Higher-order nonlinear priors for surface reconstruction

Tolga Tasdizen Ross Whitaker

University of Utah, School of Computing
Technical Report UUCS-02-016

School of Computing
University of Utah
Salt Lake City, UT 84112 USA

December 9, 2002

Abstract

For surface reconstruction problems with noisy and incomplete range data, a Bayesian estimation approach can improve the overall quality of the surfaces. The Bayesian approach to surface estimation relies on a likelihood term, which ties the surface estimate to the input data, and the prior, which ensures surface smoothness or continuity. This paper introduces a new high-order, nonlinear prior for surface reconstruction. The proposed prior can smooth complex, noisy surfaces, while preserving sharp, geometric features, and it is a natural generalization of edge-preserving methods in image processing, such as anisotropic diffusion. The use of these higher-order surface priors requires solving a fourth-order partial differential equation (PDE), which can be difficult with conventional numerical techniques. Our solution is based on the observation that the generalization of image processing to surfaces entails filtering the surface normals. This approach allows us to separate the problem into two second-order PDEs: one for processing the normals and one for refitting the surface. Furthermore, we implement the associated surface deformations using level sets, and thus the algorithm can accommodate very complex shapes with arbitrary and changing topologies. This paper gives the mathematical formulation and describes the numerical algorithms. We also present a quantitative analysis, which demonstrates the effectiveness of the algorithm, and show results using real and synthetic range data.

Chapter 1

Introduction

This paper addresses the problem of $3D$ surface reconstruction from multiple registered range images. The importance of high-quality surface reconstructions from range data is growing as range measurement technologies become more accurate and affordable. Despite the increased accuracy in these measurement devices, significant challenges to surface reconstruction remain. In particular, surface reconstruction is complicated by measurement noise and variations in measurement density. For instance, the measurement density is high at parts of the surface visible to multiple images. By contrast, parts of the surface that are occluded in all of the images have no data.

We model the uncertainties arising from measurement noise, overlapping measurements, and occlusion in a Bayesian framework, which formulates surface reconstruction as the maximization of a posterior probability function. According to Bayes rule, maximum a posteriori (MAP) estimators maximize the product of two distinct probabilities: the likelihood of the measurement data conditioned on the surface model and the prior probability distribution of the model. We refer to these two probabilities, respectively, as the *likelihood* and the *prior*. The major challenge of Bayesian surface reconstruction is the determination of these two quantities. This paper focuses on the prior probability distribution. Specifically, we attempt to correctly generalize statistical image reconstruction strategies [1, 2, 3, 4, 5] to surface reconstruction, and we propose higher-order nonlinear feature-preserving surface priors as the solution to this problem.

In previous work [6], we propose that the natural generalization of image processing to surfaces occurs via the *surface normal vectors* and show results for post-processing of noisy surfaces. For example, a smooth surface is one that has smoothly varying normals. In this paper, we present the processing of surface normals as a computational strategy for

implementing higher-order priors in the context of Bayesian surface reconstruction. In a variational optimization framework, penalty functions on the surface normals give rise to fourth-order partial differential equations (PDE). Our strategy is to solve such PDEs with a two step approach: (1) operate on the normal map of a surface, and (2) fit a surface to the processed normals and the range data. Iterating this two-step process, we can efficiently implement fourth-order PDEs by solving a set of coupled second-order PDEs. In this light, the differences between surface processing and image processing are threefold. Normals exist on a manifold (the surface) and cannot be processed using a flat metric, as is typically done with images. Normals are vector valued and constrained to be unit length; the processing techniques must accommodate this. Normals are coupled with the surface shape, thus the normals should drag the surface along as their values are modified during processing.

The remainder of this paper is organized as follows: Section 2 presents a discussion of the related work in the literature. Section 3 gives a brief summary of Bayesian image and surface reconstruction formulations. In Section 4, we will formulate higher-order feature preserving surface energy functions that give rise to a useful family of priors, and in Section 5 we will introduce a strategy to solve them. Section 6 will demonstrate results and compare them to results from previous methods, and discuss some properties of the proposed approach. Section 7 will present conclusions and directions for future research possibilities.

Chapter 2

Related Work

Surface reconstruction methods can be classified into high-level and low-level approaches. The high-level approaches are generally formulated as a problem of finding sets of geometric primitives that best represent the objects being measured [7, 8, 9, 10, 11]. High-level descriptions of scenes are useful for certain computer vision tasks such as object recognition; however, the complexity of the scenes that can be described by simple primitives is limited. Low-level methods are based on either parametric models such as surface meshes or volumetric models such as level sets. The literature demonstrates methods for zippering together meshes that describe different sides of an object [12], and fitting deformable models that expand inside a sequence of range images [13]. These approaches are suitable for use with high quality range images with relatively smaller amounts of noise. Several authors propose volumetric methods [14, 15, 16, 17, 18], which use the volume as the medium in which range measurements are combined, and then extract a surface from the volume after the data have been fused. The reconstructed surfaces are the zero-sets, or isosurfaces, of such volumes.

In this paper, we use the level set based Bayesian range map registration and surface reconstruction framework developed by Whitaker *et al.* [17, 19, 18]. This strategy uses maximum likelihood parameter estimation to register the views before combining multiple range images using a level set implementation that can represent any solid object, regardless of shape and topology. The combination of a Bayesian formulation with level set methods allows us to work with complex, noisy scenes. The Bayesian formulation requires that we determine the likelihood probability density function, which depends on the range sensor properties, and the prior probability density function, which models the space of the physical objects being measured. The role of the likelihood term is to force the fitted models to be good representations of the measured data. Whitaker *et al.* derive the likeli-

hood term from a line-of-sight error formulation, which is shown to accurately model laser range finders [17, 18]. This paper addresses the prior term. The prior serves to eliminate the measurement noise and other artifacts in the fitted model by forcing the it to adhere to certain expectations about the application domain or scene.

The use of priors have been extensively investigated in related work in image processing. For instance, Mumford and Shah formulate the problem of image segmentation in a variational framework with a Bayesian rationale [1, 2]. Nordstrom [3] reconciles Perona & Malik (P&M) anisotropic diffusion approach to edge detection [20] with the Mumford-Shah variational strategy. Other authors [21, 5, 22] also present an unified view of the reconstruction, non-linear diffusion and robust statistics approaches. As we will explain in Section 3.2, our method allows us to generalize Nordstrom's *biased anisotropic diffusion* approach for images to surfaces in 3D.

In contrast to the research in image processing, investigation of better surface priors has not been a major focus in previous surface reconstruction literature. This is partly due to the prevalent strategy of fitting shape primitives to data in computer vision. Primitives usually have only a few shape parameters, *i.e.*, height and radius for a cylinder. They impose their own structure on to the data and act as our prior belief about the contents of a scene, and hence, there is no need for a separate prior probability term in the reconstruction. Loosely structured parametric models such as surface meshes, and non-parametric implicit models such as level set surfaces of volumes, can represent complicated scenes; however, these lower level models do not impose a rigid shape structure. Therefore, prior probability terms must be included in surface reconstruction with such models for purposes of reducing the effects of noise and filling in surfaces where there is no data.

A simple prior for surface reconstruction is to penalize surface area [17, 19, 18]. A gradient descent on the surface area energy results in mean curvature flow (MCF). However, in the context of surface reconstruction, MCF suffers from several problems including volume shrinkage and elimination of sharp features (creases). A great deal of research focuses on modified second-order flows that produce better results than MCF. Using level set methods, several authors have proposed smoothing surfaces by weighted combinations of principle curvatures. For instance, Whitaker [23] proposes a nonlinear re-weighting scheme that favors the smaller curvature and preserves cylindrical structures. Faugeraus [24] proposes a smoothing by the minimum curvature.

A similar set of curvature-based algorithms have been developed for surface meshes. For instance, Taubin [25] proposes a non-shrinking Gaussian smoothing. Clarenz, *et al.* , [26] propose a modified MCF as an anisotropic diffusion of the surface. They threshold a weighted sum of the principle curvatures to determine the surface locations where edge sharpening is needed. Tangential displacement is added to the standard MCF at these lo-

cations for sharpening the edges. Another mesh-based modified MCF is proposed in [27] where a threshold on the mean curvature is used to stop over-smoothing. Taubin proposes a "linear anisotropic Laplacian operator" for meshes that is based on a separate processing of the normals [28]. Although, these flows produce results that tend to preserve sharp features, they are not a strict generalization of P&M anisotropic diffusion [20] from images to surfaces. Because they are based on re-weightings of curvature, these methods always smooth the surface in one direction or another. Thus, they do not exhibit a sharpening of details, which is achieved by the P&M equation (for images) through an inverse diffusion process.

We propose that the correct generalization of P&M anisotropic diffusion and related image processing methods entails the use of higher-order nonlinear priors. Furthermore, penalty functions on the curvature of surfaces (instead of area) can be used to generate a family of higher-order priors. One such penalty function is total curvature which is minimized by the intrinsic Laplacian of mean curvature flow. In $2D$, Chopp & Sethian [29] derive the intrinsic Laplacian of curvature for an implicit curve, and solve the resulting fourth-order PDE. In related works [30, 31], approximations of higher-order geometric surface flows have been applied to surface fairing in computer graphics.

Chapter 3

Bayesian reconstruction

This section provides a brief discussion of the Bayesian formulation for the image/surface reconstruction framework.

3.1 Surface reconstruction

Let $\mathscr{S}$ and M be a surface and the collective set of measured data, respectively. We will assume that M consists of a set of n registered range images, $\left\{R^{(j)}\right\}_{j=1}^{n}$. For all the examples in this paper, we used the registration methods described in [19, 18]. Then, the posterior probability of $\mathscr{S}$ given the data is

$$P(\mathscr{S} \mid M) = \frac{P(M \mid \mathscr{S})\, P(\mathscr{S})}{P(M)} \tag{3.1}$$

where $P(M \mid \mathscr{S})$ is the likelihood term and $P(\mathscr{S})$ is the prior. Because the goal is to find the surface that maximizes the posterior, the denominator $P(M)$, which is a constant normalization factor that is independent of $\mathscr{S}$, can be dropped. Typically, MAP estimators are implemented as minimizations of the negative logarithm of the posterior probability

$$\hat{\mathscr{S}} = \arg\inf_{\mathscr{S}} [-\ln P(M \mid \mathscr{S}) - \ln P(\mathscr{S})], \tag{3.2}$$

where $\hat{\mathscr{S}}$ is the estimator.

Using the independence of the range images, the log-likelihood can be expressed as a sum $\ln P(M \mid \mathscr{S}) = -\sum_{j=1}^{n} \ln P\left(R^{(j)} \mid \mathscr{S}\right)$. Whitaker *et al.* [17, 19, 18] formulate the conditional likelihood of a range image as a volume integral over the object Ω enclosed by $\mathscr{S}$. In this way the gradient descent for the log-likelihood term is expressed as a function, $f(\mathbf{x})$, where $\mathbf{x}$ denotes surface locations in the volume. The function $f(\mathbf{x})$ provides a weighted sum of the effects of measurements from different scans on the point $\mathbf{x}$. Consequently, the gradient descent that minimizes (3.2) is described by the surface motion

$$\frac{\partial \mathscr{S}}{\partial t} = -f(\mathscr{S})N - \delta P(\mathscr{S}), \tag{3.3}$$

where δ denotes the Euler-Lagrange operator and N is the surface normal. Notice that the effects of the likelihood and the prior terms on the gradient descent are additive; hence, they can be analyzed separately. The priors that we introduce in this paper are not particular to the range data reconstruction problem; they are suitable for use in any Bayesian surface reconstruction problem. For example, the results of this paper can also be applied to tomographical surface reconstruction [32].

3.2 Image reconstruction

In this section, we give a brief discussion of the use of priors in image reconstruction and their relationship to P&M anisotropic diffusion. Mumford and Shah [1, 2] propose an energy minimization approach for image reconstruction/segmentation. The Mumford-Shah energy is the sum of three terms: (i) the quadratic data-model discrepancy, (ii) a quadratic penalty on the variation of the piecewise smooth model over the image domain except on a set of discontinuities which are modeled by a separate binary model, and (iii) the length of the discontinuities. The first term is the data likelihood term, and the latter two terms act as the prior in this energy. This minimization problem is difficult to solve because of the dependence on binary functions.

Nordstrom [3] reconciled the Mumford-Shah energy minimization approach with the P&M anisotropic diffusion approach to edge detection [20]. This connection can be seen by looking at P&M diffusion from a variational perspective. The P&M diffusion PDE is the gradient descent derived by the first variation of

$$\int_U G\left(\| \nabla I \|^2\right) \, dx \, dy, \tag{3.4}$$

where U is the image domain. In its original form, P&M diffusion was introduced to replace Laplacian smoothing, which is equivalent to the solution of the heat equation $\partial I / \partial t =$

$\nabla \cdot \nabla I$, with a nonlinear PDE

$$\partial I / \partial t = \nabla \cdot \left[g \left(\| \nabla I \|^2 \right) \nabla I \right], \tag{3.5}$$

where I is the grey-level image and g, the derivative of G with respect to $\| \nabla I \|^2$, is the edge stopping function. Perona & Malik suggested using $g(x) = e^{-\|\nabla I\|^2 / 2\mu}$, where μ is a positive, free parameter that controls the level of contrast of edges that can affect the smoothing process. Notice that $g(\| \nabla I \|^2)$ approaches 1 for $\| \nabla I \| \ll \mu$ and 0 for $\| \nabla I \| \gg \mu$. Edges are generally associated with large image gradients, and thus diffusion across edges is stopped while relatively flat regions undergo smoothing. A mathematical analysis shows that solutions to (3.5) can actually exhibit an inverse diffusion near edges, and can enhance or sharpen smooth edges that have gradients greater than μ [33]. These properties make P&M diffusion a good candidate for edge detection.

Nordstrum's *biased anisotropic diffusion* converts the variational form of anisotropic diffusion to a reconstruction energy by adding a data term to (3.4). This yields the variational energy

$$\int_U \left[(I_o - I)^2 + G \left(\| \nabla I \|^2 \right) \right] \, dx \, dy, \tag{3.6}$$

where I_o is the input image. The non-linear anisotropic penalty on the variation of the model, $G(\| \nabla I \|^2)$, acts as the prior. This penalty term is practically equivalent to the sum of the quadratic model variation and the linear length of discontinuities terms in the Mumford-Shah formulation. Hence, with the addition of the data-model discrepancy term P&M diffusion is transformed into a variational framework without the need for an explicit binary image for modeling the discontinuities. This is an important practical advantage over minimizing the Mumford-Shah energy. In this paper, we use energies of the same form as Nordstrum's biased anisotropic diffusion.

Chapter 4

Geometric surface priors

Using the Gibbs distribution, probability distributions for priors are commonly expressed in terms of energy functions. Let $E(\mathscr{S})$ be a function that maps surfaces to scalar energy values. Then the prior can be constructed as

$$P(\mathscr{S}) = \frac{1}{Z} e^{-\alpha E(\mathscr{S})} \tag{4.1}$$

where α is a parameter that controls the sharpness of the distribution and Z is a normalizing constant which ensures that the probability distribution function integrates to unity [34]. According to (4.1) surfaces with lower energy states have a higher probability of occurring. Geiger and Yuille [34] observe that the parameter α reflects the strength of our belief in the prior. All surfaces are equally likely when $\alpha = 0$; whereas, for $\alpha \to \infty$ only those surfaces with the lowest possible energy have non-zero probabilities of occurrence. Substituting the Gibbs prior (4.1) for the prior term in (3.3) gives

$$-\delta \ln P(\mathscr{S}) = -\delta \ln \frac{1}{Z} e^{-\alpha E(\mathscr{S})} = -\delta \left(\ln \frac{1}{Z} - \alpha E(\mathscr{S}) \right) = \alpha \delta E(\mathscr{S}) \tag{4.2}$$

where α now appears as a relative weight on the surface energy term in (3.3) with respect to the log-likelihood term. Because the gradient descent of the negative logarithm of the prior is the Euler-Lagrange of the energy function $E(\mathscr{S})$, the rest of this paper is concerned with constructing and solving higher-order feature preserving energy functions that model real surfaces better than previous energy functions used in the literature.

Surface area is a commonly used surface prior energy based on the underlying assumption that amongst surfaces that represent a data set equally well, those that have smaller area are simpler than surfaces of larger area, and therefore have a higher chance of occurrence

in the physical world. A gradient descent on surface area gives the mean curvature flow PDE [33]

$$\delta E(\mathscr{S}) = \mathrm{HN} = \left(\frac{\kappa_1 + \kappa_2}{2}\right)\mathrm{N} \tag{4.3}$$

where κ_1, κ_2 are the principal curvatures and H is the mean curvature of the surface $\mathscr{S}$, and N is the surface normal. Despite the simplicity of its solution, surface area is not a realistic shape prior. Problems with this approach will be demonstrated in Section 6. To successfully model the smoothness constraints on realistic surfaces, a general, higher-order and non-linear prior is needed.

A second-order energy function is *total curvature*,

$$\int_S \kappa_1^2 + \kappa_2^2 \, dS \tag{4.4}$$

which has been shown to deform surfaces into spheres when minimized [35]. Chopp & Sethian [29] derived the intrinsic Laplacian of curvature flow which minimizes the total curvature for a level set curve, and solve the resulting fourth-order nonlinear PDE. However, they argue that the numerical methods used to solve fourth-order flows are not practical, because they lack long term stability. They propose several new numerical schemes, but none are found to be completely satisfactory due to their slow computation and inability to handle singularities. Total curvature for level set *surfaces* (4.4), which is minimized by the intrinsic Laplacian of mean curvature flow, is an even harder problem.

A more general energy function, which we set out to solve in this paper, is

$$\int_S G\left(\kappa_1^2 + \kappa_2^2\right) \, dS, \tag{4.5}$$

where G is a general non-linear function of curvature. This general second-order energy is based on functions of total curvature which include (4.4) and the generalization of Nordstrom's *biased anisotropic diffusion* (3.6) to surfaces. Minimizing this general energy form together with the data likelihood term will require solving fourth-order surface PDEs.

Chapter 5

A splitting strategy for higher-order priors

In this section, we introduce a method for solving general fourth-order surface PDEs, such as the intrinsic Laplacian of mean curvature flow, by breaking them into two second-order PDEs using the surface normals. A stable and efficient numerical solution to this flow is also key to the solution of the more general energy form given by (4.5). A two-step solution to intrinsic Laplacian of mean curvature flow for meshes is proposed by [31]. However, that approach works only for meshes, and relies on analytic properties of the steady-state solutions, $\Delta \mathrm{H} = 0$, by fitting surface primitives that have those properties. Thus, the formalism does not generalize to applications, such as surface reconstruction, where the solution is a combination of measured data and the total curvature term and might not satisfy $\Delta \mathrm{H} = 0$.

Another example of a splitting strategy can be found in [36], where the authors penalize the smoothness of a vector field while simultaneously forcing the gradient directions of a gray scale image to closely match the vector field. The penalty function on the normal field is proportional to the divergence of the normal vectors. This forms a high-order interpolation function, which is shown to be useful for image inpainting—recovering missing patches of data in 2D images. The strategy of simultaneously penalizing the divergence of a normal field, and the mismatch of this field with the image gradient, is related to the our splitting strategy. However, our formulation emphasizes the processing of normals on an arbitrary surface manifold (rather than the flat geometry of an image), with an explicit relationship to fourth-order surface flows. Furthermore, this paper establishes new directions for surface flows— toward edge-preserving surface reconstruction. The proposed two-step solution allows the surface shape to lag the normals as they are filtered and then refitted by a separate process. Figure 5.1 shows this three step process graphically in 2D—shapes give rise to

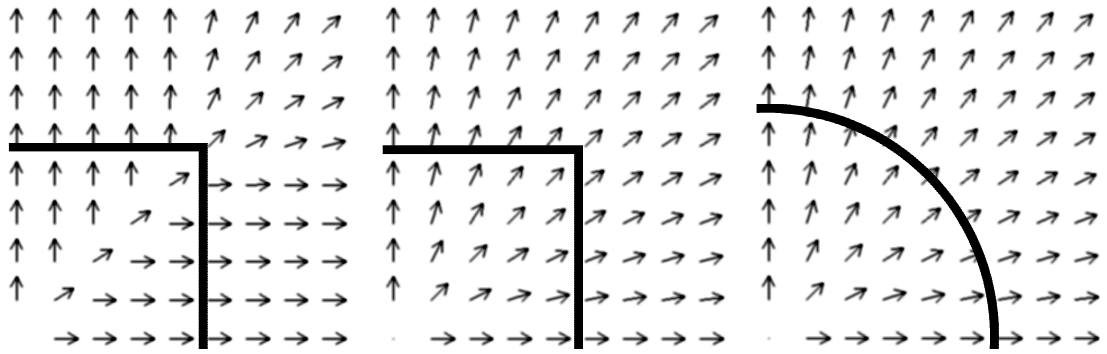


Figure 5.1: Shown here in 2D, the surface flow process begins with a shape and constructs a normal map from the distance transform (left), modifies the normal map according to a PDE derived from a penalty function (center), and re-fits the shape to the normal map (right).

normal maps, which, when filtered, give rise to new shapes.

5.1 Notation

To facilitate the discussion, we use the Einstein notation convention, where the subscripts indicate tensor indexes, and repeated subscripts within a product represent a summation over the index (across the dimensions of the underlying space). Furthermore, we use the convention that subscripts on quantities represent derivatives, except where they are in parenthesis, in which case they refer to a vector-valued variable. Thus, ϕ_i is the gradient vector of a scalar quantity $\phi : \mathbb{R}^n \mapsto \mathbb{R}$. The Hessian is ϕ_{ij}, and the Laplacian is ϕ_{ii}. A vector field is $v_{(i)}$, where $v : \mathbb{R}^n \mapsto \mathbb{R}^n$, and the divergence of that field is $v_{(i)i}$. Scalar operators behave in the usual way. Thus, gradient magnitude is $|\phi_i| = \sqrt{\phi_i \phi_i}$ and the differential for a coordinate system is $dx_{(i)} = dx_1 dx_2 \ldots dx_n$.

Level set surface models rely on the notion of a regular surface, a collection of 3D points with a topology that allows each point to be modeled *locally* as a function of two variables. We can describe the deformation of such a surface using the 3D velocity of each of its constituent points, *i.e.*, $\partial s_{(i)}(t)/\partial t$ for all $s_{(i)} \in \mathscr{S}$. If we represent the surface implicitly at each time t, then

$$\mathscr{S} = \left\{ s_{(i)}(t) \mid \phi\left(s_{(i)}(t), t\right) = 0 \right\}. \tag{5.1}$$

Surfaces defined in this way divide a volume into two parts: inside ($\phi > 0$) and outside ($\phi < 0$). It is common to choose ϕ to be the signed distance transform of $\mathscr{S}$, or an approximation

thereof.

The surface remains a level set of ϕ over time, and thus taking the total derivative with respect to time (using the chain rule) gives

$$\frac{\partial \phi}{\partial t} = -\phi_j \frac{\partial s_{(j)}}{\partial t} \tag{5.2}$$

The gradient ϕ_j is proportional to the surface normal. Thus $\partial s_{(j)}/\partial t$ affects ϕ only in the direction of the surface normal— motion in any other direction is merely a change in the parameterization.

5.2 Curvature from normal maps

When using implicit representations one must account for the fact that derivatives of functions defined on the surface are computed by projecting their 3D derivatives onto the surface tangent plane. Let $N_{(i)} : \mathbb{R}^3 \to S^3$ be the normal map, which is a field of normals that are everywhere perpendicular to the family of embedded isosurfaces of ϕ — thus $N_{(i)} = \phi_i / \sqrt{\phi_k \phi_k}$. The 3×3 projection matrix for the implicit surface normal is $P_{(ij)} = N_{(i)} N_{(j)}$, and $P_{(ij)} V_{(i)}$ returns the projection of $V_{(i)}$ onto $N_{(i)}$. Let $I_{(ij)}$ be the identity matrix. Then the projection onto the plane that is perpendicular to the vector field $N_{(i)}$ is the *tangent projection operator*, $T_{(ij)} = I_{(ij)} - P_{(ij)}$. Under typical circumstances, the normal map $N_{(i)}$ is derived from the gradient of ϕ, and $T_{(ij)}$ projects vectors onto the tangent planes of the level sets of ϕ. However, the computational strategy we propose allows ϕ to lag the normal map. Therefore, the tangent projection operators differ, and we use $T^{\phi}_{(ij)}$ to denote projections onto the tangent planes of the level sets of ϕ and $T^{N}_{(ij)}$ to denote projections onto planes perpendicular to the normal map.

The shape matrix [37] of a surface describes its curvature independent of the parameterization. For an implicit surface, it is obtained by differentiating the normal map and projecting the derivative onto the surface tangent plane. The norm of the shape matrix is the sum of squared principal curvatures

$$\kappa^2 = \left|\left| N_{(i)j} T^{\phi}_{(jk)} \right|\right|^2. \tag{5.3}$$

If $N_{(i)}$ is derived directly from ϕ, this gives

$$\kappa^2 = \left|\left| T^{\phi}_{(il)} \phi_{lj} T^{\phi}_{(jk)} \right|\right|^2. \tag{5.4}$$

We now express the general surface energy function introduced in (4.5) in terms of the normals of the surface

$$\mathscr{G}\left(\kappa^2\right) = \int_{\mathscr{S}} G(\kappa^2) dx_{(i)}. \tag{5.5}$$

If we take the first variation of (5.5) with respect to ϕ using (5.4) for total curvature, we obtain a fourth-order PDE on ϕ. On the other hand, if we use (5.3) for total curvature and take the first variation of (5.5) with respect to $N_{(i)}$, allowing ϕ to remain fixed, we obtain a second-order PDE on $N_{(i)}$. This paper uses the latter strategy.

As we process the normal map $N_{(i)}$, letting ϕ lag, we must ensure that it maintains the unit length constraint, $N_{(i)}N_{(i)} = 1$. This is expressed in the penalty function using Lagrange multipliers. The constrained penalty function is

$$\mathscr{G}(\kappa^2) + \int_{\mathscr{S}} \lambda\left(x_{(l)}\right)\left(N_{(k)}N_{(k)} - 1\right) dx_{(l)}, \tag{5.6}$$

where $\lambda\left(x_{(l)}\right)$ is the Lagrange multiplier at $x_{(l)}$. The next step is to derive the first variation. The curvature (5.3) can be written as

$$\kappa^2 = \left|\left|N_{(i)j}\right|\right|^2 - \frac{\left|\left|N_{(i)j}\phi_j\right|\right|^2}{\phi_k\phi_k}. \tag{5.7}$$

Using (5.7) and solving for λ in (5.6) introduces a projection operator $T^N_{(ij)}$ on the first variation of $\mathscr{G}$, which keeps $N_{(i)}$ unit length. The the first variation of (5.6) with respect to $N_{(i)}$ is

$$T^N_{(ij)}\frac{d\mathscr{G}}{dN_{(j)}} = 2\ T^N_{(ij)}\left[g\left(\kappa^2\right)\left(N_{(j)k} - \frac{N_{(j)l}\phi_l\phi_k}{\phi_m\phi_m}\right)\right]_k, \tag{5.8}$$

where g is the derivative of G with respect to κ^2. A gradient descent on this metric $\partial N_{(i)}/\partial t = -T^N_{(ij)}d\mathscr{G}/dN_{(j)}$, results in a PDE that minimizes (5.5). The process of projecting derivatives of N onto the level sets of ϕ is the same as solving the constrained diffusion equation on $N_{(i)}$, using the method of solving PDEs on implicit manifolds described by [38]. We will discuss several choices for G in Section 6.

5.3 Surface refitting

We have shown how to evolve the normals to minimize functions of curvature; however, the final goal is to process the surface, which requires deforming ϕ. Therefore, the next step is

to relate the deformation of the level sets of ϕ to the evolution of $N_{(i)}$. Suppose that we are given the normal map $N_{(i)}$ to some set of surfaces, but not necessarily level sets of ϕ—as is the case if we filter $N_{(i)}$ and let ϕ lag. We can manipulate ϕ so that it fits the normal field $N_{(i)}$ by minimizing a penalty function that quantifies the discrepancy. This penalty function is

$$\mathscr{D}(\phi) = \int_U \left[\sqrt{\phi_i \phi_i} - \phi_i N_{(i)} \right] dx_{(j)}, \tag{5.9}$$

where $U \subset \mathbb{R}^3$ is the domain of ϕ. The first variation of this penalty function with respect to ϕ is

$$\frac{d\mathscr{D}}{d\phi} = -||\phi_k|| \left[\left(\frac{\phi_j}{\phi_m \phi_m} \right)_j - N_{(j)j} \right] = -||\nabla\phi|| \left[\mathrm{H}^\phi - \mathrm{H}^\mathrm{N} \right] \tag{5.10}$$

where H^ϕ is the mean curvature of the level set surface and H^N is half the divergence of the normal map. Then, the gradient descent PDE that minimizes (5.9) is $d\phi/dt = -d\mathscr{D}/d\phi$. Thus according to (5.10), the surface moves as the difference between its own curvature and that of the normal field. The factor of $\| \nabla\phi \|$, which is typical with level set formulations, comes from the fact that we are manipulating the shape of the level set, which is embedded in ϕ, as in (5.2).

The gradient descent for the MAP surface estimator (3.3) is a weighted sum of the refitting term, which we have derived in (5.10), and a data term. Therefore, the final update rule for ϕ is

$$\frac{\partial\phi}{\partial t} = -f(\mathscr{S})N + \alpha||\nabla\phi|| \left[\mathrm{H}^\phi - \mathrm{H}^\mathrm{N} \right], \tag{5.11}$$

where α is a free parameter that determines the relative weight of the prior term with respect to the data term. We will refer to α as the prior weight.

We can now put together the normal processing stage and the surface refitting stage to solve fourth-order level set flows. The entire process is depicted in Figure 5.2. We have derived a gradient descent for the normal map that minimize the surface energy functions proposed in Section 5.2. This process is denoted in Figure 5.2 as the dN/dt loop. This loop iterates dN/dt for a fixed number [20] of applications so that the difference between ϕ and the surface defined by the normals remain small. The surface fitting to the the combined normal map and data terms is formulated as a gradient descent in (5.11). This process is the $d\phi/dt$ loop in Figure 5.2. The overall algorithm shown in Figure 5.2 repeats these two steps to minimize the penalty functions in terms of the surface. This main loop iterates until the RMS change in ϕ as a result of the application of $d\phi/dt$ becomes small (less than 10^{-6} in our implementation) which signals convergence. Details of the numerical implementation will be discussed in Appendix A. In Appendix B, we will show that the overall process of simultaneously solving these PDEs as described is equivalent to solving the original fourth-order PDE.

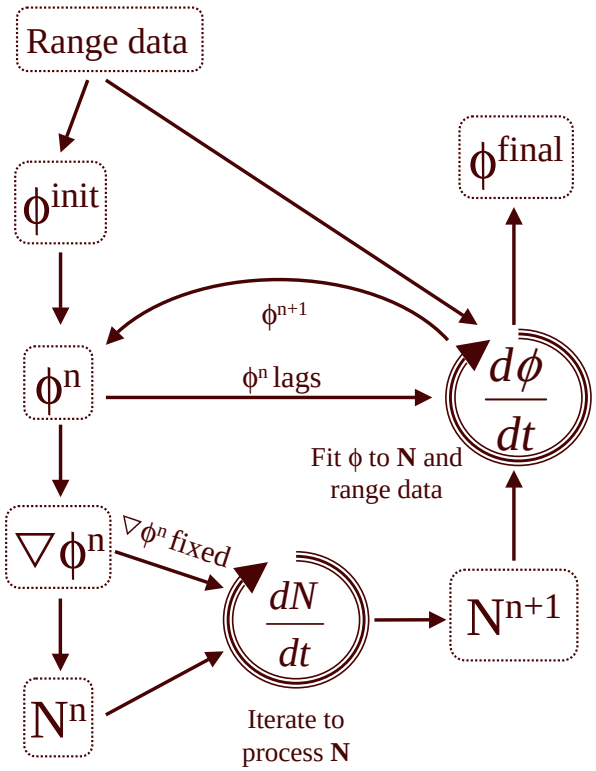


Figure 5.2: Flow chart

Chapter 6

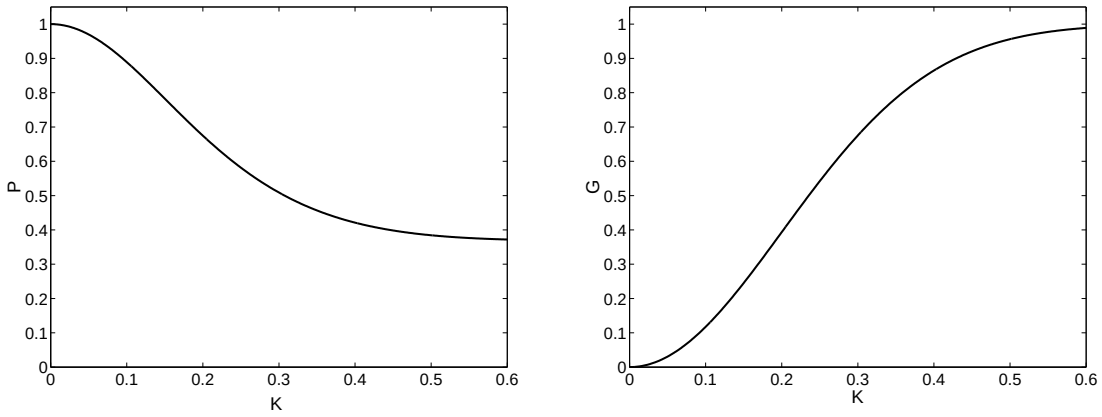
Experiments

In Section 5, we developed a framework that allows us to use priors that are based on high-order, general functions of curvature. We now explore two such functions and compare them to the surface area prior. Total curvature of a surface as defined in (4.4) can be obtained in our formulation by choosing G to be the identity function, *i.e.*, $G(\kappa^2) = \kappa^2$. The derivative of G with respect to κ^2 is $g(\kappa^2) = 1$. We refer to this choice of G as the *isotropic curvature prior* because it corresponds to solving the heat equation on the normal map with a constant, scalar conduction coefficient. Minimizing the total curvature of a surface works well for smoothing surfaces and eliminating noise, but, like Gaussian blurring, it also deforms or removes important features.

The generalization of P&M anisotropic diffusion to surfaces is achieved from variational principles by choosing the appropriate function of the squared curvature in (5.5). For instance,

$$G(\kappa^2) = 2\mu^2 \left(1 - e^{-\frac{\kappa^2}{2\mu^2}}\right), \text{ and } g(\kappa^2) = e^{-\frac{\kappa^2}{2\mu^2}}. \tag{6.1}$$

We name this choice of G, which is graphed in Figure 6.1(b), the *anisotropic curvature prior*. The minimization of the penalty function in (5.8) with the anisotropic curvature prior gives a vector-valued anisotropic diffusion on the level set surface—a generalization of P&M diffusion to surfaces. This flow is a modified version of intrinsic Laplacian of mean curvature flow that preserves or enhances areas of high curvature, which we call *creases*. Creases are the generalization of edges in images to surfaces. The preservation of creases is achieved by the flat tails of G, which limits the penalty on high curvature areas. Note that μ in (6.1) was fixed at 0.2 for all the experiments in this paper. Unlike, in P&M image diffusion, this parameter does not need to be changed for different surface reconstructions. In the context of P&M image diffusion, the units of μ are in gray levels; consequently, the



(a) (b)

Figure 6.1: (a) A heavy tailed Gaussian probability distribution function. This function can be scaled to integrate to 1 because in our discrete implementation κ has a finite upper bound. (b) Graph of $G = 2\mu^2 \left(1 - e^{-\frac{\kappa^2}{2\mu^2}}\right)$ vs. κ with $\mu = 0.2$. G is also the logarithm of (a).

optimal choice of μ is image dependent. However, in surface reconstruction, the units are in curvature, see (6.1), which is data independent. This makes it possible to choose a μ value that gives consistent results over a broad range of surfaces.

The anisotropic curvature prior can also be analyzed from a statistical point of view. Typically, robust estimators use heavy tailed Gaussian probability distributions such as the one illustrated in Figure 6.1(a). Such probability distribution functions assign non-zero probabilities to outliers, in our case large κ values. The logarithm of the heavy tailed Gaussian has the same form as the anisotropic G we have chosen. In fact, using the Gibbs distribution (4.1), we can see that the prior probability distribution achieved by $G(\cdot)$ is proportional to $P = e^{-G(\cdot)}$. The parameter μ in (6.1) controls the κ values for which G and P flatten out.

Section 6.1 will present a discussion of the quantitative differences between the isotropic and anisotropic curvature priors and the surface area prior using synthetic data as ground truth. We will also investigate the effects of the prior weight on the results. Section 6.2 will present surface reconstruction examples from real data measured by a laser range finder.

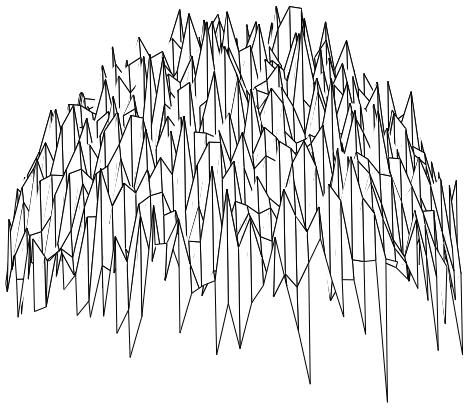


Figure 6.2: A noisy range image of a sphere plotted as a depth map.

6.1 Experiments with synthetic data

The experiments presented in this section use geometric shapes for which we can construct analytical distance transforms. We use the following experiment setup:

1. Build range images from the analytical distance transform using the model for the laser range finder located at several positions,
2. Add independent Gaussian noise to the range images to simulate measurement noise,
3. Reconstruct a surface model from the noisy range images, and
4. Compare the resulting surface model to the analytical shape by computing the root mean square (rms) geometric distance between the two surfaces.

The first shape we examine is a sphere with radius 1 unit. All other distances are relative to this measurement unit. For this experiment we simulate six range finders located at a distance of 3.5 units from the center of the sphere along the six cardinal directions. Independent Gaussian noise with a standard deviation that is 10% of the sphere's radius (0.1 units), is added to each range image. One of the noisy range images is shown in Figure 6.2 as a depth map. We reconstruct surface models from these noisy range images using the three priors under investigation with a range of weights, α. For each choice, we run the algorithm described in Section 5 until it converges. We decide that the algorithm has converged when the rms change in ϕ for any $d\phi/dt$ iteration fell below the threshold 1.0×10^{-6}. Then, we calculate the rms distance between the original model and the reconstructed model; denote this distance by $\mathscr{E}$.

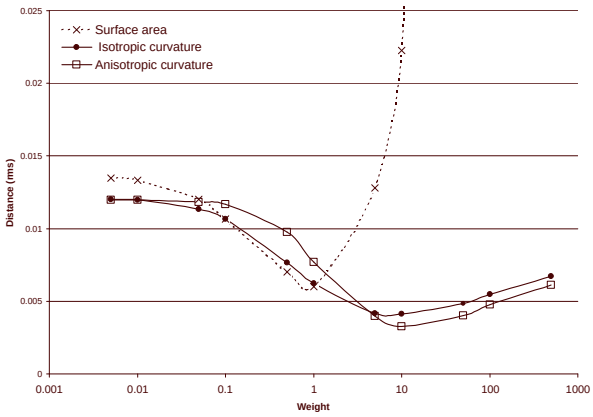
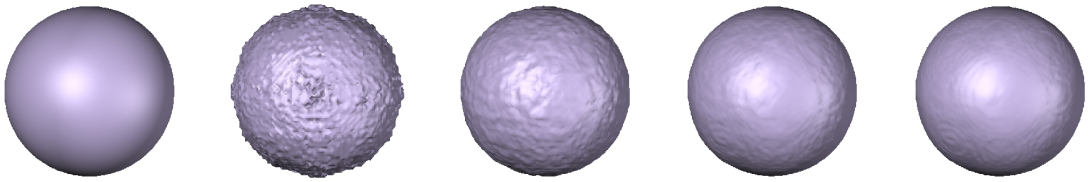


Figure 6.3: Rms geometric distance between the analytical sphere and the reconstructed surface.

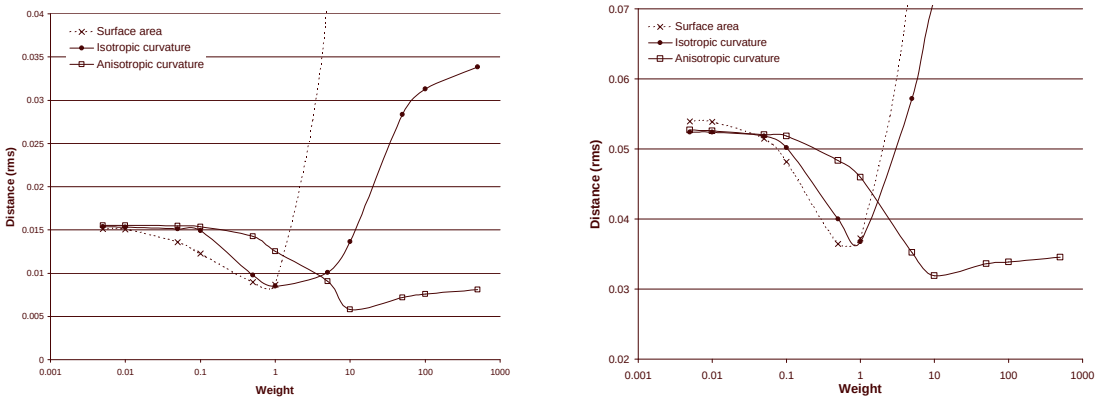
Figure 6.3 plots $\mathscr{E}$ against the $\log\alpha$ for the different priors under examination. The units on the *y-axis* are the same as the units used to described the size of the shape. It can be observed from Figure 6.3 that the limiting value for $\mathscr{E}$ as $\alpha \to 0$ is approximately 0.0125. This limit is the error obtained if surface reconstruction is performed without a prior (relying only on the data). This error level is smaller than the noise added to the range images because of the averaging effect of using multiple range images. The anisotropic and the isotropic curvature priors at their optimal weight provide a 75% reduction on this error. On the other hand, surface area provides slightly better than a 50% reduction at its optimal weight. These best reconstructions are illustrated in Figure 6.4. The shapes of the error plots is more important than the results at optimal choices of weight. The plot for the surface area prior dips down sharply around $\alpha = 1$ which indicates a small range of useful weights for this prior. The surface area prior performs especially poorly as α is increased beyond 1; this is due to the fact that the surface area prior causes shrinkage in the surface models. In practice, this will mean difficulties for the user in choosing a weight that works different reconstruction scenarios. In contrast, both of the higher-order priors have relatively flat error plots because their limiting behaviors match the model in this example. Laplacian of mean curvature flow has been shown to deform surfaces to spheres [35]. Isotropic curvature reconstruction is as good as the anisotropic curvature reconstruction because the sphere does not contain creases.

To examine the differences between the priors further, we experiment with a cube and another piecewise planar shape which we will name “3d-cross”. In these experiments, we used 8 range finder locations (one in each octant). Figure 6.6 (a) and (b) show the original cube with sides 1 unit long, and the surface initialization from the noisy range images, respectively. Independent Gaussian noise with standard deviation 0.1 was added to the simulated data to create the noisy range images. The results (see Figure 6.6) with the surface



(a) (b) (c) (d) (e)

Figure 6.4: (a) Analytical sphere, (b) surface model initialization from noisy data. Resulting surface model for the (c) surface area prior with $\alpha = 1$, (d) the isotropic curvature prior with $\alpha = 5$, and (e) the anisotropic curvature prior with $\alpha = 10$.



(a) (b)

Figure 6.5: Rms geometric distance between the reconstructed surface and the analytical models for (a) the cube, and (b) the "3d-cross".

area and isotropic curvature priors are noisier and with rounded corners compared to the successfully denoised, approximately piecewise planar results obtained with the anisotropic curvature reconstruction. The last example, shown in Figure 6.7(a), amplifies the differences observed with the cube experiment. It is not possible to denoise the surface without causing severe distortion to the shape with the other priors. Anisotropic curvature reconstruction does not suffer from this trade-off between smoothing and distortion. It provides a very good (denoised and relatively undistorted) reconstruction across a range of values for the prior weight (see Figure 6.7(g) and (h)). The error plots for the cube and the "3d-cross" shown in Figure 6.5 confirm that the anisotropic curvature reconstruction error flattens out as α increases beyond its optimal value.

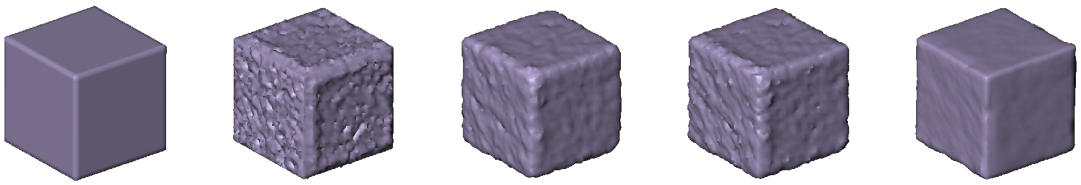


Figure 6.6: (a) Analytical cube, (b) surface model initialization from noisy data. Resulting surface model for (c) the surface area prior with $\alpha = 1$, (d) the isotropic curvature prior with $\alpha = 1$, and (e) the anisotropic curvature prior with $\alpha = 10$.

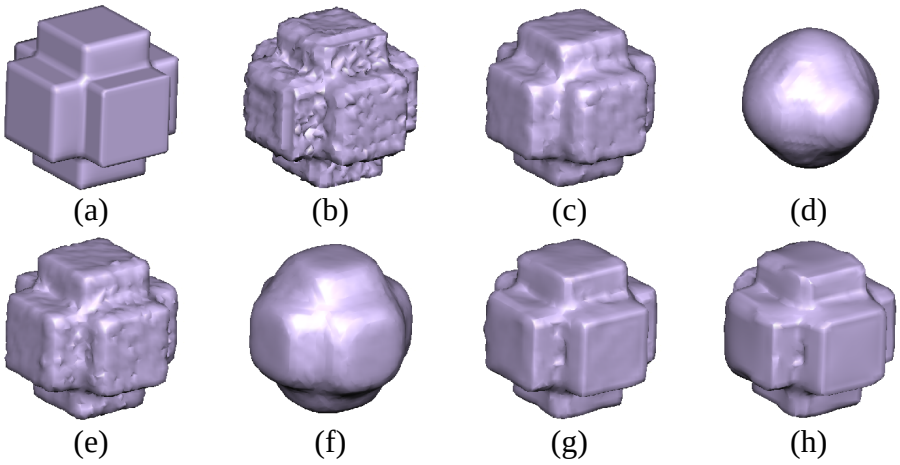
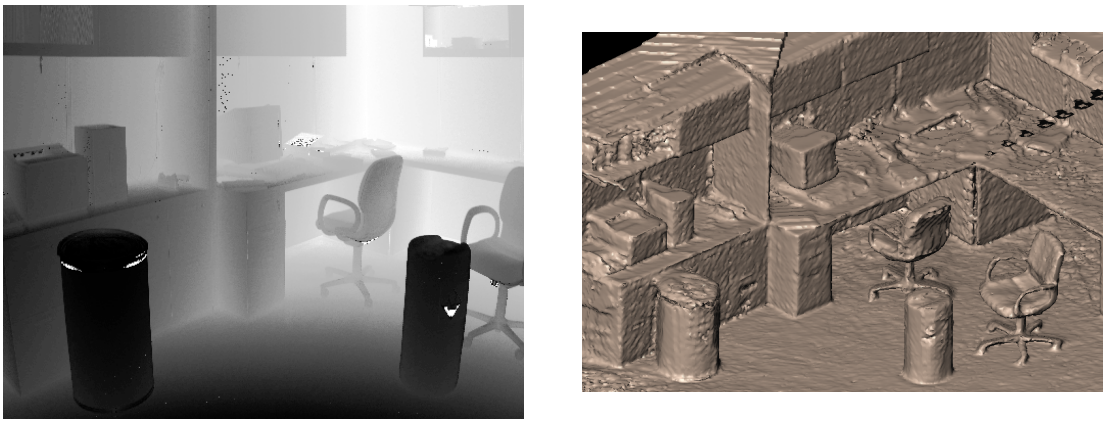


Figure 6.7: (a) Analytical "3d-cross", (b) surface model initialization from noisy data. Resulting surface model for the surface area prior with (c) $alpha = 1$ and (d) $\alpha = 10$. For the isotropic curvature prior with (e) $\alpha = 1$ and (f) $\alpha = 10$. For the anisotropic curvature prior with (g) $\alpha = 10$ and (h) $\alpha = 100$.



(a) (b)

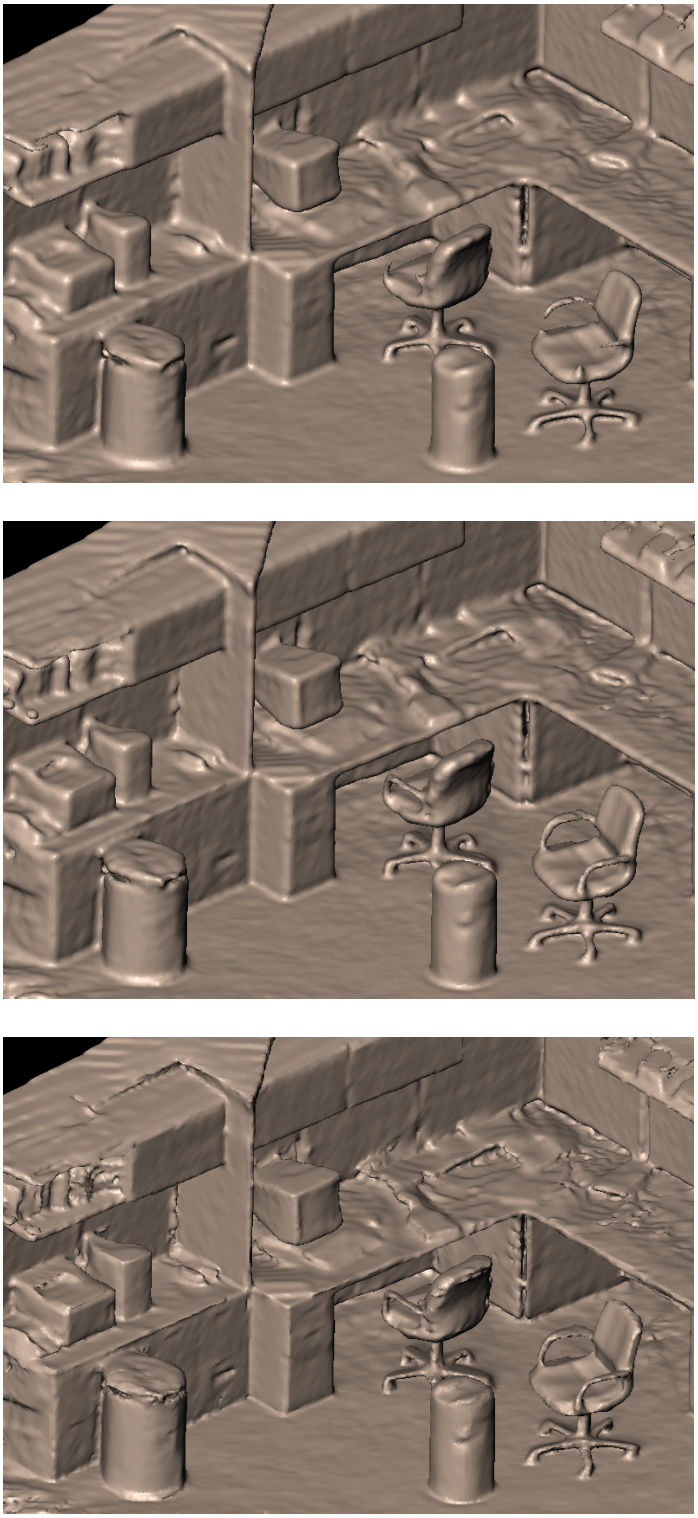
Figure 6.8: (a) One of the range images used in the surface reconstruction experiment, (b) surface initialization.

6.2 Surface reconstruction from measured range data

In this section, we will investigate these surface priors in the context of the reconstruction of an office scene shown in Figure 6.8. This example involves 12 range scans of a room which were registered using the methods described in [18]. One of these 12 range images is shown in Figure 6.8(a). The model initialization is shown in Figure 6.8(b). In addition to measurement noise, this initialization contains artifacts such as the multiple disconnected pieces in the upper right corner. Figure 6.9 illustrates the results obtained with the surface area, isotropic and anisotropic curvature priors. As in Section 6.1, we let the algorithms run until convergence. The prior term weights were chosen after experimenting with different values and picking those that produced the qualitatively best results. The results with the anisotropic curvature prior remains true to the actual surface while eliminating noise. On the other hand, the results with the surface area prior that produce a comparable amount of denoising demonstrate shape distortion such as the breaking of the arms of the chairs and the rounding of the creases on the desk and computer equipment. The isotropic curvature prior does not cause any severe shape distortions, but it smoothes edges, creases and corners as expected. This experiment illustrates the importance of the anisotropic curvature prior in reconstructions involving scenes with high curvature features and sharp creases.

Figure 6.10 demonstrates the robustness of the different priors by a close up examination of a portion of the scene. Figure 6.10 (a) and (b) are the initialization and the visible imagery for one of the chairs in the scene. Figure 6.10 (c), (e) and (g) illustrate the results

obtained by qualitatively choosing good values for α. Figure 6.10(d), (f) and (h) illustrate the results if α is chosen to be 10 times this value. These results show that the anisotropic curvature is least sensitive to the choice of the prior weight. Another well known problem with mean curvature flow and in our case the surface area prior can easily be observed in Figure 6.10(d); the beam connecting the base to the seat is being pinched-off. Finally, Figure 6.11 illustrates the reconstruction of a tank with the anisotropic curvature prior. This reconstruction used 12 range images.



(a)

(b)

(c)

Figure 6.9: Results with (a) surface area, (b) isotropic curvature and (c) anisotropic curvature priors.

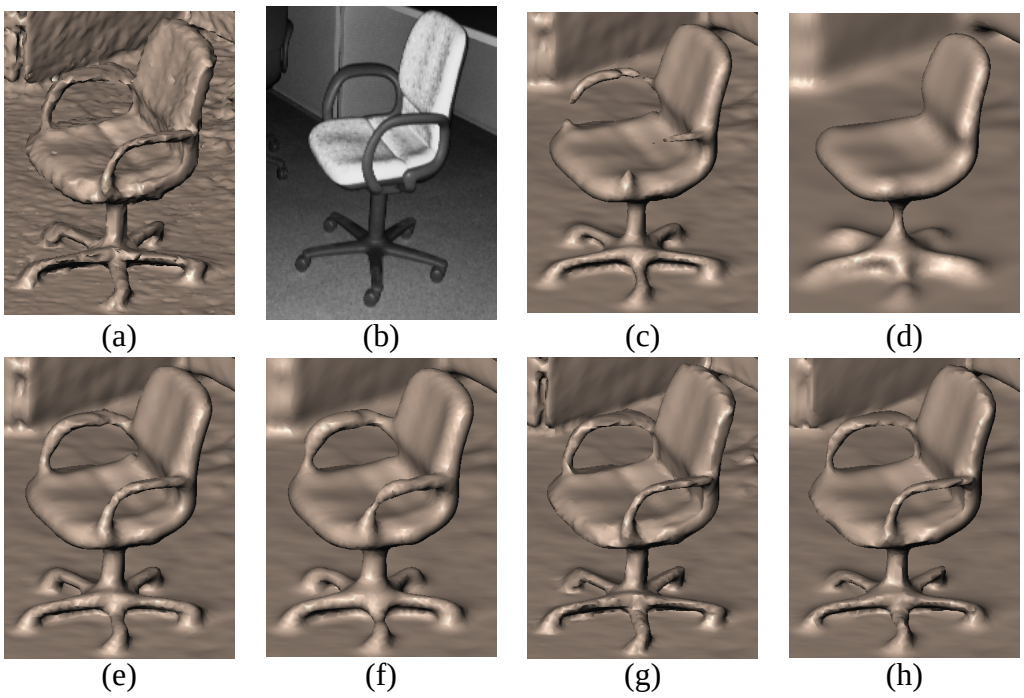


Figure 6.10: (a) Initial surface model for chair, (b) a visible light image of the chair taken from a similar point of view. Results for the surface area prior with weights (c) 1, and (d) 10. Results for the isotropic curvature prior with weights (e) 1, and (f) 10. Results for the anisotropic curvature prior with weights (g) 1, and (h) 10.

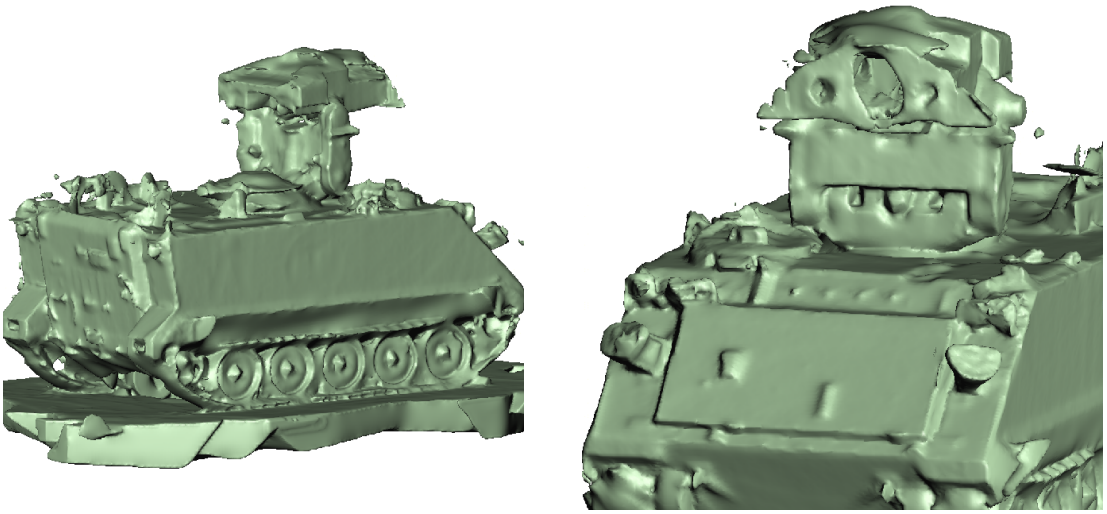


Figure 6.11: Reconstruction of a tank (with the anisotropic curvature prior) from range data.

Chapter 7

Conclusion

Anisotropic diffusion and image reconstruction techniques based on robust metrics have been shown to be very useful tools in image processing. We generalize these methods to surface reconstruction via minimizing a general family of penalty functions of curvature. Isotropic and anisotropic total curvature of a surface are members of this family. An anisotropic curvature penalty leads to a surface reconstruction prior that preserves creases while denoising the input. This process is shown to have important quantitative and qualitative advantages over using a surface area prior. Because, the data term can be chosen independently from the prior, the ideas introduced in this paper can be applied to other forms of surface reconstruction such as applications in tomography.

The minimization of these second-order penalty functions require solving fourth-order PDEs on level sets. Our approach is based on the proposition that the *natural* generalization of image processing to surfaces occurs via the normals. Variational processes on the surface have corresponding variational formulations on the surface normals. This formulation leads to a strict generalization of P&M anisotropic diffusion to surfaces. Normals are processed separately from the surface. We use a metric on the surface manifold, rather than a simple, flat metric. By processing the normals separately from the surface, we can solve a pair of coupled second-order equations instead of a fourth-order equation. Typically, we allow one equation (the surface) to lag the other. This method is numerically more stable and computationally less expensive than solving the fourth-order PDE directly. We solve these equations using implicit surfaces, representing the implicit function on a discrete grid, modeling the deformation with the method of level sets. Because of the implementation, the method applies equally well to surfaces that can be represented in a volume.

The main shortcoming of this method is the computation time, which is significant. How-

ever, the current process lends itself to parallelism. The advent of cheap, specialized, stream-processing hardware promises significantly faster implementations that take advantage of the the inherent parallelism in the process [39]. Another possibility is to use multi-threading or clusters of computers. Regarding the algorithm itself, the use of adaptive or multi-resolution level set strategies could also improve the processing time.

Appendix A

Numerical Implementation

By embedding surface models in volumes, we convert equations that describe the movement of surface points to nonlinear PDEs defined on a volume. The next step is to discretize these PDEs in space and time. The embedding function ϕ is defined on the volume domain U and time. The PDEs are solved using a discrete sampling with forward differences along the time axis. There are two issues with discretization: (i) the accuracy and stability of the numerical solutions, (ii) the increase in computational complexity introduced by the dimensionality of the domain.

For brevity, we will discuss the numerical implementation in 2D— the extension to 3D is straightforward. The function $\phi : U \mapsto \mathbb{R}$ has a discrete sampling $\phi[p,q]$, where $[p,q]$ is a grid location and $\phi[p,q] = \phi(x_p, y_q)$. We will refer to a specific time instance of this function with superscripts, *i.e.* $\phi^n[p,q] = \phi(x_p, y_q, t_n)$. For a vector in 2-space v, we use $v_{(x)}$ and $v_{(y)}$ to refer to its components consistent with the notation of Section 4. In our calculations, we need three different approximations to first-order derivatives: forward, backward and central differences. We denote the type of discrete difference using superscripts on a difference operator, *i.e.*, $\delta^{(+)}$ for forward differences, $\delta^{(-)}$ for backward differences, and δ for central differences. For instance, the differences in the x-direction on a discrete grid with unit spacing are

$$\begin{aligned} \delta_x^{(+)}\phi[p,q] &\triangleq \phi[p+1,q] - \phi[p,q], \\ \delta_x^{(-)}\phi[p,q] &\triangleq \phi[p,q] - \phi[p-1,q], \text{ and} \\ \delta_x\phi[p,q] &\triangleq \frac{\phi[p+1,q] - \phi[p-1,q]}{2}, \end{aligned} \tag{A.1}$$

where the time superscript has been left off for conciseness. The application of these dif-

ference operators to vector-valued functions denotes componentwise differentiation.

The positions of derivatives computed with forward and backward differences are staggered off the grid by $1/2$ pixels. For instance, $\delta_x^{(+)}\phi[p,q]$ as defined in (A.1) uses information from positions $[p+1,q]$ and $[p,q]$ with equal weights; hence, it exists at $[p+1/2,q]$. This is staggered by $1/2$ pixels in the x direction from the grid. To keep track of staggered locations, we will use the following notation: α, $\overset{x+}{\alpha}$, and $\overset{y+}{\alpha}$ will denote the variable α computed at $[p,q]$, $[p+1/2,q]$, and $[p,q+1/2]$, respectively.

The initialization ϕ_{init} is obtained by methods discussed in [17]. In describing the numerical implementation, we will follow the flow chart in Figure 5.2 for the n'th iteration of the main loop. First, compute the normal vectors from ϕ^n as unit vectors in the direction of the gradient of ϕ^n. The gradient of ϕ^n is computed with central differences as

$$\phi_i^n[p,q] \approx \begin{pmatrix} \delta_x\phi^n[p,q] \\ \delta_y\phi^n[p,q] \end{pmatrix}, \tag{A.2}$$

and the normal vectors are initialized as

$$N_{(i)}^{u=0}[p,q] = \phi_i^n[p,q] / \parallel \phi_i^n[p,q] \parallel . \tag{A.3}$$

Because ϕ^n is fixed and allowed to lag behind the evolution of $N_{(i)}$, the time steps in the evolution of $N_{(i)}$ are denoted with a different superscript, u. For this evolution, $\partial N_{(i)}/\partial t = -d\mathscr{G}/dN_{(i)}$, given by (5.8), is implemented with smallest support area operators. The Laplacian of a function can be applied in two steps, first the gradient and then the divergence. In 2D, the gradient of the normals produces a 2×2 matrix, and the divergence operator in (5.8) collapses this to a 2×1 vector. The diffusion of the normal vectors in the tangent plane of the level sets of ϕ^n, requires us to compute the flux in the x and y directions. The "columns" of the flux matrix are independently computed as

$$\overset{x+}{M}{}_{(i)}^{u} \approx \delta_x^{(+)} N_{(i)}^u - \overset{x+}{C}{}_{(i)}^{u} \left(\delta_x^{(+)}\phi^n\right) , \quad \overset{y+}{M}{}_{(i)}^{u} \approx \delta_y^{(+)} N_{(i)}^u - \overset{y+}{C}{}_{(i)}^{u} \left(\delta_y^{(+)}\phi^n\right) \tag{A.4}$$

where the time index n remains fixed as we increment u, and

$$\overset{x+}{C}{}_{(i)}^{u} = \overset{x+}{N}{}_{(i)j}^{u} \frac{\overset{x+}{\phi}{}_j^n}{\overset{x+}{\phi}{}_k^n \overset{x+}{\phi}{}_k^n} , \quad \overset{y+}{C}{}_{(i)}^{u} = \overset{y+}{N}{}_{(i)j}^{u} \frac{\overset{y+}{\phi}{}_j^n}{\overset{y+}{\phi}{}_k^n \overset{y+}{\phi}{}_k^n} . \tag{A.5}$$

Staggered derivatives are computed with forward differences. They are located on a grid that is offset from the grid where ϕ and $N_{(i)}$ are defined, as shown Figure A.1 for the 2D case. Furthermore, because the offset is a half pixel only in the direction of the differentiation, the locations of $\delta_x^{(+)}N_{(i)}$ and $\delta_y^{(+)}N_{(i)}$ are different, but are the same locations as the

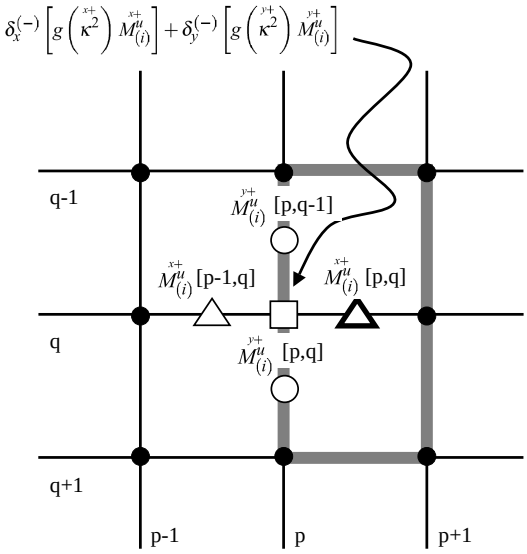


Figure A.1: Computation grid

flux in (A.4). To evaluate (A.5), derivatives of ϕ and $N_{(i)}$ must computed at $[p+1/2,q]$ and $[p,q+1/2]$. These computations are done with the smallest support area operators, using the symmetric 2×3 grid of samples around each staggered point, as shown in Figure A.1 with the heavy rectangle. For instance, the gradient of ϕ staggered in the *x*-direction is

$$\overset{x+}{\phi}{}^n_j = \phi^n_j[p+\frac{1}{2},q] \approx \begin{pmatrix} \delta_x^{(+)}\phi[p,q] \\ \frac{1}{2}\left(\delta_y\phi[p,q]+\delta_y\phi[p+1,q]\right) \end{pmatrix}. \tag{A.6}$$

The staggered gradients of the normals are computed in the same way as (A.6).

Backward differences of the flux are used to compute the divergence operation in (5.8)

$$\left[-\frac{d\mathscr{G}}{dN_{(i)}}\right]^u \approx \delta_x^{(-)}\,\overset{x+}{M}{}^u_{(i)} + \delta_y^{(-)}\,\overset{y+}{M}{}^u_{(i)}\,. \tag{A.7}$$

To evaluate (5.8), we also need to compute $g(\kappa^2)$ at the precise locations where the flux (A.4) are located. For this, we need the squared intrinsic curvature

$$\overset{x+}{\kappa}{}^2 = \overset{x+}{N}{}^u_{(i)j}\overset{x+}{N}{}^u_{(i)j} - \overset{x+}{C}{}^u_{(i)}\overset{x+}{C}{}^u_{(i)} \quad , \quad \overset{y+}{\kappa}{}^2 = \overset{y+}{N}{}^u_{(i)j}\overset{y+}{N}{}^u_{(i)j} - \overset{y+}{C}{}^u_{(i)}\overset{y+}{C}{}^u_{(i)} \tag{A.8}$$

at the required locations. Then (5.8) for anisotropic diffusion is computed as

$$\left[-\frac{d\mathscr{G}}{dN_{(i)}}\right]^u \approx \delta_x^{(-)}\left[g\left(\overset{x+}{\kappa^2}\right)\overset{x+}{M}{}^u_{(i)}\right] + \delta_y^{(-)}\left[g\left(\overset{y+}{\kappa^2}\right)\overset{y+}{M}{}^u_{(i)}\right] \tag{A.9}$$

Notice that these backward differences are defined at the original ϕ grid location $[p,q]$ because they undo the forward staggering in the flux locations. Thus, both components of $d\mathscr{G}/dN_{(i)}$ are located on the original grid for ϕ. Using the tangential projection operator in (5.8), the new set of normal vectors are computed as

$$N_{(i)}^{u+1} = N_{(i)}^{u} + T^{N}\left[\frac{d\mathscr{G}}{dN_{(i)}}\right]^{u} = N_{(i)}^{u} + \left[\frac{d\mathscr{G}}{dN_{(i)}}\right]^{u} - \left(\left[\frac{d\mathscr{G}}{dN_{(j)}}\right]^{u} N_{(j)}^{u}\right) N_{(i)}^{u}. \tag{A.10}$$

Starting with the initialization in (A.3) for $u = 0$, we iterate (A.10) for a fixed number of steps, 25 iterations for the examples in this paper. In other words, we do not aim at minimizing the energy given in (5.6) in the $d\mathscr{G}/dN$ loop of Figure 5.2; we only reduce it.

Once the evolution of N is concluded, ϕ is refitted to the modified normal vectors and the range data term according to (5.11). Using the main loop index, we denote the evolved normals by $N_{(i)}^{n+1}$(see Figure 5.2. To solve (5.11) we calculate H^{ϕ} and $H^{N^{n+1}}$. The latter is the induced mean curvature of the normal map, in other words, it is the curvature of the hypothetical target surface that fits the normal map. Calculation of curvature from a field of normals is given by

$$H^{N^{n+1}} \approx \delta_x N_x^{n+1} + \delta_y N_y^{n+1}, \tag{A.11}$$

where we have used central differences on the components of the normal vectors. This needs to be computed once at initialization as the normal vectors remain fixed during the refitting phase. Let v be the time step index in the $d\mathscr{D}/d\phi$ loop. H^{ϕ^v} is the mean curvature of the moving level set surface at time step v and is calculated from ϕ^v with the smallest area of support

$$H^{\phi^v} \approx \delta_x^{(-)} \frac{\delta_x^{(+)}\phi^v}{\phi_j^{x+}\phi_j^{x+}} + \delta_y^{(-)} \frac{\delta_y^{(+)}\phi^v}{\phi_j^{y+}\phi_j^{y+}}, \tag{A.12}$$

where the gradients in the denominators are staggered to match the locations of the forward differences in the numerator. The staggered gradients of ϕ in the denominator are calculated using the 2×3 neighborhood as in (A.6).

The PDE in (5.11) is solved with a finite forward differences, but with the upwind scheme for the gradient magnitude, to avoid overshooting and maintain stability [40]. The upwind method computes a one-sided derivative that looks in the upwind direction of the moving wave front, and thereby avoids overshooting. Moreover, because we are interested in only a single level set of ϕ, solving (5.11) over all of U is not necessary. Different level sets evolve independently, and we can compute the evolution of ϕ only in a narrow band around the level set of interest and re-initialize this band as necessary [41, 42]. See [43] for more details on numerical schemes and efficient solutions for level set methods. A discussion of

the solution to the data term in (5.11) is beyond the scope of this paper and can be found in [17, 18].

Using the upwind scheme and narrow band methods, ϕ^{v+1} is computed from ϕ^{v} according to (5.10) using the curvatures computed in (A.11) and (A.12). This loop is iterated until the energy in (5.9) ceases to decrease; let v^{final} denote the final iteration of this loop. Then we set ϕ for the next iteration of the main loop (see Figure 5.2) as $\phi^{n+1} = \phi^{v^{final}}$ and repeat the entire procedure until convergence. Convergence properties of the proposed methods were discussed in Section 6.

Appendix B

Intrinsic Laplacian of Mean Curvature Flow

In this section, we will show that the overall process of simultaneously solving the two PDEs (5.8) with $G(\kappa^2) = \kappa^2$ and (5.10) is equivalent to the fourth-order intrinsic Laplacian of mean curvature flow [29] on the original surface. Using this result, we can state that solving (5.8) with $G(\kappa^2) = \kappa^2$ and (5.11), which is a weighted sum of (5.10) and a data term, in the same manner is equivalent to using total curvature (4.4) as the prior in surface reconstruction. This establishes the mathematical foundation of the proposed method.

Lets analyze one iteration of the main loop in Fig. 5.2. Before processing the normals, they are derived from ϕ^n, and we have $N^n_{(i)} = \phi^n_i / \sqrt{\phi^n_j \phi^n_j}$. Evolving the normal map once according to (5.8) for a small amount of time dt gives

$$N^{n+1}_{(i)} = N^n_{(i)} - T^N_{(ij)} \frac{d\mathscr{G}}{dN_{(j)}} dt. \tag{B.1}$$

If we immediately apply (5.10) to fit ϕ to this new normal map

$$\frac{\partial \phi}{\partial t} = \| \phi^n_k \| \left(\mathrm{H}^{\phi} - \left[\mathrm{N}^{\mathrm{n}}_{(\mathrm{i})} - \mathrm{T}^{\mathrm{N}}_{(\mathrm{ij})} \frac{\mathrm{d}\mathscr{G}}{\mathrm{dN}_{(\mathrm{j})}} \mathrm{dt} \right]_{\mathrm{i}} \right). \tag{B.2}$$

Because $N^n_{(i)}$ is derived directly from ϕ^n, we have $N^n_{(i)i} = \mathrm{H}^{\phi}$, which gives the expression for changes in ϕ in order to make up this infinitesimal lag:

$$\frac{\partial \phi}{\partial t} = - \| \phi^n_k \| \left[T^N_{(ij)} \frac{d\mathscr{G}}{dN_{(j)}} \right]_i. \tag{B.3}$$

We can express the penalty function $\mathcal{G}$ in terms of either $N_{(i)}$ or ϕ, and take the first variation with respect to either of these quantities. The relationship between $d\mathcal{G}/dN_{(i)}$ and $d\mathcal{G}/d\phi$ is established by integrating (by parts)—it is

$$\frac{d\mathcal{G}}{d\phi} = -\left[T_{ij}^{\phi} \frac{d\mathcal{G}}{dN_{(j)}} \right]_i , \tag{B.4}$$

where $T_{ij}^{\phi} = T_{ij}^{N}$ as $dt \to 0$. Here we have assumed that ϕ is a signed distance function; therefore $\| \phi_k \| = 1$. In our level set implementation, ϕ is maintained as an approximation to the signed distance transform. The update on ϕ after it lags $N_{(i)}$ by some small amount is actually

$$\frac{\partial \phi}{\partial t} = \| \phi_k \| \frac{d\mathcal{G}}{d\phi}, \tag{B.5}$$

which is a gradient descent, of the level sets of ϕ, on $\mathcal{G}$— the total curvature of the surface.

Bibliography

[1] D. Mumford and J. Shah, "Boundary detection by minimizing functionals," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1985.

[2] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Comm. on Pure and Applied Mathematics*, vol. 42, pp. 577–685, 1989.

[3] N. Nordstrom, "Biased anisotropic diffusion—a unified regularization and diffusion approach to edge detection," *Image and Vision Comp.*, vol. 8, no. 4, pp. 318–327, 1990.

[4] J. Shah, "Segmentation by nonlinear diffusion," in *Proc. Conf. on Computer Vision and Pattern Recognition*, pp. 202–207, 1991.

[5] M. J. Black, G. Sapiro, and D. H. Marimont, "Robust anisotropic diffusion," *IEEE Transactions on Image Processing*, vol. 7, pp. 421–432, March 1998.

[6] T. Tasdizen, R. Whitaker, P. Burchard, and S. Osher, "Geometric surface smoothing via anisotropic diffusion of normals." To appear in IEEE Visualization, October 2002.

[7] R. M. Bolle and D. B. Cooper, "On optimally combining pieces of information, with application to estimating 3-d complex-object position from range data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 619–638, September 1986.

[8] R. Bajcsy and F. Solina, "Three dimensional object representation revisited," in *Proc. 1st Intl. Conf. on Comp. Vision*, pp. 231–240, June 1987.

[9] A. P. Pentland, "Recognition by parts," in *Proc. 1st Intl. Conf. on Comp. Vision*, pp. 612–620, June 1987.

[10] G. Taubin, "Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation," *IEEE Trans. Pattern Anal. Mach. Intelligence*, vol. 13, no. 11, pp. 1115–1138, 1991.

[11] D. DeCarlo and D. Metaxas, "Adaptive shape evolution using blending," in *Fifth International Conference on Computer Vision*, pp. 834–839, IEEE Computer Society Press, 1995.

[12] G. Turk and M. Levoy, "Zippered polygon meshes from range images," in *SIGGRAPH '94 Proceedings*, pp. 311–318, July 1994.

[13] Y. Chen and G. Médioni, "Fitting a surface to 3-D points using an inflating ballon model," in *Second CAD-Based Vision Workshop* (A. Kak and K. Ikeuchi, eds.), vol. 13, pp. 266–273, IEEE, 1994.

[14] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," *Computer Graphics (SIGGRAPH '96 Proceedings)*, July 1996.

[15] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," *Computer Graphics*, vol. 26, no. 2, pp. 71–78, 1992.

[16] A. Hilton, A. J. Stoddart, J. Illingworth, and T. Windeatt, "Reliable surface reconstruction from multiple range images," in *ECCV*, Springer-Verlag, 1996.

[17] R. T. Whitaker, "A level-set approach to 3D reconstruction from range data," *Int. J. Computer Vision*, vol. 29, no. 3, pp. 203–231, 1998.

[18] R. Whitaker and J. Gregor, "A maximum likelihood surface estimator for dense range data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, p. To appear., October 2002.

[19] J. Gregor and R. T. Whitaker, "Indoor scene reconstruction from sets of noisy range images," *Graphical Models*, vol. 63, pp. 304–332, 2002.

[20] P. Perona and J. Malik, "Scale space and edge detection using anisotropic diffusion," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 629–639, July 1990.

[21] D. Geiger and A. Yuille, "A common framework for image segmentation," *International Journal of Computer Vision*, vol. 6, no. 3, pp. 227–243, 1991.

[22] W. Snyder, Y.-S. Han, G. Bilbro, R. Whitaker, and S. Pizer, "Image relaxation: restoration and feature extraction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 620–624, June 1995.

[23] R. T. Whitaker, "Volumetric deformable models: Active blobs," in *Visualization In Biomedical Computing* (R. A. Robb, ed.), SPIE, 1994.

[24] L. Lorigo, O. Faugeras, E. Grimson, R. Keriven, R. Kikinis, A. Nabavi, and C.-F. Westin, "Co-dimension 2 geodesic active contours for the segmentation of tubular strucures," in *Proceedings of Computer Vision and Pattern Recognition*, 2000.

[25] G. Taubin, "A signal processing approach to fair surface design," in *Proceedings of SIGGRAPH'95*, pp. 351–358, 1995.

[26] U. Clarenz, U. Diewald, and M. Rumpf, "Anisotropic geometric diffusion in surface processing," in *Proceedings of IEEE Visualization*, pp. 397–405, 2000.

[27] Y. Ohtake, A. G. Belyaev, and I. A. Bogaevski, "Polyhedral surface smoothing with simultaneous mesh regularization," in *Geometric Modeling and Processing*, 2000.

[28] G. Taubin, "Linear anisotropic mesh filtering," Tech. Rep. RC22213, IBM Research Division, October 2001.

[29] D. L. Chopp and J. A. Sethian, "Motion by intrinsic laplacian of curvature," *Interfaces and Free Boundaries*, vol. 1, pp. 1–18, 1999.

[30] W. Welch and A. Witkin, "Free-form shape design using triangulated surfaces," in *Proceedings of SIGGRAPH'94*, pp. 247–256, 1994.

[31] R. Schneider and L. Kobbelt, "Generating fair meshes with g^1 boundary conditions," in *Geometric Modeling and Processing Proceedings*, pp. 251–261, 2000.

[32] R. Whitaker and V. Elangovan, "A direct approach to estimating surfaces in tomographic data," *Journal of Medical Image Analysis*, p. To appear., September 2002.

[33] G. Sapiro, *Geometric Partial Differential Equations and Image Analysis*. Cambridge University Press, 2001.

[34] D. Geiger and A. Yuille, "A common framework for image segmentation," *International Journal of Computer Vision*, vol. 6, no. 3, pp. 227–243, 1991.

[35] A. Polden, "Compact surfaces of least total curvature," tech. rep., University of Tubingen, Germany, 1997.

[36] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera, "Filling-in by joint interpolation of vector fields and gray levels," *IEEE Trans. on Image Processing*, vol. 10, pp. 1200–1211, August 2001.

[37] M. P. DoCarmo, *Differential Geometry of Curves and Surfaces*. Prentice Hall, 1976.

[38] M. Bertalmio, L.-T. Cheng, S. Osher, and G. Sapiro, "Variational methods and partial differential equations on implicit surfaces," *J. Computational Physics*, vol. 174, pp. 759–780, 2001.

[39] M. Rumpf and R. Strzodka, “Level set segmentation in graphics hardware,” in *International Conference on Image Processing*, pp. 1103–1106, 2001.

[40] S. Osher and J. A. Sethian, “Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulation,” *J. Computational Physics*, vol. 79, pp. 12–49, 1988.

[41] D. Adalsteinson and J. A. Sethian, “A fast level set method for propogating interfaces,” *J. Computational Physics*, pp. 269–277, 1995.

[42] D. Peng, B. Merriman, S. Osher, H.-K. Zhao, and M. Kang, “A pde based fast local level set method,” *J. Computational Physics*, vol. 155, pp. 410–438, 1999.

[43] J. A. Sethian, *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Material Sciences*. Cambridge University Press, 1996.