kappa \equiv |\kappa(l_i, l_j)|$  and  $d \equiv \|l_i - \tilde{p}_i\|_+$ . We use [1] for automatic calculation of derivatives.

Second, the algorithm solves the minimization problem

$$\delta L^* = \arg \min_{\delta L} \mathcal{L}(\delta L) + \lambda \|\delta L\|^2$$

where  $\lambda$  is a positive damping factor, which determines the trust region. The method uses an inexact iterative algorithm for this task.

The last stage of iteration is to compare the predicted energy change  $\mathcal{L}(\delta L^*) - \mathcal{L}(\vec{0})$  with the actual energy change  $\mathcal{L}(q^k, L + \delta L^*) - \mathcal{L}(\vec{0})$ . Depending on the result of comparison the method updates variables  $L$  and damping factor  $\lambda$ . For more details please refer to [41].

The most computationally expensive part of Alg.2 is trust region optimization described in this subsection. From the technical point of view it consists of derivatives computation and basic linear algebra operations. Fortunately, these operations could be easily parallelized on GPU. We leave the GPU implementation for a future work.

### 4.1. Quadratic vs Absolute Curvature

Previous sections assume squared curvature, but everything can be adapted to the absolute curvature too. We only need to discuss how to optimize (12) for the absolute curvature. We use the following approximation:

$$\frac{\|l_i - p_j\|}{\|p_i - p_j\|} \approx \frac{\|l_i - p_j\|^2}{\|p_i - p_j\|^2} \cdot w_{ij} \quad (15)$$

where

$$w_{ij} = \frac{\|p_i - p_j\| + \epsilon}{\|l_i - p_j\| + \epsilon}$$

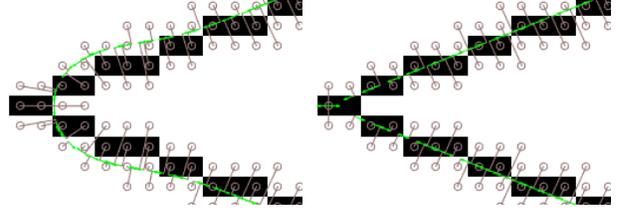
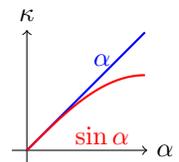


Figure 4. The difference between squared (left) and absolute (right) curvature approximations on an artificial example. Note the ballooning bias of squared curvature.

and  $\epsilon$  is some non-negative constant. If  $\epsilon = 0$  we have an approximation of the absolute curvature, if  $\epsilon \rightarrow \infty$  we have an approximation of the squared curvature.

The trust region approach (see Section 4) works with approximations of functions. It does not require any particular approximation like in the Levenberg-Marquardt method [19, 41]. Thus we can approximate the absolute curvature by treating  $w_{ij}$  as constants in (15) and linearizing  $\kappa(l_i, l_j)$  analogously to the squared curvature case.

The approximation of curvature given by [26] is derived under the assumption that the angles between neighbor tangents are small. Under this assumption the sine of an angle is approximately equal to the angle. And the approximation essentially computes the sines of the angles rather than the angles themselves. As a result it significantly underestimates the curvature of sharp corners.



For example, let us consider the integral of absolute curvature over a circle and a square. The integral of the approximation is  $2\pi$  and 4 correspondingly, while the integral of the true absolute curvature is  $2\pi$  in both cases. So the energy using this approximation of absolute curvature tends to distribute curvature into a small number of sharp corners showing strong bias to straight lines. Although approximation of squared curvature also underestimates curvature of sharp corners, it does not have a strong bias to straight lines. See figures 1 and 4 for comparison of the approximations.

## 5. Applications

### 5.1. Contrast edges

Here we consider an application of our method to edge detection and real-valued edge localization.

Sobel gradient operator [36] returns the gradient magnitude and direction for every image pixel. The high gradient magnitude is an evidence of a contrast edge. The direction of the gradient is a probable direction of the edge. We use the output of the gradient operator to define data fidelity terms of energy (4). For every pixel  $\tilde{p}_i$  let  $g_i$  be the gradient vector returned by the operator. We normalize vectors  $g_i$  by the sample variance of their magnitudes over the whole im-

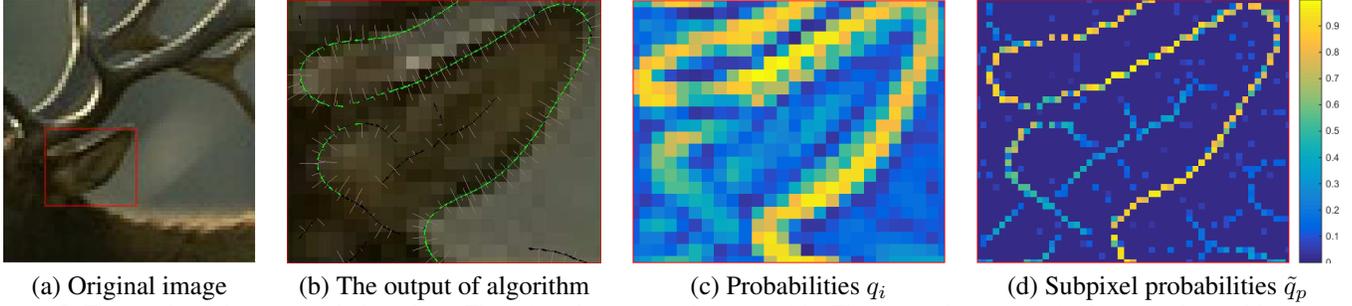


Figure 5. The result of the proposed algorithm. The original image is shown on (a). The zoomed in region is shown with a red box. Estimated tangents are shown in (b). Green color denotes tangents corresponding to pixels  $\tilde{p}_i$  such that  $q_i \geq \frac{1}{2}$ , and tangents corresponding to pixels with  $q_i \geq \frac{1}{4}$  are shown in black. (c) shows probabilities  $q_i$ . (d) shows the probabilities at doubled resolution produced by projecting points to their tangents:  $\tilde{q}_p = q_i$ .

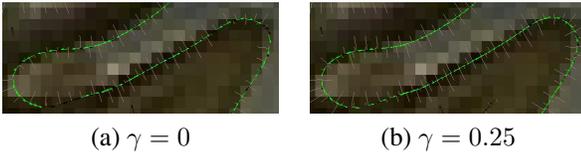


Figure 6. The effect of  $\gamma$  in energy (4). Tangents  $l_i$  whose  $q_i \geq \frac{1}{2}$  are shown in green, tangents such that  $\frac{1}{4} < q_i < \frac{1}{2}$  are shown in black. Increasing  $\gamma$  results in increasing probabilities  $q_i$  of well aligned tangents.

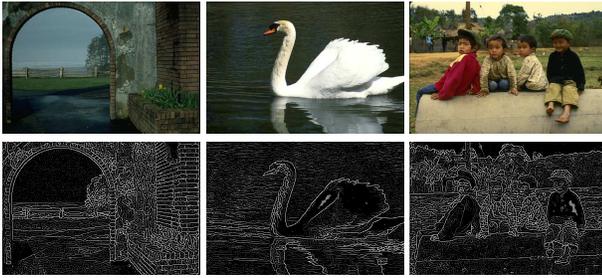


Figure 7. Examples of the output. The first row shows original images from CFGD database [11]. The second row shows edge masks at the original resolution produced by our algorithm.

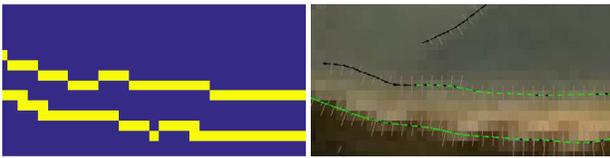


Figure 8. Comparison with Canny edge detector [6]. Note that Canny only produces the labeling of the pixels.

age. We define likelihood  $\lambda_i$  using hand picked linear transformation of the gradient magnitude:  $\lambda_i = 1.8 - 1.4 \cdot \|g_i\|$ . These parameters were optimized on a single picture shown in Fig.5(a). The initial tangents (line 1 of Alg.2)  $l_i$  are collinear with gradients  $g_i$  and pass through pixels  $\tilde{p}_i$ .

The results in figures 1, 5-9 was obtained by optimizing energy (4) using (3) as a soft constraint, with parameters  $\sigma = 1$ ,  $\gamma = 0.25$  and 8-connected neighborhood system  $N$ .

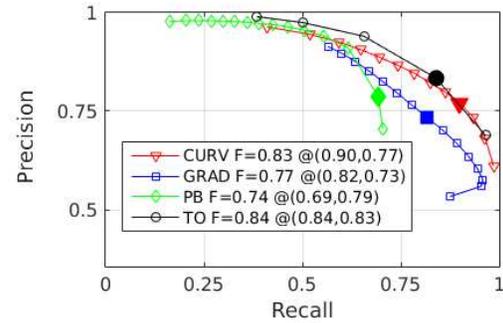


Figure 9. Comparison of our method (CURV) with the baseline gradients (GRAD), Pb [23] and the third order filter (TO) [38] on the database of [11]. Evaluation of Pb & TO is given by [11].

According to our model pixel  $\tilde{p}_i$  is a noisy measurement of point  $p$  on a contrast edge. Denoised point  $p_i$  is the projection of  $\tilde{p}_i$  onto  $l_i$ . To generate an edge mask (possibly at higher resolution) we can quantize  $p_i$  and use  $q_i$  as values at quantized  $p_i$ . If during this process we have a conflict such that several points are quantized into same pixel we choose the one with maximum probability. Fig.5(d) shows an edge mask whose resolution was doubled. Fig.7 shows examples of the edge mask at the original resolution.

We also compared our results with a few edge detection algorithms whose result is an edge mask, see Fig.9. This shows that our general method achieves F-measure of 0.83, which is very close to F-measure of 0.84, given by the best evaluated algorithm in [38]. Please note that [38] was designed specifically for edge detection in images, while our approach is a generic method for thin structure delineation.

## 5.2. Vessels in 3D

Vessel center-line localization in 3D medical volumes is an important task for medical diagnostics and pre-clinical drug trials.

For the experiments in this section we used a *microscopic computer tomography* [10, 15] scan of the mouse's heart. The scan is a 3D volume of size 585x525x892. For

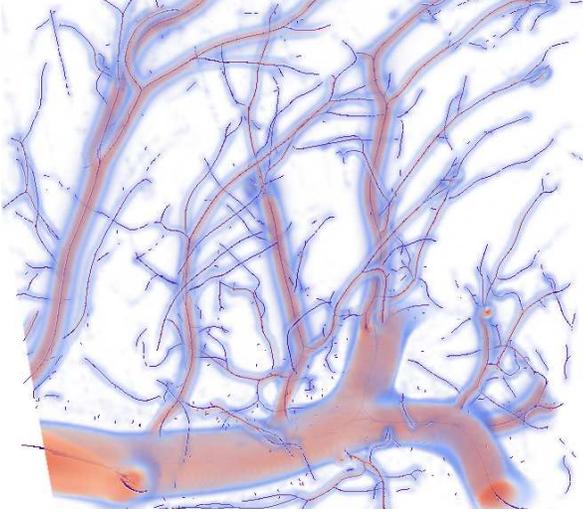


Figure 10. Example output of vessel center-line detection in 3D. Only tangents  $l_i$  with probabilities  $q_i \geq \frac{1}{2}$  are shown (in purple).

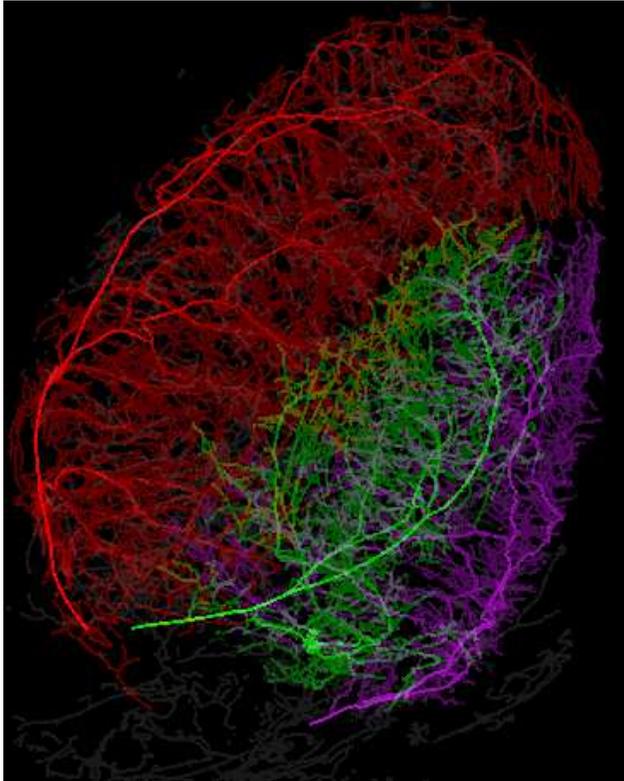


Figure 11. Center-line fitting for mouse heart. Three main branches are show in color. Other tangents are shown in dark gray.

the both experiments the volume was preprocessed with a popular vessel detection filter of [9]. For every voxel  $\tilde{p}_i$  the filter returns *vesselness measure*  $v_i$  such that higher values of  $v_i$  indicate higher likelihood of vessel presence at  $\tilde{p}_i$ . The filter also estimates direction  $g_i$  and scale  $\sigma_i$  of a vessel.

For this application we use extension (5) of energy (2).

Coefficient  $\frac{1}{\sigma}$  in front of the soft constraint in the energy determines how far tangents  $l_i$  can move from voxels  $\tilde{p}$ . Since this data has high variability in vessel thickness, we cannot use the same  $\sigma$  for every voxel. We substitute  $\sigma_i$  produced by the vesselness filter for  $\sigma$  in energy (5):

$$E(L, X) = \sum_{(i,j) \in N} \kappa^2(l_i, l_j) x_i x_j + \sum_i \left( \frac{1}{k^2 \sigma_i^2} \|l_i - \tilde{p}_i\|^2 + \beta m(g_i, l_i) + \lambda_i \right) x_i$$

where  $k$  is a positive constant and  $\lambda_i$  is obtained from vesselness measure  $v_i$  by the same linear transformation that we use in Section 5.1. We set  $\beta = 0.5$  and  $k = 20$  and use 26-connected neighborhood system  $N$ .

For the first experiment we cropped the volume forming a subvolume of size  $81 \times 187 \times 173$ . We also removed 85% of voxels with the lowest values of  $v_i$ . That yields about  $3 \cdot 10^6$  variables to be optimized. Fig.10 shows the result.

The goal of the second experiment is to extract a few trees describing the cardiovascular system of the whole heart. To decrease the running time we perform Canny's [6] hysteresis thresholding to detect one-dimensional ridges in the volume. We substitute vesselness measure for intensity gradients in Canny's procedure. Then we set  $q_i = 1$  for voxels detected as ridges and  $q_i = 0$  for other voxels. This yields approximately the same number of optimization variables. Then we optimize tangents by the algorithm described in Section 4. Then the estimated center-line points are grouped based on the tangent and proximity information into a graph and a minimum spanning tree algorithm extracts the trees. The result is shown in Fig.11.

## 6. Conclusion

We present a novel general early-vision framework for *simultaneous* detection and delineation of thin structures with sub-pixel localization and real-valued orientation estimation. The proposed energy combines likelihoods, indicator (detection) variables and squared or absolute curvature regularization. We present an algorithm that optimizes localization and orientation variables considering all possible configuration of indicator variables. We discuss the properties of the proposed energy and demonstrate a wide applicability of the framework on 2D and 3D examples.

In the future, we plan to explore better curvature approximations, extend our framework to image segmentation with curvature regularization, and improve the running time by developing parallel GPU implementation.

## Acknowledgements

We thank Olga Veksler (University of Western Ontario) for insightful discussions.

## References

- [1] S. Agarwal, K. Mierle, and Others. Ceres solver. <http://ceres-solver.org>. 6
- [2] T. D. Alter and R. Basri. Extracting salient curves from images: An analysis of the saliency network. *IJCV*, 27(1):51–69, 1998. 1
- [3] L. Alvarez, P.-L. Lions, and J.-M. Morel. Image selective smoothing and edge detection by nonlinear diffusion. ii. *SIAM Journal on numerical analysis*, 29(3):845–866, 1992. 1
- [4] C. M. Bishop et al. *Pattern recognition and machine learning*, volume 4. springer New York, 2006. 4, 5
- [5] K. Bredies, T. Pock, and B. Wirth. Convex relaxation of a class of vertex penalizing functionals. *Journal of Mathematical Imaging and Vision*, 47(3):278–302, 2013. 1
- [6] J. Canny. A computational approach to edge detection. *PAMI*, (6):679–698, 1986. 1, 7, 8
- [7] T. F. Chan and J. Shen. Nontexture inpainting by curvature-driven diffusions. *Journal of Visual Communication and Image Representation*, 12(4):436–449, 2001. 1
- [8] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society*, pages 1–38, 1977. 5
- [9] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever. Multiscale vessel enhancement filtering. In *MICCAI98*, pages 130–137. Springer, 1998. 4, 8
- [10] P. Granton, S. Pollmann, N. Ford, M. Drangova, and D. Holdsworth. Implementation of dual-and triple-energy cone-beam micro-ct for postreconstruction material decomposition. *Medical physics*, 35(11):5030–5042, 2008. 7
- [11] Y. Guo and B. Kimia. On evaluating methods for recovering image curve fragments. In *CVPRW*. 3, 7
- [12] Y. Guo, N. Kumar, M. Narayanan, and B. Kimia. A multi-stage approach to curve extraction. In *ECCV*. 2014. 2
- [13] G. Guy and G. Medioni. Inferring global perceptual contours from local features. In *CVPR*, 1993. 1, 2
- [14] S. Heber, R. Ranftl, and T. Pock. Approximate envelope minimization for curvature regularity. In *ECCV*, 2012. 1
- [15] D. W. Holdsworth and M. M. Thornton. Micro-ct in small animal and specimen imaging. *Trends in Biotechnology*, 20(8):S34–S39, 2002. 7
- [16] G. Kamberov and G. Kamberova. Ill-posed problems in surface and surface shape recovery. In *CVPR*, 2000. 2
- [17] B. B. Kimia, A. R. Tannenbaum, and S. W. Zucker. Shapes, shocks, and deformations i: the components of two-dimensional shape and the reaction-diffusion space. *IJCV*, 15(3):189–224, 1995. 1, 2
- [18] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *PAMI*, 28(10):1568–1583, 2006. 4
- [19] K. Levenberg. A method for the solution of certain nonlinear problems in least squares. *Quarterly of Applied Mathematics* 2, pages 164–168, 1944. 6
- [20] G. Li and S. W. Zucker. Differential geometric inference in surface stereo. *PAMI*, 32(1):72–86, 2010. 1
- [21] S. Z. Li. *Markov random field modeling in image analysis*. Springer Science & Business Media, 2009. 5
- [22] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, volume 2, pages 416–423, 2001. 3
- [23] D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *PAMI*, 26(5):530–549, 2004. 7
- [24] P. MomayyezSiahkal and K. Siddiqi. 3d stochastic completion fields for mapping connectivity in diffusion mri. *PAMI*, 35(4):983–995, 2013. 1, 2, 4
- [25] C. Nieuwenhuis, E. Toeppe, L. Gorelick, O. Veksler, and Y. Boykov. Efficient squared curvature. In *CVPR*, Columbus, Ohio, 2014. 1
- [26] C. Olsson and Y. Boykov. Curvature-based regularization for surface approximation. In *CVPR*, pages 1576–1583. IEEE, 2012. 2, 3, 6
- [27] C. Olsson, J. Ulén, and Y. Boykov. In defense of 3d-label stereo. In *CVPR*, pages 1730–1737. IEEE, 2013. 1, 2, 3, 6
- [28] C. Olsson, J. Ulén, Y. Boykov, and V. Kolmogorov. Partial enumeration and curvature regularization. In *ICCV*, pages 2936–2943. IEEE, 2013. 1
- [29] P. Parent and S. W. Zucker. Trace inference, curvature consistency, and curve detection. *PAMI*, 11:823–839, 1989. 2
- [30] G. Parisi. *Statistical field theory*, volume 4. Addison-Wesley New York, 1988. 5
- [31] T. Schoenemann, F. Kahl, and D. Cremers. Curvature regularity for region-based image segmentation and inpainting: A linear programming relaxation. In *ICCV*, Kyoto, 2009. 1
- [32] T. Schoenemann, F. Kahl, S. Masnou, and D. Cremers. A linear framework for region-based image segmentation and inpainting involving curvature penalization. *IJCV*, 2012. 1
- [33] T. B. Sebastian, P. N. Klein, and B. B. Kimia. Recognition of shapes by editing their shock graphs. *PAMI*, 2004. 2
- [34] K. Siddiqi, S. Bouix, A. Tannenbaum, and S. W. Zucker. Hamilton-jacobi skeletons. *IJCV*, 48(3):215–231, 2002. 1
- [35] K. Siddiqi and S. Pizer. *Medial representations: mathematics, algorithms and applications*, volume 37. Springer Science & Business Media, 2008. 1, 2
- [36] I. Sobel and G. Feldman. A 3x3 isotropic gradient operator for image processing. 1968. 6
- [37] P. Strandmark and F. Kahl. Curvature regularization for curves and surfaces in a global optimization framework. In *EMMCVPR*, pages 205–218. Springer, 2011. 1
- [38] A. Tamrakar and B. B. Kimia. No grouping left behind: From edges to curve fragments. In *ICCV*. IEEE, 2007. 7
- [39] L. R. Williams and D. W. Jacobs. Stochastic completion fields: A neural model of illusory contour shape and salience. *Neural Computation*, 9(4):837–858, 1997. 1, 2, 4
- [40] O. Woodford, P. Torr, I. Reid, and A. Fitzgibbon. Global stereo reconstruction under second-order smoothness priors. *PAMI*, 31(12):2115–2128, 2009. 1
- [41] S. Wright and J. N. Holt. An inexact levenberg-marquardt method for large sparse nonlinear least squares. *The Journal of the Australian Mathematical Society. Series B. Applied Mathematics*, 26(04):387–403, 1985. 6