Courbure discrète : théorie et applications

Rencontre organisée par :
Laurent Najman and Pascal Romon

18-22 novembre 2013

Carl Olsson and Yuri Boykov

Tangential Approximation of Surfaces

Vol. 3, n° 1 (2013), p. 51-60.

<http://acirm.cedram.org/item?id=ACIRM_2013__3_1_51_0>

Centre international de rencontres mathématiques
U.M.S. 822 C.N.R.S./S.M.F.
Luminy (Marseille) France

Actes des rencontres du C.I.R.M.
Vol. **3** n° 1 (2013) 51-60

Tangential Approximation of Surfaces

Carl Olsson and Yuri Boykov

Abstract

In the Computer Vision community it is a common belief that higher order smoothness, such as curvature, should be modeled using higher order interactions. For example, 2nd order derivatives for deformable (active) contours are represented by triple cliques. Similarly, the 2nd order regularization methods in stereo predominantly use MRF models with scalar (1D) disparity labels and triple clique interactions. In this paper we give an overview of an energy minimization framework for *tangential approximation of surfaces* developed in [21, 22]. The framework uses higher dimensional labels to encode second order smoothness with pairwise interactions. Hence, many generic optimization algorithms (e.g. message passing, graph cut, etc.) can be used to optimize the proposed regularization functionals. The accuracy of our approach for representing curvature is demonstrated by theoretical and empirical results on real data sets from multi-view reconstruction and stereo.

1. Introduction

Surface estimation from point measurements is an important problem in Computer Vision. This paper gives an overview of an energy minimization framework for *tangential approximation of surfaces* developed in [21, 22]. The approach assumes that we are given noisy position estimates/probabilities of points sampled from the surface. To each point measurement our method assigns a tangent plane that represents the exact surface position and orientation. 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 includes a regularization term that encourages tangents to agree on some underlying piece-wise smooth surface by estimating 2nd order smoothness (such as curvature or 2nd derivative).

Our framework is based on a graphical model. In general, graphical models are widely used in vision for problems like dense stereo [30, 5], surface estimation [29], image segmentation [5], inpainting [25], etc. Perhaps the most common regularizer is *length* (or *area* in 3D). This smoothness criteria corresponds to graphical models with simple pair-wise potentials, which admit very efficient global optimization algorithms like [5, 15].

Recently, the vision community has begun to actively explore models using curvature-based regularization or similar second-order smoothness priors [25, 9, 26, 30]. Typically, 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 [21, 22] we take an alternative approach. 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. We therefore get an easier to handle pairwise interaction at the expense of a larger search space. This makes it possible to evaluate curvature using only pairwise interactions. Thus, standard combinatorial optimization methods, e.g. TRW-S [15], can be readily applied. The idea of simplifying optimization problems by lifting the label space has also been applied in the context of variational formulations for optical flow and segmentation [20, 23, 6]. To

Text presented during the meeting "Discrete curvature: Theory and applications" organized by Laurent Najman and Pascal Romon. 18-22 novembre 2013, C.I.R.M. (Luminy).

demonstrate out general approach we consider two applications; Point Cloud Regularization and Stereo Reconstruction.

1.1. **Regularization of Point Clouds.** Point clouds obtained from structure-from-motion type procedures [11] typically contain significant amounts of noise. One reason for this is that there is no regularization term that encourages point positions to agree on smooth surfaces in the final solution. To regularize the point cloud we estimate a “true" position and a tangent plane, for each point, such that neighboring points agree on an underlying piecewise smooth surface. The tangent interaction that we use can be shown to estimate normal curvature of the surface. Our approach does not assume any particular topology, nor does it have to be closed or “orientable".

Piece-wise constant regularization has previously been applied to similar geometric problems, e.g. [13, 8]. 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 [28] in the context of non-rigid structure from motion, similar *local-to-global* approach may work even without regularization. In general, however, piecewise 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 [12]. Popular approaches are least squares surfaces (MLS) [1], locally optimal projections (LOP) [19], and anisotropic point cloud diffusion [16]. However, none of these methods regularize curvature. Perhaps, the closest to our approach is [27] 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).

1.2. **Stereo Reconstruction.** The goal of stereo reconstruction is to compute a depth estimate for every pixel in an image. Doing so requires that every pixel in the image is matched to a corresponding pixel in another image. Due to ambiguous texture this matching is rarely unique and as a result the stereo problem is most often ill posed. Resolving these ambiguities requires adding knowledge of what types of surfaces that we can expect to see in natural scenes, in the form of regularization.

First order methods [5, 14, 10] often implicitly assume fronto-parallel planes. For example, standard piecewise smooth (e.g. truncated linear or quadratic) pairwise regularization potentials assign higher cost to surfaces with larger tilt with respect to the camera [5]. To model surfaces more accurately Birchfeld and Tomasi [2] introduced 3D-labels corresponding to arbitrary 3D planes. However, this approach is limited to piecewise planar scenes. To address more general scenes recent papers use 2nd derivative regularization [22, 18, 30]. There are two ways of modeling such higher order smoothness potentials. Woodford et al. [30] retain the scalar disparity labels while using triple-cliques to penalize 2nd derivatives of the reconstructed surface. This encourages near planar smooth disparity maps. The optimization problem is however made more difficult due to the introduction of non-submodular triple interactions. In contrast, [22, 18] use 3D-labels corresponding to tangent planes to encode 2nd order smoothness as pairwise interactions. It is shown in [22] that in contrast to the triple-cliques used by Woodford et al. [30] the 3D-label formulation is often submodular (or near submodular) making fusion moves easier to solve optimally using standard methods like Roof duality [24].

1.3. **Optimization Background.** In this section we briefly review the concept of fusion moves [17]. Given two (possibly very different) solution proposals the fusion move tries to compute the best possible combination, with respect to the objective function, by selecting the best parts from each proposal. This allows modifying a large number of pixels simultaneously thereby escaping poor local minima.

Consider the optimization of an arbitrary second order pseudo-boolean function (PBF) of n variables, usually expressed as,

$$\min_{\boldsymbol{x}\in\boldsymbol{L}^n} E(\boldsymbol{x}) = \min_{\boldsymbol{x}\in\boldsymbol{L}^n} \sum_{p\in\mathcal{V}} U_p(x_p) + \sum_{(p,q)\in\mathcal{E}} V_{pq}(x_p, x_q) \tag{1.1}$$

where $\boldsymbol{L} = \{0, 1\}$. We think of the objective function as being defined on a graph $G = (\mathcal{V}, \mathcal{E})$. To each node $p \in \mathcal{V}$ we want to assign a binary value, at a cost specified by the unary term U_p. In addition, for each edge $(p, q) \in \mathcal{E}$ there is a term V_{pq} that specifies a costs for combinations of assignments to p and q. If V_{pq} is submodular, that is

$$V_{pq}(0,0) + V_{pq}(1,1) \leq V_{pq}(0,1) + V_{pq}(1,0), \tag{1.2}$$

this can be efficiently solved [5]. Even with V_{pq} non-submodular roof-duality (RD) [4, 24] can be used. RD will give a partial solution where labeled variables are guaranteed to be correct for an optimal solution and some variables are left *unlabeled*.

Lempitsky et al. [17] proposed a way to minimize (1.1) when $\boldsymbol{L} = \mathbb{R}$. Given two assignments $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$ we fuse them into a new one with lower energy by solving

$$\min_{\boldsymbol{z}\in\{0,1\}^n} E\big(\boldsymbol{z}\cdot\boldsymbol{x}_0 + (\mathbf{1}-\boldsymbol{z})\cdot\boldsymbol{x}_1\big), \tag{1.3}$$

where $\cdot$ is element-wise multiplication. If we solve (1.3) using RD and then set $\boldsymbol{z} = \mathbf{0}$ for all unlabeled variables the autarky results in [24] gives

$$E\left(\boldsymbol{z}\cdot\boldsymbol{x}_0 + (\mathbf{1}-\boldsymbol{z})\cdot\boldsymbol{x}_1\right) \leq \min\left(E\left(\boldsymbol{x}_0\right), E\left(\boldsymbol{x}_1\right)\right). \tag{1.4}$$

Therefore we can iteratively minimize (1.1) by proposing new solutions and fusing them with the old solution.

The possibility to decrease the energy for each fusion move is an attractive feature, however there is no guarantee on how many variables will be labeled in each fusion move. For submodular fusion moves we are guaranteed to label all variables. Minimizing a submodular function is also faster in practice [24].

2. An Energy Approach to Tangential Estimation

Next we present the approach for tangent estimation. Recall the general formulation of our problem: Given noisy point position estimates/probabilities we want to estimate the true point positions by enforcing smoothness of the unknown underlying surface. Note that estimating 2nd order smoothness of a surface from point samples requires at least 3 points; first derivatives are estimated using pairs of points then second derivatives using pairs of pairs. To avoid triple interactions like these our labels encode both point position and derivatives using a tangent plane. In this way we are able to encode our smoothness using pairwise interactions at the expense of having to search a larger label space.

We can think of the method as an energy minimization problem on a graph $G = (\mathcal{V}, \mathcal{E})$. To each node $p \in \mathcal{V}$ we will assign a tangent $\boldsymbol{t}_p = (\mathbf{n}_p, a_p)$, where $\mathbf{n}_p \in \mathbb{R}^3$, $||\mathbf{n}_p|| = 1$ and $a_p \in \mathbb{R}$. Note that we assume that each tangent also provides (a unique) estimate of the sought point position $\boldsymbol{p}$. For example, in Section 3 $\boldsymbol{p}$ will be the point on the tangent that has the smallest distance to the point measurement. This limits the dimension of our labels to 3. If this is not the case then in addition to the tangent itself we would have to estimate a position on the plane requiring 2 additional degrees of freedom.

The energy has the form

$$E(\boldsymbol{t}) = \sum_{(p,q)\in\mathcal{E}} V_{pq}(\boldsymbol{t}_p, \boldsymbol{t}_q) + \sum_{p\in\mathcal{V}} U_p(\boldsymbol{t}_p). \tag{2.1}$$

The unary term U_p only depends on the tangent assigned to p. This term is sometimes refereed to as data term since it is constructed from point measurements/probabilities. In Section 3 we will use the distance between the estimated and the measured point and in Section 4 it will be based on photo-consistency.

The pairwise term V_{pq} provides the regularization. This term will encourage assignments where neighboring nodes have similar tangents. Here the neighborhood is defined by the edge set $\mathcal{E}$. Specifically the term V_{pq} works by considering the effect of switching the assigned tangent at q

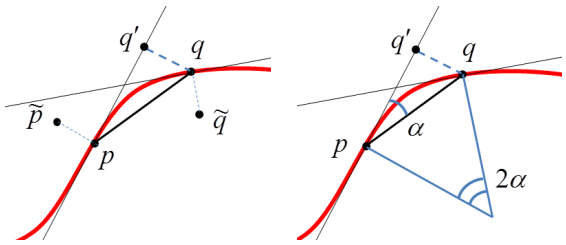

Figure 3.1. Left: pairwise interaction approximating curvature corresponds to quotient (3.12). Right: the quotient $\frac{|q-q'|}{|p-q|^2}$ yields half the curvature at p under the assumption that p and q belong to a constant curvature segment.

to its neighboring tangent $\boldsymbol{t}_p$. Intuitively, if the surface is smooth, then the tangents at nearby nodes p and q should be similar. Therefore the point estimations should not change much when switching tangents. An alternative way of thinking of this type of interaction is as the deviation from the 1st order approximation at p (tangent $\boldsymbol{t}_p$) measured at q, which is a measure of the 2nd order properties of the surface. Due to the restriction to pairwise potentials we need to measure surface smoothness indirectly by estimating smoothness of curves on the surface. More complex regularization terms like Gaussian curvature require higher order potentials.

Note that we do not explicitly construct any surface. It is only locally implicitly represented using the tangents. Therefore our approach does not assume any particular surface topology, nor does it have to be closed or "orientable". Furthermore, in contrast to methods that estimate smoothness using explicit polygonal approximations, such as triangulations, we do not need to know in which order the points are connected. For each node p we just compute smoothness estimates using all points in a small neighborhood.

3. Point Cloud Regularization

We assume that points $\tilde{\boldsymbol{p}}$ and $\tilde{\boldsymbol{q}}$ are noisy measurements of the points $\boldsymbol{p}$ and $\boldsymbol{q}$ on a curve on the underlying surface (see Figure 3.1). To $\tilde{\boldsymbol{p}}$ and $\tilde{\boldsymbol{q}}$ we assign tangents $(\mathbf{n}_p, a_p)$ and $(\mathbf{n}_q, a_q)$. The estimated "true" points $\boldsymbol{p}$ and $\boldsymbol{q}$ will be the orthogonal projections of $\tilde{\boldsymbol{p}}$ and $\tilde{\boldsymbol{q}}$ onto the assigned tangents, that is

$$\boldsymbol{p} = \tilde{\boldsymbol{p}} - (\mathbf{n}_p^\top \tilde{\boldsymbol{p}} + a_p)\mathbf{n}_p \tag{3.1}$$

$$\boldsymbol{q} = \tilde{\boldsymbol{p}} - (\mathbf{n}_q^\top \tilde{\boldsymbol{q}} + a_q)\mathbf{n}_q. \tag{3.2}$$

Given the estimations $\boldsymbol{p}$ and $\boldsymbol{q}$ and their tangents we would like to estimate how smooth the underlying curve is. There are several ways of doing this. Bruckstein et al. [7] 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 tangents with pairwise interactions. To measure the difference between the two tangent planes we therefore compute the projections (see Fig.3.1 left)

$$\boldsymbol{p}' = \boldsymbol{p} - (\mathbf{n}_q^\top \boldsymbol{p} + a_q)\mathbf{n}_q \tag{3.3}$$

$$\boldsymbol{q}' = \boldsymbol{q} - (\mathbf{n}_p^\top \boldsymbol{q} + a_p)\mathbf{n}_p, \tag{3.4}$$

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

$$|\boldsymbol{p}' - \boldsymbol{p}| + |\boldsymbol{q}' - \boldsymbol{q}| \tag{3.5}$$

should be small since the tangents are close to each other. The term (3.5) is closely related to curvature. To see this we assume that $(\mathbf{n}_p, a_p)$, $(\mathbf{n}_q, a_q)$ are the true tangent planes at the points

$\boldsymbol{p}$ and $\boldsymbol{q}$, and that

$$\boldsymbol{p} = \alpha(t_1) \tag{3.6}$$

$$\boldsymbol{q} = \alpha(t_2), \tag{3.7}$$

where α 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 $\mathbf{n}_p^\top \boldsymbol{p} + a_p = 0$ we have

$$|\boldsymbol{q} - \boldsymbol{q}'| = |\mathbf{n}_p^\top \boldsymbol{q} + a_p| = |\mathbf{n}_p^\top (\boldsymbol{q} - \boldsymbol{p})| \tag{3.8}$$

Using the Taylor expansion of α at t_1

$$\alpha(t) = \alpha(t_1) + \dot{\alpha}(t_1)(t - t_1) + \frac{1}{2}\ddot{\alpha}(t_1)(t - t_1)^2 + O((t - t_1)^3), \tag{3.9}$$

together with (3.6)-(3.7), the term (3.8) can be written

$$|\mathbf{n}_p^\top \left(\dot{\alpha}(t_1)(t_2 - t_1) + \frac{1}{2}\ddot{\alpha}(t_1)(t_2 - t_1)^2\right) + O((t - t_1)^3)|. \tag{3.10}$$

Since $\dot{\alpha}(t_1)$ is the tangent at $\boldsymbol{p}$ it is perpendicular to the normal $\mathbf{n}_p$ so the first term vanishes. Furthermore, since α is parametrized by arc length, $\ddot{\alpha}$ is perpendicular to the tangent $\dot{\alpha}$. Therefore $\mathbf{n}_p^\top \ddot{\alpha}(t_1)$ is only the length of $\ddot{\alpha}(t_1)$ (plus or minus, depending on the direction of $\mathbf{n}_p$), which is the curvature. Hence if we divide $|\boldsymbol{q} - \boldsymbol{q}'|$ 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

$$|\boldsymbol{p} - \boldsymbol{q}|^2 = |\dot{\alpha}(t_1)|(t_2 - t_1)^2 + O((t_2 - t_1)^3) \tag{3.11}$$

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

$$\frac{|\boldsymbol{q} - \boldsymbol{q}'|}{|\boldsymbol{p} - \boldsymbol{q}|^2} = \frac{\frac{1}{2}|\mathbf{n}_p^\top \ddot{\alpha}(t_1) + O(t_2 - t_1)|}{|1 + O(t_2 - t_1)|}. \tag{3.12}$$

This expression will tend to half of the curvature when t_2 tends to t_1, and therefore approximates curvature well if $\boldsymbol{p}$ and $\boldsymbol{q}$ are close enough. Note that if we think of the normals as coming form an underlying surface then the interaction can be interpreted as measuring normal curvature of the surface in the direction $\boldsymbol{q}' - \boldsymbol{p}$.

Figures 3.2 shows the computation of the quotient (3.12) for the curve $\sqrt{3}\cos(t)$. To approximate the integral of the absolute curvature we use

$$\frac{1}{2}\int |\kappa| d\sigma \approx \sum_i \frac{|\boldsymbol{p}_{i+1} - \boldsymbol{p'}_{i+1}|}{|\boldsymbol{p}_{i+1} - \boldsymbol{p}_i|}, \tag{3.13}$$

where $\boldsymbol{p}_i$ 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 3.2 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 $\boldsymbol{p}$ and $\boldsymbol{q}$ is on a circle with center at the origin, see Figure 3.1 right. If we parametrize the curve by arc length the angle 2α is t/r. By the cosine theorem we get

$$|\boldsymbol{p} - \boldsymbol{q}|^2 = 2r^2(1 - \cos(\frac{t}{r})) \tag{3.14}$$

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

$$|q - q'| = |p - q'| \sin(\frac{t}{2r}) \tag{3.15}$$

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

$$\frac{|\boldsymbol{q} - \boldsymbol{q}'|}{|\boldsymbol{p} - \boldsymbol{q}|^2} = \frac{\sin(\frac{t}{2r})}{\sqrt{2r^2(1 - \cos(\frac{t}{r}))}} = \frac{\sin(\frac{t}{2r})}{2r|\sin(\frac{t}{2r})|} = \frac{1}{2r}, \tag{3.16}$$

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

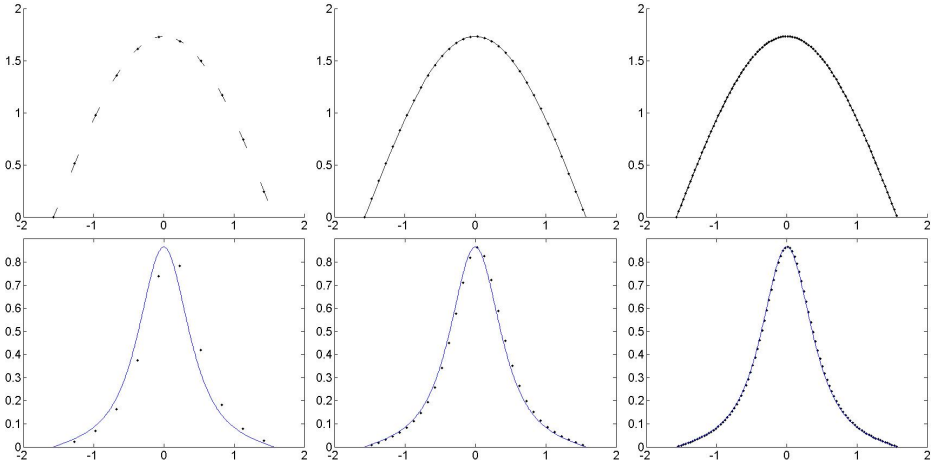

Figure 3.2. Computation of the curvature of $\sqrt{3}\cos(t)$, $-\pi/2 \leq t \leq \pi/2$. Top: The sample points and their tangent planes for various sampling rates. Bottom: The computed curvature at the sample points, and the true curvature of the function. The values of the integral approximations are 59.6265, 59.9600 and 59.9961 degrees respectively (converted from radians).

3.1. **Experiments.** In this section we present some experimental results. For the energy we use (2.1) with

$$V_{pq}(\boldsymbol{t}_p, \boldsymbol{t}_q) = \min\left(\frac{|\boldsymbol{q}-\boldsymbol{q}'|}{|\boldsymbol{p}-\boldsymbol{q}|}, \tau\right), \tag{3.17}$$

where τ is a threshold that ensures that we do no over penalize transitions between surfaces. Note that we consider the edges of $\mathcal{E}$ to be undirected. Therefore if V_{pq} is present in the energy then so is V_{qp} and therefore the energy is symmetric. Since all nodes do not necessarily have the same number of neighbors we weight the pairwise interaction V_{pq} with one over the number of neighbors of p.

The unary term is simply the squared distance $U_p(\boldsymbol{t}_p) = |\tilde{\boldsymbol{p}} - \boldsymbol{p}|^2$. For further details see [21].

We apply it to the two real datasets depicted in figures 3.3 and 3.4. 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. Figure 3.3 and 3.4 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.

To generate proposals for the fusion moves we use an approach similar to that of [8, 13]. First candidate tangents are computed using random sampling. These are then refined using local optimization of the energy and fused with the current solution.

For the computations we used the code available from [15] and the Matlab wrapper from [30]. Since our energy is not of any of the standard forms supported by [15], we used the general lookup table form (which is highly inefficient according [15]) to setup our potentials. Specialized software should therefore be able to speed up these computations considerably.

4. Stereo Reconstruction

In dense stereo the objective is to compute a depth $z(p)$ (distance to the surface/object from the image) for each pixel p in an images. The result is an estimate of the surface geometry for all points on the object that are visible in the image.

The cost of assigning a particular depth to a pixel is based on comparison of pixel appearances. From the pixel coordinate and the given depth it is possible to compute the position of the corresponding point in a second image. Comparing the appearances (e.g. RGB-values) of these two pixels gives a cost of that depth. Individual depth estimates are usually unreliable due

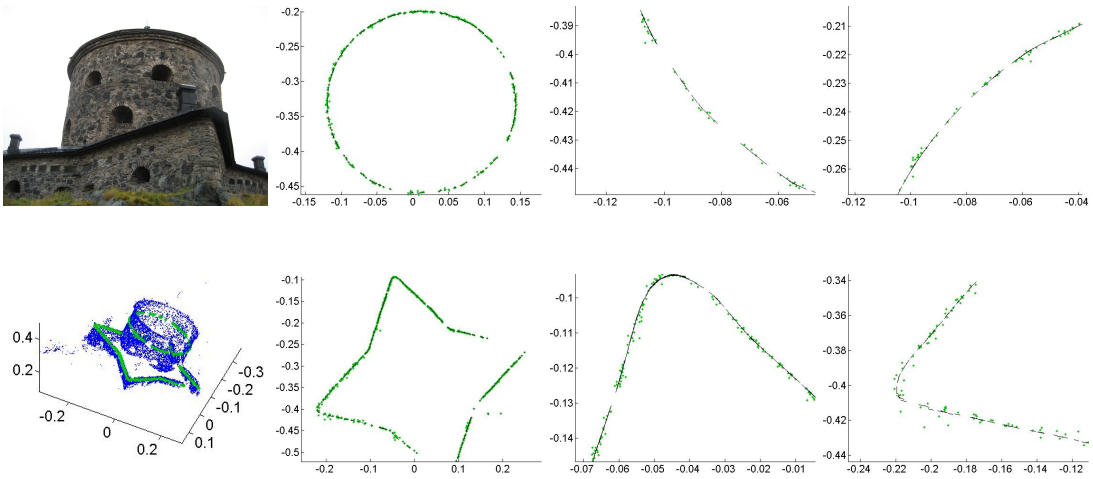

Figure 3.3. Visualization of the resulting tangent planes along two planar cuts. *1st column*: Image of the scene, Input 3D points (blue, 18270 in total) and points (green) close to the two cutting planes. *2nd column*: Points (green) projected onto the plane and line segments (black) obtained when intersecting the corresponding tangent planes with the cutting plane. *3rd and 4th columns*: Two closeups.

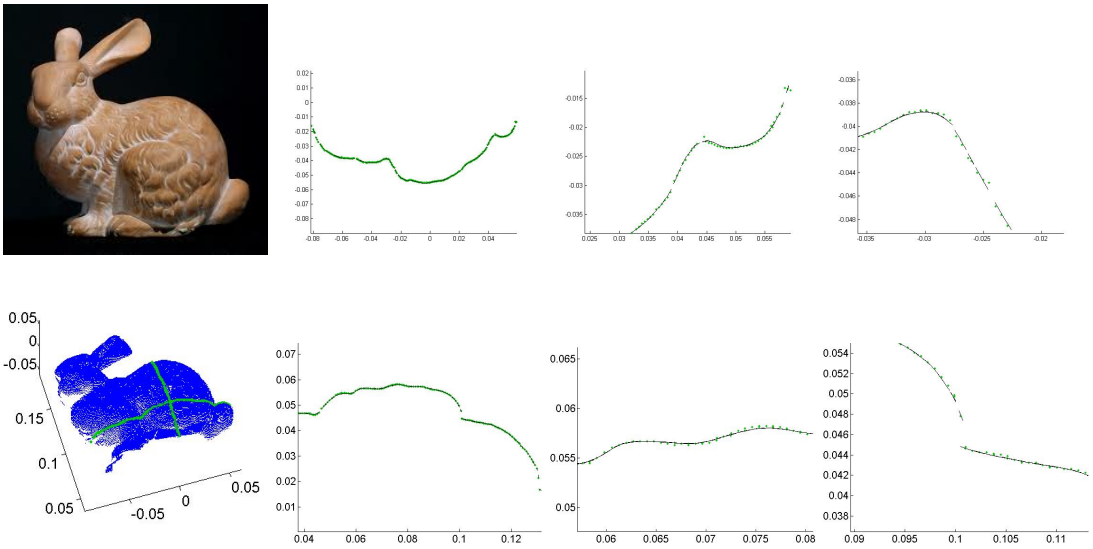

Figure 3.4. Visualization of the resulting tangent planes along two planar cuts. *1st column*: Image of the scene, Input 3D points (blue, 40256 in total) and points (green) close to the two cutting planes. *2nd column*: Points (green) projected onto the plane and line segments (black) obtained when intersecting the corresponding tangent planes with the cutting plane. *3rd and 4th columns*: Two closeups.

to noise and ambiguous texture and we therefore need to apply regularization. However directly penalizing 2nd derivative of the depth function is not a good idea. In general the projection of a plane will not yield a linear depth function unless the camera is affine (which can be seen from (4.4) below). Hence, such an energy would assign a 3D-plane a nonzero penalty. Therefore we will instead measure the deviation from the tangent plane along the viewing ray.

Let $\boldsymbol{p}_h$ and $\boldsymbol{q}_h$ denote the homogeneous coordinates (with third coordinate 1) and $\boldsymbol{p}$ and $\boldsymbol{q}$ regular Cartesian coordinates of the two pixels p and q. We will assume a pinhole camera model where the center camera has been calibrated and normalized to be of form $[\boldsymbol{I}\ \boldsymbol{0}]$. Given a function $z : \mathcal{I} \mapsto \mathbb{R}_+$ that gives a depth for each pixel the 3D points $\boldsymbol{P}$ and $\boldsymbol{Q}$ corresponding to p and q can

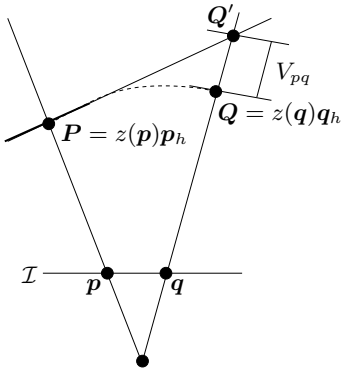

Figure 4.1. Left: Geometric interpretation of the stereo smoothness interaction.

be computed (in regular Cartesian coordinates) using the simple formulas

$$\boldsymbol{P} = z(\boldsymbol{p})\boldsymbol{p}_h, \tag{4.1}$$

$$\boldsymbol{Q} = z(\boldsymbol{q})\boldsymbol{q}_h. \tag{4.2}$$

By $\boldsymbol{t}_p = (\mathbf{n}_p, a_p)$, where $\mathbf{n}_p \in \mathbb{R}^3$, $\|\mathbf{n}_p\| = 1$ and $a_p \in \mathbb{R}$ we denote the tangent plane at p given by the equation

$$\mathbf{n}_p^T \boldsymbol{x} + a_p = 0. \tag{4.3}$$

Consider the intersection point $\boldsymbol{Q}'$ between the viewing ray at $\boldsymbol{q}$ and the tangent plane at $\boldsymbol{p}$ (see Figure 4.1). We let $\mathcal{T}_p z : \mathcal{I} \mapsto \mathbb{R}_+$ be the depth function of the tangent plane at point $\boldsymbol{p}$, that is, $\boldsymbol{Q}' = \mathcal{T}_p z\,(\boldsymbol{q})\,\boldsymbol{q}_h$. We can calculate the tangent function using

$$\mathcal{T}_p z\,(\boldsymbol{q}) = -\frac{a_p}{\mathbf{n}_p^T \boldsymbol{q}_h}. \tag{4.4}$$

(Here we are assuming that the viewing ray is not completely contained in the tangent plane.) Note that even though this function represents a plane in 3D it is usually not linear in $\boldsymbol{q}$.

To encourage smooth assignments we use the cost

$$V_{pq}(\boldsymbol{t}_p, \boldsymbol{t}_q) = \|\boldsymbol{Q} - \boldsymbol{Q}'\| = |\mathcal{T}_p z\,(\boldsymbol{q}) - z(\boldsymbol{q})|\,\|\boldsymbol{q}_h\|. \tag{4.5}$$

The geometric interpretation of this expression can be seen in Figure 4.1. Given the estimated tangent plane at $\boldsymbol{p}$ and the depth at $\boldsymbol{q}$ the interaction computes the distance between the estimated 3D point and the tangent plane along the viewing ray. The smoothness term will penalize deviations from planes and thereby encourage solutions with small second derivatives. Similarly to the previous section it can be shown that this interaction will penalize

$$\left| \frac{(z')^2 - zz''}{2z} \right|. \tag{4.6}$$

The term $(z')^2/z$ compensates for the fact that viewing rays are not parallel in the pinhole camera model. (In case of parallel rays the interaction just approximates $|z''|$.) The reason for not using the same interaction as in Section 3 for stereo is illustrated in Figure 4.2. The interaction from Section 3 can be interpreted as fitting a constant curvature segment between $\boldsymbol{p}$ and $\boldsymbol{q}$ with tangent $\boldsymbol{t}_p$. This only works well if $\boldsymbol{p}$ and $\boldsymbol{q}$ are close (recall that $\boldsymbol{p}$ and $\boldsymbol{q}$ were 3D points in Section 3), since this is when the Taylor expansion is valid. As can be seen in Figure 4.2 for large depth differences the curvature will be significantly underestimated. In contrast the interaction presented in this section can be interpreted as an approximation with $\frac{(z')^2 - zz''}{2z}$ which is a more reasonable approximation. In addition it is valid as long as the pixel coordinates $\boldsymbol{p}$ and $\boldsymbol{q}$ are close and therefore it is not sensitive to large depth differences between pixels.

4.1. **Experiments.** Next we evaluate the proposed framework on a couple of multiple view stereo data sets. For the energy we use

$$V_{pq}(\boldsymbol{t}_p, \boldsymbol{t}_q) = \min(\boldsymbol{Q} - \boldsymbol{Q}', \tau). \tag{4.7}$$

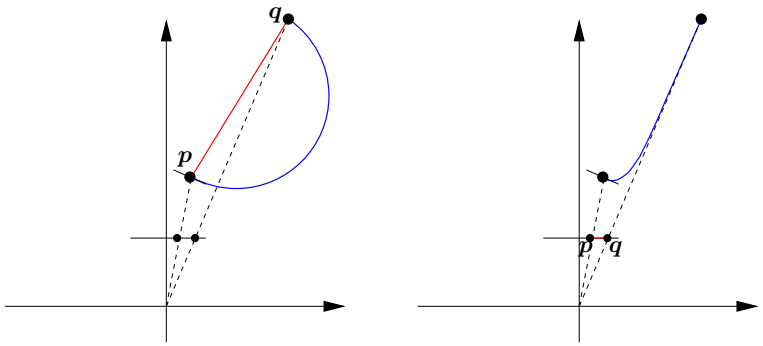

Figure 4.2. Differences between the two interactions in Section 3 and Section 4.

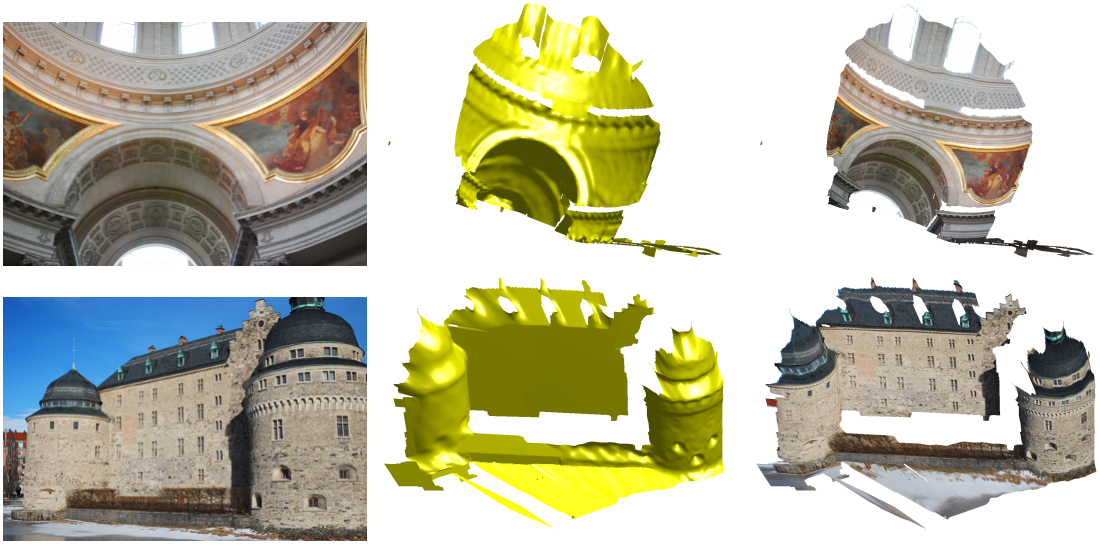

Figure 4.3. Resulting surface estimations. Left: Image. Middle: Estimated surface. Right: Estimated surface with texture.

For the data term U_p we use normalized cross correlation (NCC) computed at different possible depths. Quadratic interpolation is used to compute values in between the sampled depths. See [22] for further details on implementations.

To generate proposals we use random sampling of planes and local optimization. In the sampling step we select a pixel and a small neighborhood around it. Using the best local maximum of the normalized cross correlation for each viewing ray in the neighborhood we create a 3D cloud and fit a plane using RANSAC. Fusion moves with these planar proposals are particularly effective and can in fact be shown to be submodular [22].

Figure 4.3 shows the result of applying this approach to two datasets.

References

[1] Marc Alexa, Johannes Behr, Daniel Cohen-or, Shachar Fleishman, David Levin, and Claudio T. Silva. Computing and rendering point set surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 9:3–15, 2003.

[2] S. Birchfield and C. Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *International Conference on Computer Vision*, 1999.

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

[4] E. Boros and P.L. Hammer. Pseudo-boolean optimization. *Discrete applied mathematics*, 123(1):155–225, 2002.

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

[6] Kristian Bredies, Thomas Pock, and Benedikt Wirth. Convex relaxation of a class of vertex penalizing functionals. *Journal of Mathematical Imaging and Vision*, pages 1–25, 2012.

[7] A.M. Bruckstein, A.N. Netravali, and T.J. Richardson. Epi-convergence of discrete elastica. *Applicable Analysis*, 79(1-2):137–171, 2001.
[8] Andrew Delong, Anton Osokin, Hossam Isack, and Yuri Boykov. Fast approximate energy minimization with label costs. *International Journal of Computer Vision*, 96(1):1–27, January 2012.
[9] Noha El-Zehiry and Leo Grady. Fast global optimization of curvature. In *Proc. of CVPR 2010*. IEEE Computer Society, IEEE, June 2010.
[10] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient belief propagation for early vision. *Int. J. Comput. Vision*, 70(1):41–54, October 2006.
[11] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.
[12] H. Huang, D. Li, Hao Zhang, Uri Ascher, and Daniel Cohen-Or. Consolidation of unorganized point clouds for surface reconstruction. In *ACM Trans. on Graphics*, 2009.
[13] Hossam Isack and Yuri Boykov. Energy-Based Geometric Multi-model Fitting. *International Journal of Computer Vision*, 97(2):123–147, April 2012.
[14] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. In *European Conf. on Computer Vision*, volume III, pages 82–96, Copenhagen, Denmark, 2002.
[15] Vladimir Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28:1568–1583, October 2006.
[16] Carsten Lange and Konrad Polthier. Anisotropic smoothing of point sets. *Computer Aided Geometric Design*, 22:2005, 2005.
[17] V. S. Lempitsky, C. Rother, S. Roth, and A. Blake. Fusion moves for markov random field optimization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(8):1392–1405, 2010.
[18] G. Li and S.W. Zucker. Differential geometric inference in surface stereo. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(1):72–86, 2010.
[19] Yaron Lipman, Daniel Cohen-Or, David Levin, and Hillel Tal-Ezer. Parameterization-free projection for geometry reconstruction. In *ACM Trans. om Graphics.*, 2007.
[20] Tal Nir, Alfred M. Bruckstein, and Ron Kimmel. Over-parameterized variational optical flow. *Int. J. Comput. Vision*, 76(2):205–216, February 2008.
[21] C. Olsson and Y. Boykov. Curvature-based regularization for surface approximation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.
[22] C. Olsson, J. Ulén, and Y. Boykov. In defense of 3d-label stereo. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.
[23] Guy Rosman, Shachar Shem-tov, David Bitton, Tal Nir, Gilad Adiv, Ron Kimmel, Arie Feuer, and Alfred M. Bruckstein. Over-parameterized optical flow using a stereoscopic constraint. In *SSVM*, 2011.
[24] C. Rother, V. Kolmogorov, V. S. Lempitsky, and M. Szummer. Optimizing binary mrfs via extended roof duality. In *IEEE conf. on Computer Vision and Pattern Recognition*, 2007.
[25] 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.
[26] Petter Strandmark and Fredrik Kahl. Curvature regularization for curves and surfaces in a global optimization framework. In *EMMCVPR*, pages 205–218, 2011.
[27] 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.
[28] Jonathan Taylor, Allan D. Jepson, and Kiriakos N. Kutulakos. Non-rigid structure from locally-rigid motion. In *IEEE Int. Conf. of Computer Vision and Pattern Recognition*, San Francisco, 2010.
[29] George Vogiatzis, Carlos Hernández Esteban, Philip H. S. Torr, and Roberto Cipolla. Multiview stereo via volumetric graph-cuts and occlusion robust photo-consistency. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(12):2241–2246, 2007.
[30] O.J. Woodford, P.H.S. Torr, I.D. Reid, and A.W. Fitzgibbon. Global stereo reconstruction under second order smoothness priors. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009.

Centre for Mathematical Sciences, Lund University, SWEDEN • calle@maths.lth.se

Computer Science Department, University of Western Ontario, CANADA • yuri@csd.uwo.ca