# Curvature-Based Regularization for Surface Approximation

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

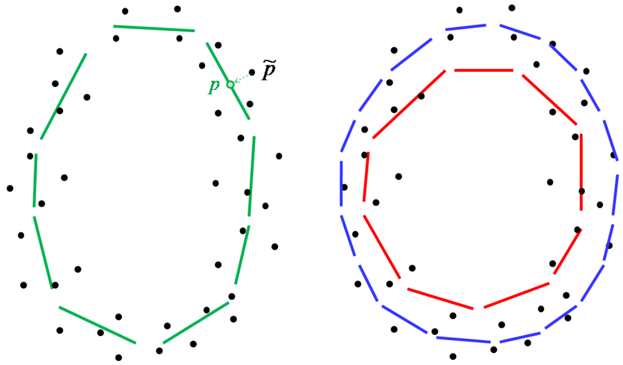
Yuri Boykov
Department of Computer Science
University of Western Ontario, Canada
yuri@csd.uwo.ca

## Abstract

*We propose an energy-based framework for approximating surfaces from a cloud of point measurements corrupted by noise and outliers. Our energy assigns a tangent plane to each (noisy) data point by minimizing the squared distances to the points and the irregularity of the surface implicitly defined by the tangent planes. In order to avoid the well-known "shrinking" bias associated with first-order surface regularization, we choose a robust smoothing term that approximates curvature of the underlying surface. In contrast to a number of recent publications estimating curvature using discrete (e.g. binary) labellings with triple-cliques we use higher-dimensional labels that allows modeling curvature with only pair-wise interactions. Hence, many standard optimization algorithms (e.g. message passing, graph cut, etc) can minimize the proposed curvature-based regularization functional. The accuracy of our approach for representing curvature is demonstrated by theoretical and empirical results on synthetic and real data sets from multi-view reconstruction and stereo.* [1]

## 1. Introduction

We present an energy minimization framework for *tangential approximation of surfaces* (TAS). For each noisy measurement (like $\tilde{p}$ in Fig.1a) our method estimates a tangent plane and a "true" point position on the underlying smooth surface (like $p$). The utilized measurements are application specific and could , for example, be 3D-positions (from multiple view geometry or laser scans), 2D projections, or photoconsistency (from image data). Our energy minimizes geometric errors and encourages tangents to agree on some underlying piece-wise smooth surface by penalizing curvature. Despite recent interest in computer vision community [15, 9, 16, 21] efficient optimization of functionals accurately approximating curvature is still a challenge. We propose a simple curvature-optimization framework that assumes no topological constraints (e.g. surface closedness) and is robust to intersections, outliers, etc. Unlike many prior works, we focus on curvature functionals that have no shape bias. For example, in 2D we use the absolute value of the curvature motivated in Fig.1.



(a) minimizing $\int |\kappa| ds$ (b) using length or $\int |\kappa|^2 ds$

Figure 1. Different regularization criteria for *tangential approximation of a surface* (e.g. an ellipsoid) from the same noisy data points. Well-known Euler's formula implies that minimization of geometric errors combined with an integral of the absolute curvature yields no shape bias (a). It is easy to show that more common squared curvature biases shapes to larger circles (blue), while standard length-based regularization (red) has "shrinking" bias to smaller circles, see (b). This figure is an illustration only.

Our framework is based on a graphical model. In general, graphical models are widely used in vision for problems like dense stereo [21, 4], surface estimation [19], image segmentation [4], inpainting [15], etc. Perhaps the most common regularizer is *length* (or *area* in 3D) since this smoothness criteria corresponds to graphical models with simple pair-wise potentials, which admit very efficient global optimization algorithms like [4, 12]. Recently, the vision community has begun to actively explore graphical models using curvature-based regularization or similar second-order smoothness priors [15, 9, 16, 21]. Typically,

[1] This work has been funded by the Swedish Foundation for International Cooperation in Research and Higher Education (STINT, grant KU2003 4682), the Swedish Foundation for Strategic Research (SSF) through the programmes *Future Research Leaders* and *Wearable Visual Information Systems*, by the European Research Council (GlobalVision grant no. 209480), and by the Canadian Foundation for Innovation (CFI 10318).

evaluation of the second-order smoothness properties of a curve/surface requires an interaction between three or more points. This leads to hard-to-optimize graphical models with higher-order cliques instead of simple pairwise potentials. In [15] the curvature problem is solved by formulating a linear integer program, which is solved by relaxing to a linear program. While this approach has been demonstrated to often provide integer or near integer solutions, the size of the linear program currently limits the usefulness of this approach. Furthermore, it requires the curve/surface to be closed.

This paper takes an alternative approach to evaluating curvature. Instead of using graphical models with higher-order interactions, we extend our label space so that the labels (tangents) encode both position and orientation of a curve/surface. This makes it possible to evaluate curvature using only pairwise interactions. Thus, standard combinatorial optimization methods, e.g. TRW-S [12], can be readily applied. To demonstrate this general approach we primarily focus on regularization of point clouds. For each noisy 3D-point we try to estimate its "true" position and a tangent plane, such that neighboring points agree on an underlying piecewise smooth surface. We do not assume any particular topology, nor does it have to be closed or "orientable".

Piece-wise constant regularization was previously applied to geometric problems, e.g. [11, 7]. Their labels are global geometric primitives (lines, circles, homographies). For a general scene such an approach may require enumerating a very large number of primitives to account for all possible surfaces in a scene. In contrast, this paper adopts an approach where global non-parametric surfaces are formed by smoothly combining locally estimated primitives (e.g. tangent lines/planes). As shown by [18] in the context of non-rigid structure from motion, similar *local-to-global* approach may work even without regularization. In general, however, piece-wise smooth MRF priors [3] are necessary to build global surfaces from local primitives estimated from ambiguous data with noise and outliers.

In the graphics community surface estimation from point clouds is widely studied [10]. Popular approaches are least squares surfaces (MLS) [1], locally optimal projections (LOP) [14], and anisotropic point cloud diffusion [13]. However, none of these methods regularize curvature. Perhaps, the closest to our approach is [17] that uses active surface elements, a.k.a. *surfels*. In principle, curvature could be estimated from pairwise potentials between the surfels. However, this is difficult in practice since surfels correspond to a label space with six degrees of freedom (point position and a local coordinate frame). In contrast, our labels have only three d.o.f. (tangent plane).

Our main contribution is a simple graphical framework for tangential approximation of surfaces (TAS) that uses pairwise potentials to regularize curvature. No assumptions about surface topology are made. In practice, standard optimization algorithms (e.g. TRWs) find good solutions for our energies. We discuss curvature regularization measures that have no shape biases. The accuracy of our approach to curvature is shown both theoretically and empirically.

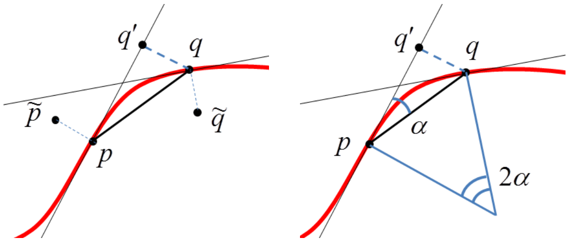


Figure 2. Left: pairwise interaction approximating curvature corresponds to quotient (12). Right: the quotient $\frac{|q-q'|}{|p-q|^2}$ yields half the curvature at $p$ under the assumption that $p$ and $q$ belong to a constant curvature segment.

## 2. Regularizing 2D-Point-Clouds

For simplicity, we first discuss our TAS framework in 2D. In this paper we primarily focus on estimating models from point measurements corrupted by noise. This is a very common problem in computer vision. In many instances the model itself can not be inferred from only point measurements and additional knowledge needs to be incorporated, often in the form of regularization. Consider a set of points that are noisy observations from some underlying curve. We will assume that the underlying curve is piecewise smooth and that the samples are sufficiently dense and cover the whole curve. The goal of this paper is to recover the "true" (noiseless) positions of the points in the point cloud. We do not recover the curve itself but rather an approximation of it, consisting of the tangents to the curve at each estimated point.

We assume that points $\tilde{p}$ and $\tilde{q}$ are noisy measurements of the points $p$ and $q$ on the underlying curve (see Figure 2). To $\tilde{p}$ and $\tilde{q}$ we assign (as labels) tangent lines $(n_p, d_p)$ and $(n_q, d_q)$ where $n_p$, $n_q$ are unit vectors representing the normals of the tangent lines and scalars $d_p$ and $d_q$ are distances from these lines to the origin. The estimated "true" points $p$ and $q$ will be the projections of $\tilde{p}$ and $\tilde{q}$ onto the assigned tangent lines. Depending on the particular application these projections can take different forms. For example, if the measurement errors are modeled as Gaussian noise in the direction of the normal (which is reasonable for curves with low curvature) then the estimated points $p$ and $q$ are orthogonal projections of observations $\tilde{p}$ and $\tilde{q}$ onto the corresponding tangent lines (see Fig.2 left)

$$p = \tilde{p} - (n_p^\top \tilde{p} + d_p) n_p \quad (1)$$

$$q = \tilde{p} - (n_q^\top \tilde{q} + d_q) n_q. \quad (2)$$

Given the estimations $p$ and $q$ and their tangents we would like to estimate how smooth the underlying curve is. If it is smooth and the points $p$ and $q$ are close then the two tangents at $p$ and $q$ should be similar. There are several possible choices for measuring the difference between the tangents. The most obvious is perhaps to measure the angle between the normals. Since angles are related to curvature, this may seem a good choice. However this measurement does not depend on the position of the estimated points. Even if the tangents are very far apart at the estimated points the angle between them might be small.

Bruckstein etal [5] use angles between consecutive line segments to measure curvature. While this gives an easy type of interaction it requires an explicit representation of the estimated polygonal curve. In addition it does not seem possible to generalize this measure to the 3-dimensional case without resorting to triple cliques. In contrast we are seeking an implicit local estimation using tangent lines/planes with pairwise interactions. To measure the difference between the two tangent planes we therefore compute the projections (see Fig.2 left)

$$p' = p - (n_q^\top p + d_q)n_q \qquad (3)$$

$$q' = q - (n_p^\top q + d_p)n_p, \qquad (4)$$

that is, the estimated points projected onto the neighboring tangent plane. If the underlying surface is smooth then

$$|p' - p| + |q' - q| \qquad (5)$$

should be small since the tangents are close to each other. Note that the projections (3) and (4) are not application dependent. For the smoothness estimate we will always use orthogonal projection. The reason for this will become apparent in the following section.

### 2.1. Relation to Curvature

The term (5) turns out to be closely related to curvature. To see this we assume that $(n_p, d_p)$ , $(n_q, d_q)$ are the true tangent planes at the points $p$ and $q$, and that

$$p = \alpha(t_1) \qquad (6)$$

$$q = \alpha(t_2), \qquad (7)$$

where $\alpha$ is a smooth curve. Furthermore, we will assume that this curve is parametrized by arc-length. (This is however no restriction since any curve with derivative $\dot{\alpha} \neq 0$ can be re-parametrized.) Since $n_p^\top p + d_p = 0$ we have

$$|q - q'| = |n_p^\top q + d_p| = |n_p^\top (q - p)| \qquad (8)$$

Using the Taylor expansion of $\alpha$ at $t_1$,

$$\alpha(t) = \alpha(t_1) + \dot{\alpha}(t_1)(t - t_1) + \frac{1}{2}\ddot{\alpha}(t_1)(t - t_1)^2 + O((t - t_1)^3). \qquad (9)$$

Together with (6)-(7), the term (8) can be written

$$|n_p^\top \left(\dot{\alpha}(t_1)(t_2 - t_1) + \frac{1}{2}\ddot{\alpha}(t_1)(t_2 - t_1)^2\right) + O((t - t_1)^3)|. \qquad (10)$$

Since $\dot{\alpha}(t_1)$ is the tangent at $p$ it is perpendicular to the normal $n_p$ so the first term vanishes. Furthermore, since $\alpha$ is parametrized by arc length, $\ddot{\alpha}$ is perpendicular to the tangent $\dot{\alpha}$. Therefore $n_p^\top \ddot{\alpha}(t_1)$ is only the length of $\ddot{\alpha}(t_1)$ (plus or minus, depending on the direction of $n_p$), which is the curvature. Hence if we divide $|q - q'|$ by $(t_2 - t_1)^2$ we get roughly half the curvature. However, as we estimate the underlying curve (or any parametrization of it) $t_1$ and $t_2$ are unknown. Again using the Taylor expansion we note that

$$|p - q|^2 = |\dot{\alpha}(t_1)|(t_2 - t_1)^2 + O((t_2 - t_1)^3) \qquad (11)$$

Since $\alpha$ is a unit speed curve $|\dot{\alpha}| = 1$, and therefore

$$\frac{|q - q'|}{|p - q|^2} = \frac{\frac{1}{2}|n_p^\top \ddot{\alpha}(t_1) + O(t_2 - t_1)|}{|1 + O(t_2 - t_1)|}. \qquad (12)$$

This expression will tend to half of the curvature when $t_2$ tends to $t_1$, and therefore approximates curvature well if $p$ and $q$ are close enough.

Figures 3-5 shows the computation of the quotient (12) for the curve $\sqrt{3}\cos(t)$. To approximate the integral of the curvature we use

$$\frac{1}{2}\int |\kappa| d\sigma \approx \sum_i \frac{|p_{i+1} - p'_{i+1}|}{|p_{i+1} - p_i|}, \qquad (13)$$

where $p_i$ are the sampled 2D points. Since the function has amplitude $\sqrt{3}$ the derivative is $\pm\sqrt{3}$ in its endpoints. This gives 60 degree angles to the x-axis at the endpoints, which is very close to what the approximations in Figures 3-5 give.

An interesting special case where the approximation turns out to be exact is when the points are lying on a constant curvature segment, that is, a segment that is part of a circle or line. For simplicity let us assume that $p$ and $q$ is on a circle with center at the origin, see Figure 2 right. If we parametrize the curve by arc length the angle $2\alpha$ is $t/r$. By the cosine theorem we get

$$|p - q|^2 = 2r^2(1 - \cos(\frac{t}{r})) \qquad (14)$$

Since the angle $\alpha$ is $t/2r$ we see from the second triangle that

$$|q - q'| = |p - q'|\sin(\frac{t}{2r}) \qquad (15)$$

Assuming $0 \leq \frac{t}{2r} \leq \pi$, this gives us the quotient

$$\frac{|q - q'|}{|p - q|^2} = \frac{\sin(\frac{t}{2r})}{\sqrt{2r^2(1 - \cos(\frac{t}{r}))}} = \frac{\sin(\frac{t}{2r})}{2r|\sin(\frac{t}{2r})|} = \frac{1}{2r}, \qquad (16)$$

which is half of the curvature of a circle with radius $r$.

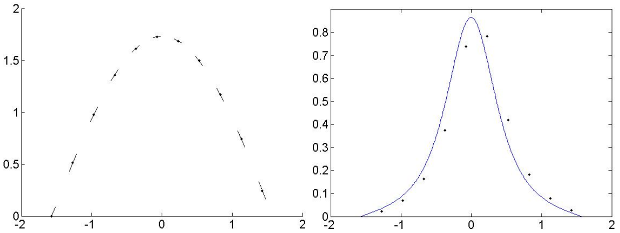


Figure 3. Computation of the curvature of $\sqrt{3}\cos(t)$, $-\pi/2 \leq t \leq \pi/2$. Left: The sample points and their tangent planes. Right: The computed curvature at the sample points, and the true curvature of the function. The value of the integral approximation is 59.6265 degrees (converted from radians).

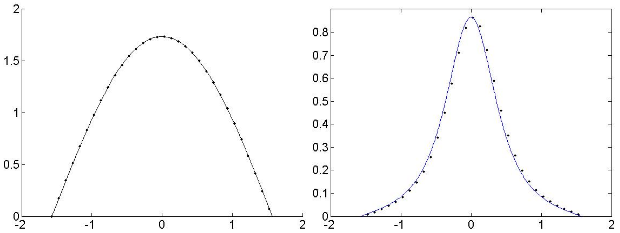


Figure 4. Same as above but with a denser sampling. The value of the integral approximation is 59.9600 degrees (converted from radians).

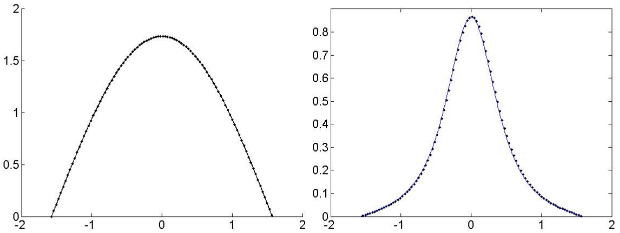


Figure 5. Same as above but with an even denser sampling. The value of the integral approximation is 59.9961 degrees (converted from radians).

## 2.2. Smoothness of a set of Points

In previous sections we showed that we can accurately measure the smoothness of a curve by evaluating (12) when the points and there normals are known. In practice the normals $n_p$ are not known but will be computed by optimizing the tangent lines. However, equation (10), shows that if the sampling rate is small enough then optimizing over $n_p$ will yield the correct solution. This can be seen by noting that if we minimize (10) with respect to $n_p$ then the dominant term $n_p^\top \dot{\alpha}(t_1)(t_2 - t_1)$ (assuming sufficiently high sampling rate) vanishes if $n_p$ is perpendicular to the tangent $\dot{\alpha}(t_1)$. Hence $n_p$ will approach the true normal when the sampling distance approaches zero. Therefore we take

$$\min_{\{(n_{p_1},\dots,n_{p_m});\ ||n_{p_i}||=1,\ \forall i\}} \sum_{i=1}^{m} \sum_{j\in\mathcal{N}(i)} \frac{|p_j - p_j'|}{|p_i - p_j|} \qquad (17)$$

as our measure of smoothness of a set of points. Here $\mathcal{N}(i)$ represents some predefined neighborhood of the point $i$. In equation (13) the neighborhood of $i$ is $i+1$, however for general point clouds the connectivity of the points is not known and therefore the neighborhood will in practice consist of nearby located points (see section 2.3). Note that since we are assuming that the points $p_i$ are fixed the above minimization problem is separable and each normal could be optimized by itself. However in general, we will also assume that the points are noisy in which case $p_i$ will also depend on its normal.

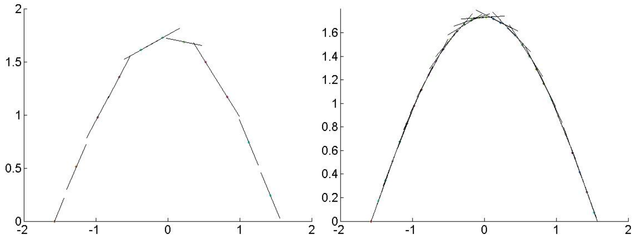


Figure 6. Estimated tangent planes for exactly measured data points at two different sampling rates. Left: The energy of the solution is 2.0221 (which corresponds to 115.8578 degrees). Right: The energy of the solution is 2.0864 (which corresponds to 119.5419 degrees)

Figure 6 shows the solutions obtained when optimizing (17) for two different sampling rates. Here we are using both forward and backward distances, and therefore (17) should evaluate to approximately 120 degrees (when converted from radians). Even at the crude sampling rate of Figure 6, the sum in (17) is close to the true value.

## 2.3. Energy

Next we consider the full problem of regularizing 2D-point clouds. We consider $\tilde{p}_i$ to be noisy measurements of the true points $p_i$. We will estimate the $p_i$ and the curve implicitly represented by its tangent lines $l_i = (n_{p_i}, d_{p_i})$ by minimizing the pairwise energy

$$\min \sum_i \sum_{j\in\mathcal{N}(i)} w_{ij} E_s(l_i, l_j) + \mu \sum_i E_d(l_i). \qquad (18)$$

Here the minimum is taken over the set of tangent lines.

The data term $E_d$ measures the quality of the estimation of $p_i$. As mentioned before this term is application dependent, and should be chosen depending on what type of noise we may expect. In the 2D case we will use

$$E_d(l_i) = (n_{p_i}\tilde{p}_i + d_{p_i})^2, \qquad (19)$$

which is the squared distance between the measured 2D point and its orthogonal projection on the tangent line. It is straight forward to use other data terms that would prefer certain directions and positions. For example in the 3D-case a data term based on photoconsistency could be used. However in this work we are mainly interested in 3D point clouds and we do not assume that any other knowledge than the point measurements them selves are available.

For the smoothness term $E_s(l_i, l_j)$ we will use one of the quotients

$$\frac{|p_j - p'_j|}{|p_i - p_j|}, \frac{|p_j - p'_j|^2}{|p_i - p_j|^3}, \frac{|p_j - p'_j|^2}{|p_i - p_j|^2} \qquad (20)$$

(Note that the estimated points $p_i$ $p_j$ depend on the tangent lines $l_i$ and $l_j$.) The first quotient corresponds to integrating absolute curvature, the second one integrating squared curvature. It is well known that integrating absolute curvature is the same as measuring the angle between the tangents at the start and endpoints of the curve (assuming the curve is part of a boundary of a convex region). Therefore this regularizer gives the same result regardless if the tangent space changes is located to a small region or spread out over the entire segment. In contrast the second one would encourage smoother transitions between tangent spaces. It does however have a growing bias. Integrating squared curvature over a circle with radius $r$ gives

$$\int \kappa^2 d\sigma = \int_0^{2\pi} \frac{1}{r^2} r d\theta = \frac{2\pi}{r}. \qquad (21)$$

Therefore we prefer to use the third quotient that also encourages smoother estimations but does not have the growing bias (being scale invariant). Since we want to handle curves which are only piecewise smooth we add a threshold $\tau$ to the smoothing term as well. This prevents the energy from overly penalizing corners and jumps where the curvature becomes very large.

Note that energy (19) does not assume that the order of the points are known. In the examples of sections 2.2 and 2.1 we only used neighboring points in the $x$-coordinate. In practice, when we have noise in both coordinates we cannot be certain of the order of the points. There are applications, such as for example stereo, where we can assume that the noise is dominant in some direction (the depth direction) where we can determine the point ordering reliably. In such cases it may be enough to use neighborhoods similar to that of section 2.2.

In the general case we assume noise in both point coordinates and an unknown topology of the surface. Therefore our neighborhood system is based on distances between the measured 3D points alone. For most of the implementations we use delaunay triangulations/tetraedizations where we have removed edges that are considered to long.

The number of neighbors is not fixed and therefore the number of terms in the inner sum of the smoothness term will vary. Since each term in the inner sum may be seen as a measurement of the curvature at $p_i$ the sum will penalize terms with many neighbors harder. Therefore we chose the weights $w_{ij}$ in (18) to be positive and sum to one over $j$. If we have any reason to believe that certain measurements are more accurate than others this may be encoded in the weights. In this section we only select them to be constant over $j$. This makes certain that the curvature at each point is counted once. Still there is a bias to selecting solutions with low curvature at areas with many points and high curvature with a small number of points. If this bias is unwanted it can be removed by multiplying with some measure of point density.

## 2.4. Optimization and Implementation

To optimize our energy we employ and alternating approach similar to that of [7, 11]. First candidate tangents are computed using random sampling and refinement using local optimization of the energy. The energy is then optimized with the current candidates using TRW-S. We use the implementation form [12] and the matlab wrapper from [21]. After the energy has been optimized we compute new candidates based on the current labeling and re-optimize.

## 2.5. Experiments

Next we present a few synthetic experiments that illustrate the properties of the different choices in (20). To generate the data of Figure (7) we sampled points from an ellipse and added Gaussian noise to the samples. We then used our algorithm to estimate the tangent planes. To the left in Figure 7 is the result of using the first term of (20). This term approximates integration of absolute curvature. The integral in (13) only depends on the total variation of the tangents and does not care weather this variation is located to a single point or spread out. This formulation often produces edgy looking solutions, especially when the data is sparse. Therefore this formulation is preferable if the underlying surface has kinks.

The two middle images show the result using the second quotient. Since this term approximates integration of squared curvature it prefers to spread the variation of the tangents thus giving a smoother result. On the other hand, it has a growing bias. Reducing the weight of the data term gives the result of the third image.

To the right is the result using the third option of (20). This formulation also prefers to spread out the variation but has no growing bias.

Figures 8 and 9 show the results of a more difficult problem comparing the three different smoothness terms of (20). Here we sampled 1300 2D points from 3 smooth curves and added Gaussian noise. In addition we added 100 outliers to the data. It can be seen that despite the noise levels, the presence of outliers and kinks and intersections of the curves, all of the formulations give reasonable results and a correct overall shape.

# 3. Regularizing 3D-Point-Clouds

In this section we consider the problem of regularizing 3D-point clouds. Our energy will be the same as for the

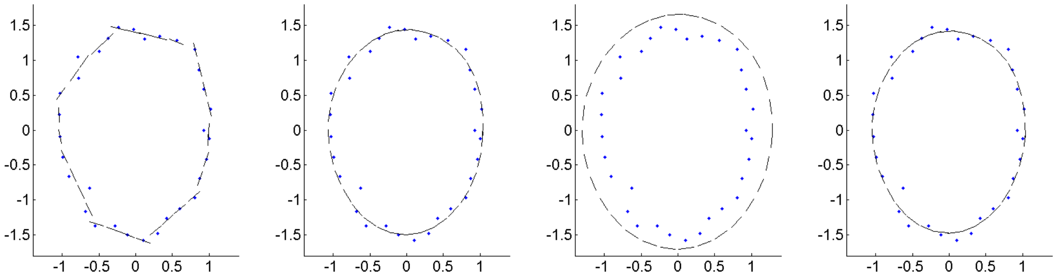


Figure 7. Ellipse experiments. From left to right: Using 1st term in (20), 2nd term, 2nd term with weaker data term, and 3rd term.

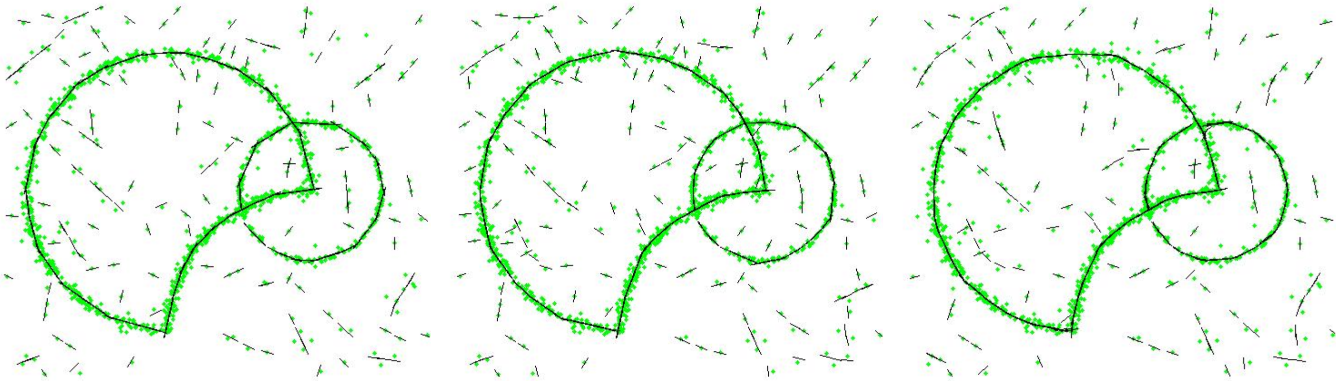


Figure 8. Input points (green) and estimated tangent lines (black) obtained using (from *left* to *right*) 1st, 2nd, and 3rd terms in (20).

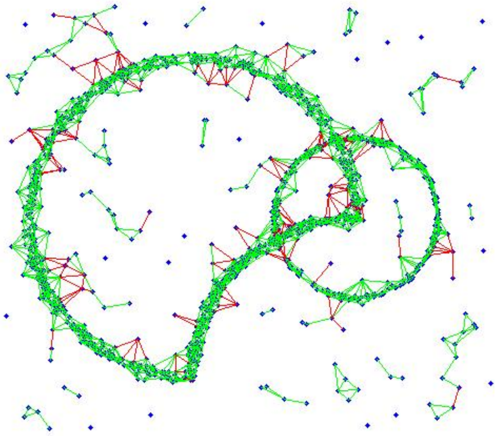


Figure 9. Neighborhood structure. Smoothness term for red edges equals threshold. (This result is similar for all variants in (20))

2D-case except that the tangent lines are replaced by tangent planes. It can be shown (in a similar way as for the 2D case) that the quotient (12) can be interpreted as normal curvature at the point $p$ in the direction towards $q$.

Popular curvature regularizers for surfaces are the Wilmore and bending energies [20, 8]. Using the Euler formula they can both be shown to integrate normal curvature over the surface [8]. However to represent these energies requires triple order cliques. In contras our energy measures the curvature of the surface by integrating along curves on the surface.

### 3.1. Unordered Point Clouds

In this section we apply our energy to 3D-point cloud regularization. Here we assume that the noise is Gaussian in the direction of the normal. Therefore the data term will be the same as that of equation (19). We apply it to the two real datasets depicted in figures 10 and 11. The first one is a set of 3D points at the surface of a castle (18270 points), and the second one is one of the laser scans of the well known Stanford bunny (40256 points) The first dataset is more noisy since it was created using a 3D reconstruction scheme, whereas the bunny was scanned in a laboratory setting. In both cases we used the third quotient of (20). Figure 10 and 11 shows the obtained results. The settings used where $\mu = 5e5$, $\tau = 0.56$ for the bunny, and and $\mu = 2500$, $\tau = 0.56$ for the castle. In both cases the solution took roughly 3 hours to compute. For the computations we used the code available from [12] and the matlab wrapper from [21]. Since our energy is not of any of the standard forms supported by [12], we used the general lookup table form (which is highly inefficient according [12]) to setup our potentials. Specialized software should therefore be able to speed up these computations considerably.

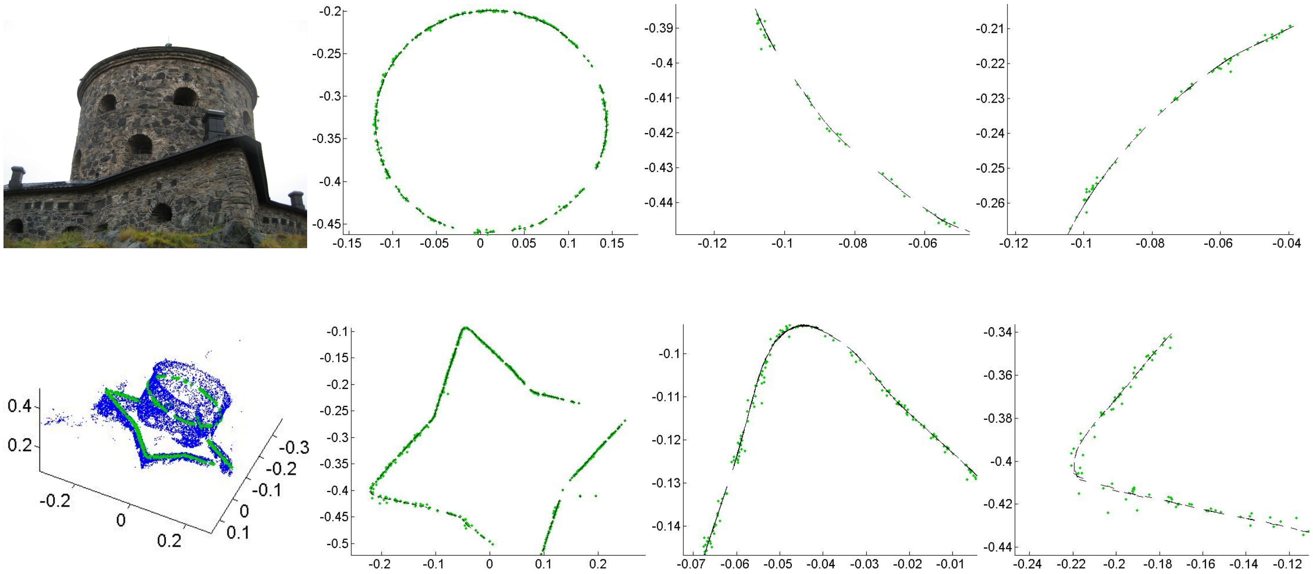


Figure 10. Visualization of the resulting tangent planes along two planar cuts. *1st column*: Image of the scene, Input 3D points (blue, 18270 in total) and points (green) close to the two cutting planes. *2nd column*: Points (green) projected onto the plane and line segments (black) obtained when intersecting the corresponding tangent planes with the cutting plane. *3rd and 4th columns*: Two closeups.

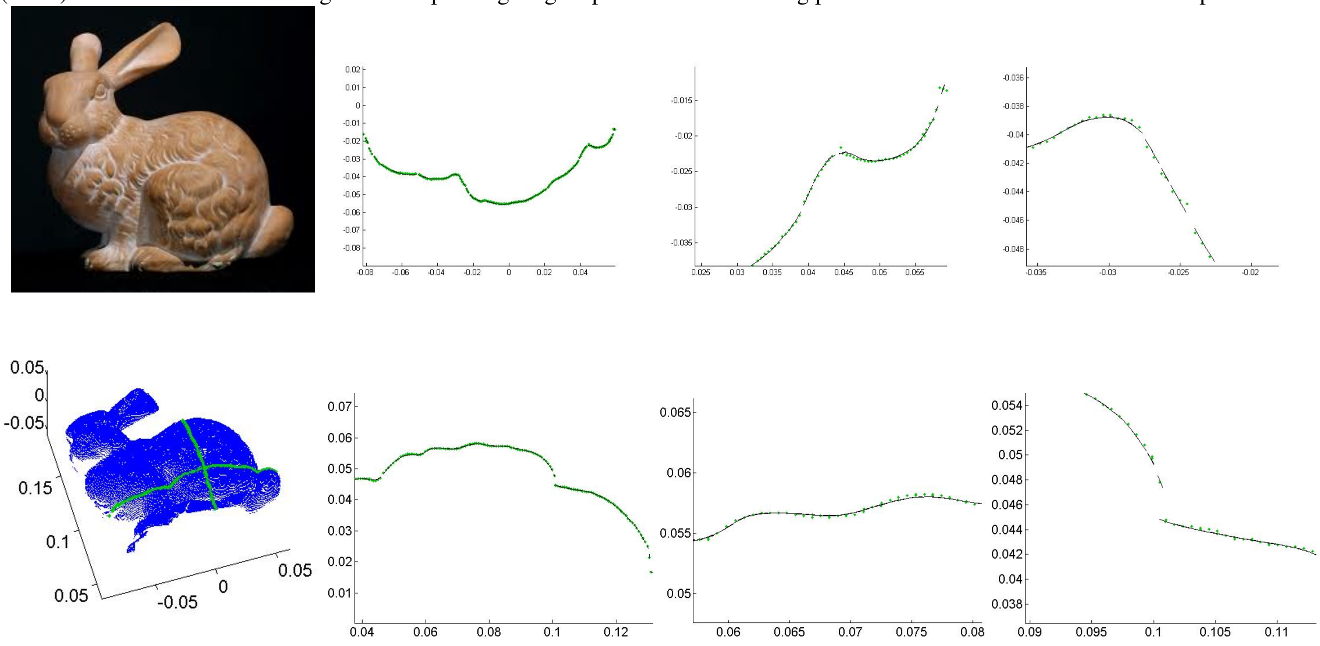


Figure 11. Visualization of the resulting tangent planes along two planar cuts. *1st column*: Image of the scene, Input 3D points (blue, 40256 in total) and points (green) close to the two cutting planes. *2nd column*: Points (green) projected onto the plane and line segments (black) obtained when intersecting the corresponding tangent planes with the cutting plane. *3rd and 4th columns*: Two closeups.

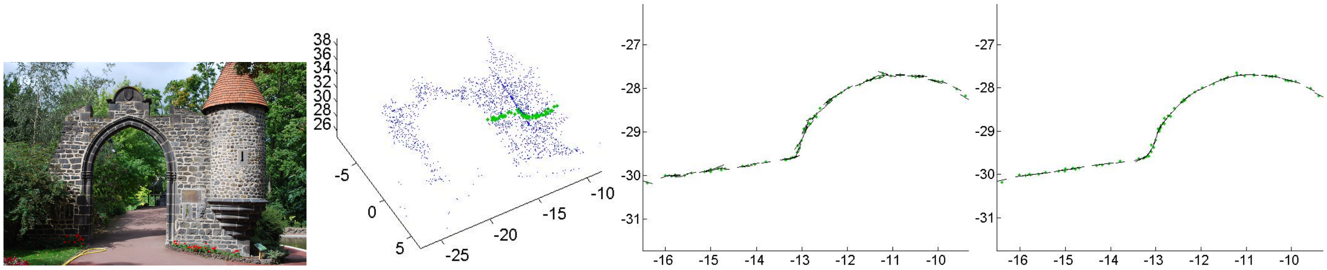


Figure 12. Visualization of the sparse stereo experiment. Left to right: Image of the scene. Input 3D points (blue, 1892 in total) and points (green) close to the cutting plane. Tangent planes estimated using no regularization. Solution with regularization.

### 3.2. Sparse Multiple View Stereo

Next we consider the problem of multiple view stereo using sparse features. Given a number of images (with known camera matrices) of a patch the orientation of the patch can be inferred by evaluating photo consistency with respect to the rest of the images. In this experiment we used the gate dataset (Figure 12), which consists of 34 images of a stone gate. We selected one of the images and ran our algorithm on all points visible in this image. In this case the data term will consist of two parts; a photo consistency part, and a distance part. The distance part measures the distance from the estimated tangent plane to the triangulated point along the viewing ray. Hence the estimated point will be the intersection of the tangent plane and the viewing ray. Other measurements such as reprojection error or Mahalanobis distance can also be used, but we choose this one for its simplicity. The photo consistency term takes a $20 \times 20$ patch around the current point and back projects using the current tangent plane estimate. To measure differences between patches we employ normalized cross correlation. In this case the connectivity of the points is known, since they are all visible in one image. We use the edges of a Delaunay triangulation in the image (removing edges that are to long) to define the neighborhood system. Figure 12 shows the results along a planar cut. Here we used the settings $\mu = 3$, $\tau = 0.56$. Further more the photo consistency cue was added with a weight of $0.1$. In the middle is the solution obtained by sampling 440 normals on the half sphere and selecting the best one based on photo consistency. To the right is the solution obtained with regularization.

## 4. Summary and Future Work

We presented a discrete optimization algorithm for curvature-based regularization using (only) pairwise potentials over extended labels describing surface orientation in addition to its position. As pointed out by A. Chambolle and by our area chair, there are related continuous [6] and discrete [5] mathematical models. Unlike [5], we represent a surface implicitly without assuming its topology. Data points are treated as noisy measurements. An interesting direction of further research is to extend our discrete pairwise framework to curvature-based image segmentation.

There are other applications besides stereo, multi-view reconstruction, or segmentation. For example, [2] has an interesting formulation of multi-object tracking as a problem of fitting global *parametric* models (e.g. splines) into locally detected features. As suggested by A. Delong, our piece-wise smooth regularization approach may work as an alternative method fusing local tracks into global *non-parametric* trajectories of objects.

## References

[1] M. Alexa, J. Behr, D. Cohen-or, S. Fleishman, D. Levin, and C. T. Silva. Computing and rendering point set surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 9:3–15, 2003. 2

[2] A. Andrienko, K. Schindler, and S. Roth. Discrete-continuous optimization for multi-target tracking. In *Proc. of CVPR 2012*. IEEE Computer Society, IEEE, June 2012. 8

[3] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, Cambridge, USA, 1987. 2

[4] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transations on Pattern Analysis and Machine Intelligence*, 2001. 1

[5] A. Bruckstein, A. Netravali, and T. Richardson. Epi-convergence of discrete elastica. *Applicable Analysis*, 79(1-2):137–171, 2001. 3, 8

[6] G. Citti and A. Sarti. A cortical based model of perceptual completion in the Roto-Translation space. *J. Math. Imaging and Vision*, 24(3):307–326, 2006. 8

[7] A. Delong, A. Osokin, H. Isack, and Y. Boykov. Fast approximate energy minimization with label costs. *International Journal of Computer Vision*, 96(1):1–27, January 2012. 2, 5

[8] M. E. and T. C. Invariant smoothness measures for surfaces. *Advances in Computational Mathematics*, 8(1):307–310, 1998. 6

[9] N. El-Zehiry and L. Grady. Fast global optimization of curvature. In *Proc. of CVPR 2010*. IEEE Computer Society, IEEE, June 2010. 1

[10] H. Huang, D. Li, H. Zhang, U. Ascher, and D. Cohen-Or. Consolidation of unorganized point clouds for surface reconstruction. In *ACM Trans. on Graphics*, 2009. 2

[11] H. Isack and Y. Boykov. Energy-Based Geometric Multi-model Fitting. *International Journal of Computer Vision*, 97(2):123–147, April 2012. 2, 5

[12] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28:1568–1583, October 2006. 1, 2, 5, 6

[13] C. Lange, K. Polthier, T. Berlin, and T. Berlin. Anisotropic smoothing of point sets. *Computer Aided Geometric Design*, 22:2005, 2005. 2

[14] Y. Lipman, D. Cohen-Or, D. Levin, and H. Tal-Ezer. Parameterization-free projection for geometry reconstruction. In *ACM Trans. om Graphics.*, 2007. 2

[15] T. Schoenemann, F. Kahl, and D. Cremers. Curvature regularity for region-based image segmentation and inpainting: A linear programming relaxation. In *Int. Conf. on Computer Vision*, Kyoto, Japan, 2009. 1, 2

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

[17] R. Szeliski, D. Tonnesen, and D. Terzopoulos. Modeling surfaces of arbitrary topology with dynamic particles. In *Computer Vision and Pattern Recognition, 1993. Proceedings CVPR '93., 1993 IEEE Computer Society Conference on*, 1993. 2

[18] J. Taylor, A. D. Jepson, and K. N. Kutulakos. Non-rigid structure from locally-rigid motion. In *IEEE Int. Conf. of Computer Vision and Pattern Recognition*, San Francisco, 2010. 2

[19] G. Vogiatzis, C. H. Esteban, P. H. S. Torr, and R. Cipolla. Multiview stereo via volumetric graph-cuts and occlusion robust photo-consistency. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(12):2241–2246, 2007. 1

[20] T. Willmore. Mean curvature of riemannian immersions. *J. London Math. Society*, 11:307–310, 1971. 6

[21] O. J. Woodford, P. H. S. Torr, I. D. Reid, and A. W. Fitzgibbon. Global stereo reconstruction under second order smoothness priors. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, Anchorage, Alaska, 2008. 1, 5, 6