Surface Recovery from 3D Point Data Using a Combined Parametric and Geometric Flow Approach

Peter Savadjiev, Frank P. Ferrie, and Kaleem Siddiqi

Centre for Intelligent Machines, McGill University
3480 University Street, Montréal, Québec H3A 2A7, Canada
{petersv,ferrie,siddiqi}@cim.mcgill.ca

Abstract. This paper presents a novel method for surface recovery from discrete 3D point data sets. In order to produce improved reconstruction results, the algorithm presented in this paper combines the advantages of a parametric approach to model local surface structure, with the generality and the topological adaptability of a geometric flow approach. This hybrid method is specifically designed to preserve discontinuities in 3D, to be robust to noise, and to reconstruct objects with arbitrary topologies. The key ideas are to tailor a curvature consistency algorithm to the case of a set of points in 3D and to then incorporate a flux maximizing geometric flow for surface reconstruction. The approach is illustrated with experimental results on a variety of data sets.

1 Introduction

Surface reconstruction from incomplete data sets is a classical problem in computer vision. The problem consists of finding a surface *S* that approximates a physical surface *P* by using a set of point coordinates sampled from the surface *P*. These point coordinates may be corrupted with noise, due to imperfections in the acquisition of the data. Like many other problems in computer vision, the problem of surface reconstruction is ill-posed. Prior knowledge about the world and the data acquisition process must therefore be used in order to make it solvable. A good algorithm for surface reconstruction should be robust to noise and result in smooth surfaces, while recovering important structural information from the data, such as edges, ridges and holes. The presence of such structural information in the reconstructed 3D model is very important for further, higher-level processing tasks, such as shape segmentation into parts, object recognition, etc. It should impose as little restrictions on the topology of the reconstructed object as possible. These issues are taken into consideration in the research presented in this paper.

The approach to surface reconstruction presented in this article combines two different philosophies, namely that of a parametric reconstruction approach, and that of a geometric flow reconstruction approach. Many algorithms for surface recovery are based on either one of the two types of approaches, but few have attempted to

A. Rangarajan et al. (Eds.): EMMCVPR 2003, LNCS 2683, pp. 325-340, 2003.

bring the two together in order to combine their advantages. Recently, Vemuri *et al.* [29, 30] have developed an interesting hybrid model which has been applied to the problem of surface recovery. However, it is not clear how their method can be tailored to preserve and model structural features such as discontinuities.

In this paper, we introduce a novel algorithm for surface reconstruction. This method is a combination between a curvature consistency algorithm, which is a parametric surface modeling approach [9, 17, 21], and a variation of the flux maximizing flow algorithm [28], which is a geometric flow algorithm implemented through the level-set method [22]. Our approach is novel for several reasons. It uses the degrees of freedom and the ease of manipulation of parameter-free geometric flow approaches, while incorporating knowledge from the local structure of the data obtained by fitting and refining local parametric patches [22]. Contrary to other flow based algorithms for surface recovery, our method is specifically designed to preserve discontinuities in 3D data, which are a very important source of information for higher-level vision processes such as shape segmentation and object recognition. We show that the inclusion of local structural information can improve the behavior of flow based algorithms for surface reconstruction. Furthermore, our approach is an improvement over the curvature consistency algorithm in that it results in a continuous surface rather than in a patchwork of overlapping surfaces. We extend the edge preserving formulation of curvature consistency [17] to 3D, which makes our technique robust to noise, while preserving discontinuities in 3D. Finally, our approach presents a novel application of the flux maximizing geometric flow.

1.1 Previous Work

Over the years, various approaches have been put forward in order to deal with surface recovery. Traditionally, parametric surface reconstruction approaches restrict the class of possible solutions to surfaces of a certain topology by fitting a model with a known (fixed) topology to the data so that it minimizes a particular error metric. This is usually done using energy-minimizing methods. Algorithms in this category can be classified by the type of model that is fit to the data. Popular choices include global models such as generalized splines [3, 25], volumetric primitives such as generalized cylinders or superquadrics, e.g. [24]. Alternatively, [9, 10, 17, 21] make use of local parameterizations by fitting a set of local parametric patches to the data. Several parametric surface recovery algorithms were developed from the concept of active deformable models (also known as active contours), introduced in [14], e.g. [18, 26].

An important inconvenience of parametric approaches is that they impose severe topological restrictions on the reconstructed surface, e.g. they require it to be homomorphic to a sphere. In this framework, it is very hard to model surfaces with a complicated topology.

Approaches in the computational geometry domain have been proposed, e.g. [1, 8], however they are very sensitive to noise in the data, regions of high curvature and outliers. Such methods reconstruct polygonal meshes. More recently, a different class of approaches that use implicit surfaces has received much interest, e.g. [2, 6, 13]. Working within the framework of variational implicit surfaces [27], Dinh *et al.* [7] use anisotropy in order to model sharp features in reconstructions. The process for locating edges and for classifying surface points as belonging to edges, corners, or flat regions is very sensitive to noise and outliers, hence tensor filtering is required in order to smooth the reconstruction. Medioni *et al.* [12, 19] have developed a separate methodology to infer structure from sparse data, known as tensor voting. It is a non-iterative process based on tensor calculus and non-linear voting. Gomes and Mojsilovic [11] propose a related variational solution to the problem implemented through an iterative algorithm.

The concept of active contours has been extended to the implicit surface representation independently by [4] and [16]. In the context of shape segmentation, they introduce geometric active contours, which are active contours represented implicitly as a level set of a higher-dimensional scalar function. An initial contour is made to evolve under forces depending on the contour's own geometric characteristics and on image-based external forces. More recently, Vemuri *et al.* [29, 30] have developed a hybrid shape modeling scheme which is based on the notion of a pedal curve. In their work, a global prior is introduced using a parameterized model, and local properties are fine-tuned using a geometric flow.

Geometric active contours have been applied to the problem of surface reconstruction from point data in [31, 33]. Zhao *et al.* [33] propose a variational method for implicit surface reconstruction that is based on a weighted minimal surface model that behaves like an elastic membrane wrapped over the data. Their flow formulation is actually a specific instance of the conformal flow originally introduced independently in [15] and [5] (see also [23] for more recent variants). Zhao *et al.* [33] use distance to the data as the weight in the conformal flow formulation of [15] and [5]. A drawback of the method in [33] is that it cannot model accurately pronounced concavities and convexities in the data, which results in the over-smoothing of corner discontinuities and in the loss of thin structures or surface borders. This is due to the use of a curvature-based regularization term which pushes the evolving curve away from high-curvature regions in the data. The algorithm in [33] is highly dependent on a good initial approximation to the real surface.

Implicit surface representations are independent of the underlying surface parameterization. This fact allows for topological flexibility in surface reconstruction algorithms, unfortunately it also makes it very difficult to exploit structural information in the data. It is not possible to model spring forces for example, or physical objects whose rigidity and tension vary along the surface, without additional surface point tracking. Structural information is lost, to the benefit of ease of manipulation. A comparison between geometric and parametric active contour models appears in [32].

2 Method Overview

The fundamental idea behind our algorithm is to reconstruct a surface by using a field of vectors normal to the surface. We assume that the input to our algorithm consists of a set of points in 3D (with *x,y,z* coordinates) located in a regularly spaced 3D voxel grid, in which voxels that correspond to point locations have been labeled accordingly. We assume that within a small neighbourhood, there is a local coordinate frame such that the data points in that neighbourhood have been sampled from a surface that can be described in the form *w = f(u,v)*, where *(u,v,w)* are the axes of the local coordinate frame. At each data point, we compute an initial estimate of the surface normal. We regularize this initial set of normal vectors by using a new extension of the curvature consistency algorithm which preserves discontinuities in 3D. Once we have obtained a smooth field of normal vectors, we use it to reconstruct the underlying surface by applying a variant of the flux maximizing flow algorithm of [28]. The flux maximizing flow is used to align an evolving surface in 3D to be normal to a given vector field.

3 Curvature Consistency with Edge Preservation

The curvature consistency algorithm achieves a stable surface representation by iteratively minimizing a functional related to the satisfaction of local constraints on the curvature of the surface. It was introduced in [21] for reconstruction in 3D voxel-based medical images. It was reformulated in [9, 17] in the context of surface reconstruction from 2½D range images, and in [10] for surface reconstruction from stereo depth data. It provides a patchwork of overlapping, interpolating functions each of which describes the surface locally.

The method makes use of a local surface model, which describes the local neighbourhood around point *P* on the surface with a quadric patch of the form

$$w = au^2 + buv + cv^2 , \qquad (1)$$

with origin at *P* and the w axis aligned with the surface normal at *P*, $\mathbf{N_p}$, as shown in Fig. 1. The information (parameters) associated with each patch are the location of the given point, the two principal directions of the paraboloid, the minimum and maximum curvatures (along the principal directions), and a coordinate frame *(u,v,w)* with the *w* axis aligned with the normal of the patch at point *P*, the *u* and *v* axis being aligned with the two principal directions. The *u* and *v* axis span the tangent plane at point *P*, T_p. All this information is stored in an “augmented Darboux frame” $D(P)$.

An initial set of such Darboux frames is computed at every data point from local estimates of surface normals, which can be obtained through various standard methods. We use least-squares fitting of local planes. Once the initial estimates are obtained from noisy data, the patchwork of surface descriptors are refined through iterative minimization of an energy functional that limits the variation of curvature with respect to the model of the surface. At convergence, each provides a consistent representation of its local surface region with respect to its neighbours.

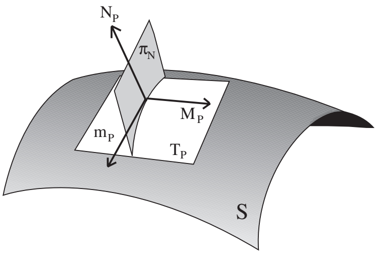


Fig. 1. Local surface representation – the augmented Darboux frame

The energy minimization is performed using variational relaxation. It is an iterative process which, at each iteration, operates by minimizing the difference between the description of the surface centered at point *P* given by *D(P)*, and that predicted by its neighbours Q_i *(i=1…n)* believed to lie on the same continuous surface as *P* and which are within a threshold distance from *P*. This set of neighbours is known as the "contextual neighborhood" [21] and is discussed further below.

Thus, at each iteration, a neighbour Q_i provides a prediction $D_i(P)$ of the augmented Darboux frame at *P*. The traditional energy-minimization approach presented in [9, 21] gives update equations of the form

$$\hat{D}(P) = \frac{1}{n}\sum_{i=1}^{n} D_i(P)\,. \tag{2}$$

In this update mechanism, each neighbour's prediction is given an equal weight in the update of the local surface descriptor. In this form of the equation, an implicit assumption of a continuous underlying surface is being made. While the effect of noise is minimized, discontinuities are smoothed over with each progressive iteration.

The authors in [17] derive a new form of Eq. (2) by casting the neighbourhood prediction update problem in estimation theoretic terms. Eq. (2) is modified such that each neighbour's contribution is weighted according to the prediction error variance, as determined by using a Kalman filter. The net effect is an anisotropic smoothing which preserves surface discontinuities. Thus, discontinuities are "learned" as the iterations proceed and do not need to be known in advance, as in [25], for example.

The edge preservation formulation of curvature consistency in [17] was designed specifically for 2½D range images. The extension to 3D of the edge-preserving formulation is straightforward, provided some technical adjustments are made to the concept of a "contextual neighborhood". In [17], the contextual neighborhood is determined simply by choosing the *n* closest neighbours of *P* in the embedding 2D grid. In 3D data, it is possible to have two or more surfaces that lie in close proximity to one another, without intersecting each other (e.g. the two sides of a thin object). Since refinement cannot proceed across surfaces, we cannot simply take the *n* closest 3D neighbors. Instead, we take those neighbors that are "sufficiently" close to the parabolic surface patch at *P*. In other words, the contextual neighborhood of *P*, consists of those neighboring points Q_i that are within radius *r* of *P* and whose local surface parameterization contains *P*, i.e. *P* is within an arbitrarily small distance from the local quadric surface patch $\Gamma = (u, v, \frac{1}{2}(\kappa_{\mathbf{M}i}u^2 + \kappa_{\mathbf{m}i}v^2))$ fit at Q_i. This is the same definition of contextual neighborhood as in [21].

4 The Flux Maximizing Geometric Flow

With the motivation to address the problem of segmenting thin elongated structures in intensity images, Vasilevskiy and Siddiqi [28] derive the geometric flow which maximizes the rate of increase of flux of an auxiliary vector field through a curve or a surface.

Let **C***(p,t)* be a smooth family of closed curves evolving in the plane. Here *t* parameterizes the family and *p* parameterizes the given curve. Without loss of generality, assume that $0 \leq p \leq 1$, i.e. that $\mathbf{C}(0,t) = \mathbf{C}(1,t)$. Consider also a vector field $\mathbf{V} = (V_1(x,y), V_2(x,y))$ defined for each point (x,y) in $\boldsymbol{R}^2$. The total inward flux of the vector field through the curve is given by the contour integral

$$Flux(t)=\int_0^1 \langle \mathbf{V},\mathbf{N}\rangle \left\| \mathbf{C}_p \right\| dp=\int_0^{L(t)} \langle \mathbf{V},\mathbf{N}\rangle ds \tag{3}$$

where $L(t)$ is the Euclidean length of the curve.

Intuitively, the inward flux through a planar closed curve provides a measure of how well the curve is aligned with the direction perpendicular to the vector field. The main theoretical result of [28] is that the direction in which the inward flux of the vector field **V** through the curve **C** is increasing most rapidly is given by

$$\frac{\partial \mathbf{C}}{\partial t}=div(\mathbf{V})\mathbf{N} \tag{4}$$

where **N** is the normal vector field of **C**. In other words, the flow which maximizes the rate of increase of the total inward flux is obtained by moving each point of the curve in the direction of the inward normal by an amount proportional to the divergence of the vector field. In the resting flux maximizing configuration, the inward normals to the curve are everywhere aligned with the direction of the vector field.

It turns out that the volumetric extension of Eq. (4) has the same form as Eq. (4). Let $\mathbf{S} : [0,1] \times [0,1] \rightarrow \boldsymbol{R}^3$ denote a compact embedded surface with (local) coordinates (u,v). The authors of [28] show that the direction in which the inward flux of the vector field **V** through the surface **S** is increasing most rapidly is given by

$$\frac{\partial \mathbf{S}}{\partial t}=div(\mathbf{V})\mathbf{N} \tag{5}$$

where **N** is the normal vector field of **S**.

The authors of [28] apply the flux maximizing flow to blood vessel segmentation. We tailor the flux maximizing flow to our problem by considering the field of normal vectors defined on the data points to be the vector field **V** whose inward flux through the evolving surface is maximized. Once the normals information has been extracted from the Darboux frame at every point, we compute an approximation to the divergence of this vector field by using a consequence of the divergence theorem, which states that the divergence at a point is defined as the net outward flux per unit area, as the area about the point shrinks to zero. Via the divergence theorem,

$$\int_{\Delta a} div(\mathbf{V}) da \equiv \int_R \langle \mathbf{V},\mathbf{N}\rangle dr \tag{6}$$

where Δa is the area, R is the bounding surface and **N** is the *outward* normal at each point on the contour surface.

For our numerical implementation we use this flux formulation along the boundaries of small spheres of varying radii. The chosen flux value at a particular location is the maximum (magnitude) flux over the range of radii. Normalization across scales is trivial, one simply has to divide by the number of entries in the discrete sum that approximates Eq. (6). We thus obtain a scalar field defined over our 3D grid.

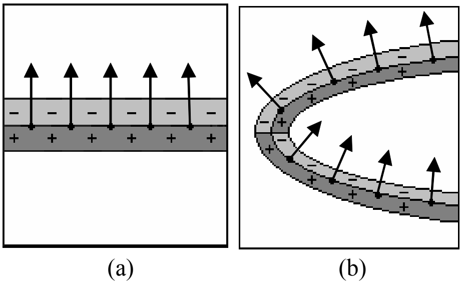


Fig. 2. Data points and their associated normal vectors, together with the flux sign distribution around them. (a) Positive and negative values form thin sheets on either side of the data points, the flux values are zero elsewhere. (b) The lack of a global notion of inside vs. outside may cause an interleaving of positive and negative sheets

Fig. 2 shows a schematic representation of a 2D planar cross-section through our 3D computational grid, for different data point configurations. As illustrated in Fig. 2(a), locations where the total outward flux (which is proportional to divergence) is positive form a thin sheet “under” the data points, and those where the total outward flux is negative form a thin sheet “above” the data points, assuming that “above” denotes the direction of the normal vectors and “under” denotes the direction opposite to it. Except for these two sheets “wrapping” the data points, the flux value is zero elsewhere. Because only local information is available, there is no reasonable notion of up vs. down on a surface, or exterior vs. interior on an object. Thus, there is always a 180° ambiguity of direction, and it is possible that the directions of the normal vectors flip from pointing “outward” to pointing “inward” from one region to another, as illustrated in Fig. 2(b). Thus, the sheets of positive and negative flux values are not necessarily continuous and can be interleaved.

According to the formulation in Eq. (5), evolving seeds initialized within a region with flux of a given sign will not leak into regions of the opposite sign, or into regions with a zero flux value. To avoid discontinuities in the reconstructed surface due to the ambiguity in normal directions, we remove the dependency on the sign of the flux value by taking its *square*. Thus, we use a numerical approximation of the following flow equation:

$$\frac{\partial \mathbf{S}}{\partial t} = div^2(\mathbf{V})\mathbf{N} \quad . \tag{7}$$

The presence of the square term in Eq. (7) implies that the flow formulation used in the hybrid method is not exactly a flux maximizing flow anymore, but rather a variant which is better adapted to deal with abrupt changes in the flux sign distribution. With this new flow formulation, we initialize a set of small seed spheres at locations where the flux magnitude is higher than a threshold, and we let them evolve according to Eq. (7). Starting the evolution at locations of high magnitude flux provides additional robustness to outliers, as outliers would be represented as isolated low magnitude flux values surrounded by zero magnitude flux values. Thus, the evolution would not spread to include them in the final reconstruction.

The flow described by Eq. (7) is implemented through the level-set representation for surfaces flowing according to functions of curvature [20]. Level-set methods represent an *n*-dimensional moving curve **C** as a level set of an *(n+1)*-dimensional scalar function ψ. Let **C** be moving with a speed F in the direction of its normal vector **N**, i.e. $\mathbf{C}_t = F\mathbf{N}$. One can show that

$$\psi_t + \mathbf{C}_t \cdot \nabla \psi = 0 \,. \tag{8}$$

Since the normal vector **N** is given by $\mathbf{N} = \nabla\psi / |\nabla\psi|$, by substitution one obtains the partial differential equation,

$$\psi_t + F|\nabla\psi| = 0 \,. \tag{9}$$

Eq. (9) is solved using a combination of straightforward discretization and numerical techniques derived from hyperbolic conservation laws [20].

The use of Eq. (7) to direct the flow of the seeds allows one to overcome the artifact depicted in Fig. 2(b), however it results in a surface that may be “thicker” than necessary, as the surface includes regions on both sides of the data points, as shown in Fig. 2. If the algorithm makes use of a global notion of outside vs. inside, it would be possible to remove the ambiguity in normal orientations, caused by using strictly local information to form initial estimates of the normal vectors. Then one may revert to using Eq. (5), which would potentially yield a thinner surface. However, the use of such a global notion implies imposing a global parametric model on the data, which goes against the philosophy of this work, which is to use only local parameterizations in order to allow the reconstruction of objects of arbitrary topology. Furthermore, it is not obvious how such a notion can be reliably computed, especially if there is more than one object in the scene. Alternative methods for thinning the surface are subject to future research, e.g. the computation of a medial surface of the “thick” reconstruction.

In the implementation presented in this paper, flux values at a point are calculated over the surface of small spheres centered at the points of interest. Since the recovery of the surface is entirely dependent on the accurate computation of flux values, future research may investigate the effect of calculating the flux through other types of surfaces, e.g. small ellipsoids or hyperboloids having the same curvature characteristics as the local surface patch at the data point.

5 Experiments and Results

In this section we present the results of an experiment that uses synthetic data to test the robustness to noise of our algorithm. We show that the application of our algorithm results in smooth reconstructions, which are qualitatively better than those obtained with the algorithm in [33].

Fig. 3(a) shows points on the surface of a hemisphere whose coordinates have been corrupted with Gaussian noise of mean 0.0, standard deviation of 1.0. The data is embedded in a 70×70×70 grid. We attempt to reconstruct the hemispherical surface using this data. As the first stage of our hybrid algorithm, we ran 15 iterations of curvature consistency. As the second and final stage of our hybrid algorithm, we applied the flux maximizing flow algorithm, using the normal vectors returned by curvature consistency. One can see the resulting surface in Fig. 3(c). The result obtained through our method, using only the flux maximizing flow and no iterations of curvature consistency to smooth the initial field of normal vectors, is shown in Fig. 3(b). It is a lot less smooth. For comparison, Fig. 3(d) shows the reconstruction from the same data obtained by using the algorithm of Zhao *et al.* [33]. The algorithm in [33] is very sensitive to noise. It does not converge to a smooth hemisphere, at some locations the surface collapses through the data points and vanishes, while at others it models the outliers in the data. These results show that the use of the local parametric information provided by the curvature consistency algorithm yields better reconstructions than the ones given either by the flux maximizing flow algorithm alone, or by the method in [33].

Our next experiment aims at showing how our method fares at recovering structure in 3D data, and at comparing it with that of Zhao *et al.* [33]. We use the Stanford bunny 3D model, which was obtained from [34]. The model is a triangle mesh with 35947 vertices and 69451 triangles that represent the surface of a bunny. We extracted the coordinates of the vertices of that mesh and we used them as a cloud of 3D points to test our algorithm. We embedded the vertices in a 160×160×160 grid. At such a grid resolution, the 35947 mesh vertices map to 30838 distinct voxels in the grid. The size of the grid was chosen as a trade-off between good data resolution and low space requirements. Fig. 4(a) shows a rendering of the voxelized data acquired from the model. Because the data is essentially noise-free, we ran our algorithm without making use of curvature consistency, we simply used the initial normal vector estimates and we ran the flux maximizing flow on them. The result of our reconstruction is shown in Fig 4(b). Fig. 4(c) shows the reconstruction obtained through the method in [33]. Note how our method captures better the discontinuities, the concavities/convexities in the data, in particular at the locations denoted by the arrows.

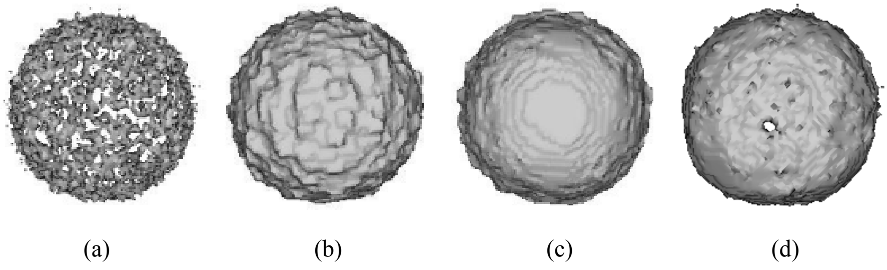


Fig. 3. (a) Data – the coordinates of the points on the hemisphere have been corrupted with zero mean Gaussian noise, standard deviation of 1.0. Reconstruction using (b) 0 iterations and (c) 15 iterations of curvature consistency in our method. (d) Reconstruction using the method of Zhao *et al.* [33]

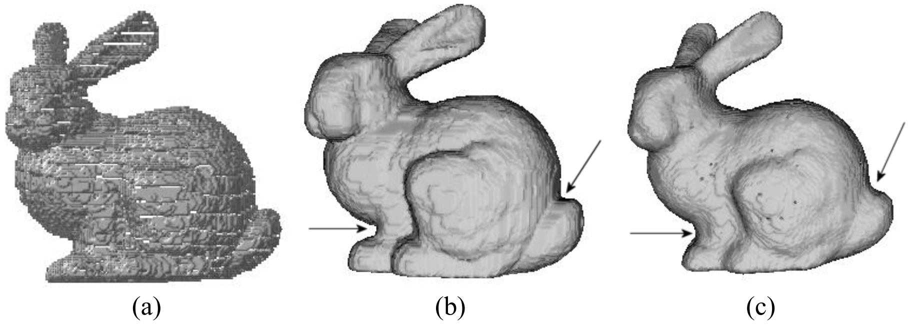


Fig. 4. The Stanford bunny. (a) Data extracted from the mesh model used in our experiments, embedded in a 160×160×160 grid. (b) Reconstruction of the Stanford bunny using our method (c) Reconstruction using the method of Zhao *et al.* [33]. Note how the structure in the data is better recovered through our method, in particular at the locations denoted by the arrows

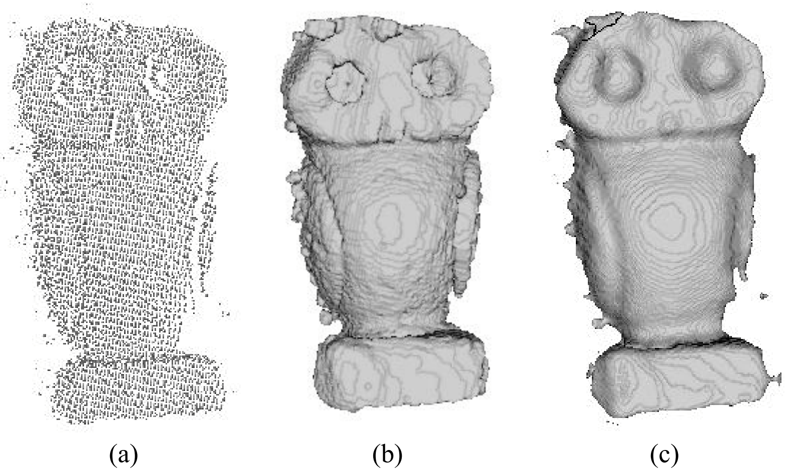


Fig. 5. (a) Owl dataset. (b) Reconstruction using our method. (c) Reconstruction using the method of Zhao *et al.* [33]. The reconstruction is smoother but does not recover discontinuities

We conducted another experiment using data acquired from a stone owl statuette through a laser range sensor. The data consists of 7648 data points distributed in a 250×162×83 voxel grid, and is shown in Fig. 5(a).

Fig. 5(b) shows the reconstruction from this data obtained with our method (using 100 iterations of curvature consistency). For comparison, we also include in Fig. 5(c) the results obtained by applying the method of Zhao *et al.* [33] on the same data. The result is smooth, but not satisfactory in terms of reconstructing the discontinuities in the data. Note how our algorithm recovers the pupil, the crease of the eye, the beak, the separation between head and body, as well as the wing, whereas the algorithm in [33] smooths them over.

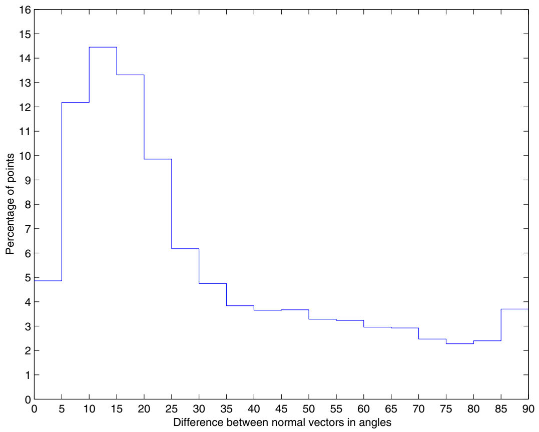
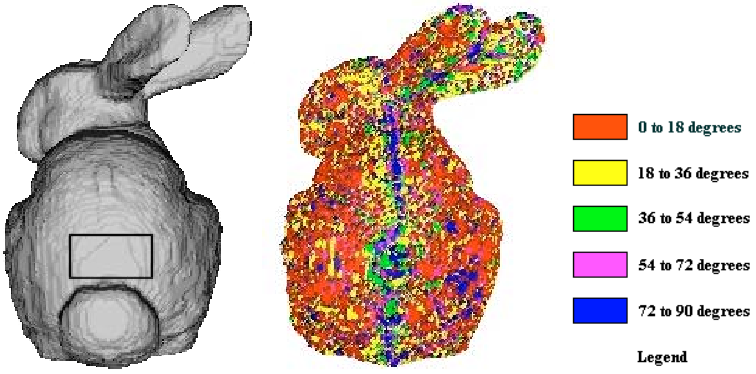


Fig. 6. Distribution of the difference in angles between the normal direction returned by curvature consistency at every data point and the normal direction on the corresponding point on the reconstructed surface of the bunny shown in Fig 7. Each bin gives the percentage of points that yield a difference falling within the bin. Median difference: 22.5°. Mean difference: 31.6°. Standard deviation: 24.2°



(a) (b)

Fig. 7. (a) Reconstruction of the bunny. The box above the tail indicates the region from which the normal vectors returned by curvature consistency are shown in Fig. 8. (b) Spatial distribution of the normal vector differences in degrees. Note: figure should be viewed in color

We next present a quantitative comparison between the curvature consistency algorithm as such, and our hybrid algorithm. The purpose of this comparison is to show that our variant of the flux maximizing flow is able to overcome to some extent the drawbacks of the curvature consistency algorithm. For each data point, we compute the smallest angle formed between the normal vector returned by curvature consistency, and the corresponding normal computed on the reconstructed surface (obtained after running the flow given by Eq. (7)). Since the reconstructed surface does not necessarily pass through each data point, we compute the normal vector on the point on the reconstructed surface which is closest to the data point. Because we compute differences between orientations, not directions, the range of possible differences is constricted to [0° - 90°].

A reconstruction of the Stanford bunny using our hybrid method (with one iteration of curvature consistency) is shown in Fig. 7(a). A histogram of the normal vector differences is shown in Fig. 6, and a spatial plot of the difference is shown in Fig. 7(b) (this figure should be viewed in color). It is possible to see that the agreement is good mostly everywhere except along the body midline. At these locations, one can observe the artifact described in Section 4 in Fig. 2(b), where the initial estimates of the normal vectors suddenly change their orientation from pointing outwards to pointing inwards. Due to the regularizing properties of the curvature consistency algorithm, a normal vector which is between two vectors that point in opposite directions (one inwards, the other outwards) will be set to point parallel to the surface. Hence, there is a difference of nearly 90° between the normal on the reconstructed surface, which is "correctly oriented", as seen in Fig. 7(a), and the normals returned by curvature consistency after one iteration, shown in Fig. 8. Fig. 8 shows a close-up on the normals of the bunny surface, from the region just above the tail indicated by the black box in Fig. 7(a). One can see that the normals change smoothly their orientation from pointing inwards to pointing outwards (going from left to right in the image). Due to the smooth transition in orientation, some normals are actually oriented parallel to the surface to be reconstructed. Our flux maximizing algorithm is able to overcome this drawback of the curvature consistency algorithm. Our algorithm can produce a surface that is oriented in the correct direction at locations where the normals returned by curvature consistency are not oriented correctly.

As discussed in Section 4, a drawback of our algorithm is that the resulting surface is "thick". By thinning the reconstructed surface and smoothing it further, one may obtain better structural properties on the surface. For example, one can expect an even better agreement between the orientation of the normals on the reconstructed surface and those returned by curvature consistency.

Our last experiment is designed to provide a quantitative measure of the localization of the reconstructed surface, by comparing it to a ground truth surface using a distance metric.

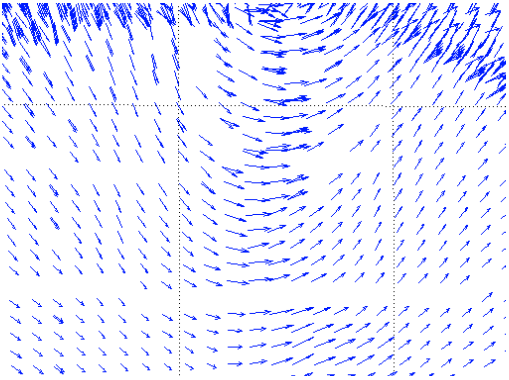


Fig. 8. A close-up on the normals on the surface of the bunny. The normals shown are taken from the back of the bunny, from the region indicated by the black box in Fig. 7(a)

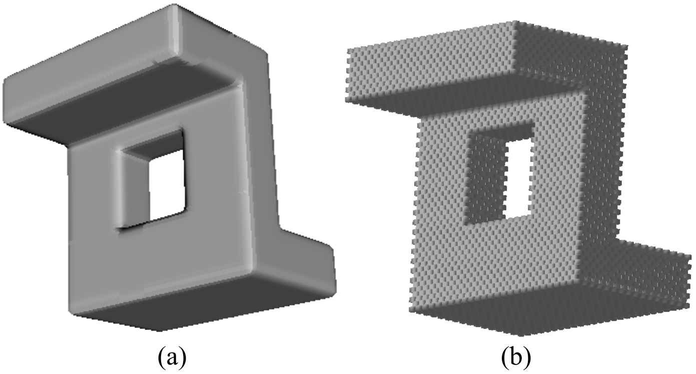


Fig. 9. (a) Ground truth surface. (b) Data sampled from the ground truth surface

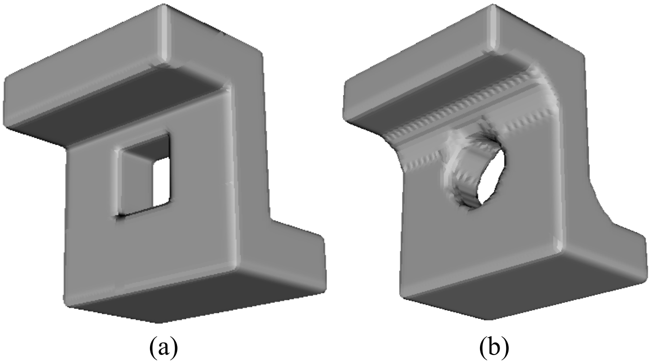


Fig. 10. (a) Reconstruction obtained through our hybrid method. u_L = 0.9, Var_L = 0.13, and max_L = 1.4. (b) Reconstruction achieved by the algorithm of Zhao *et al.*[33]. u_L = 0.2, Var_L = 0.7, and max_L = 4.7. See text for details

In the last experiment, synthetic data is used as ground truth in order to measure how well the reconstructed surface is localized with respect to the ground truth. The experiments make use of a simple distance metric *L*. Given a point on the ground truth surface, the metric *L* consists in measuring the Euclidean distance to the closest point on the reconstructed surface:

$$L = \mathrm{dist}[G_i, R_i], \tag{10}$$

where G_i [i = 1 .. n] are the points on the ground truth surface and n is the total number of points on the ground truth. For a given i, R_i is the point on the reconstructed surface closest to point G_i. Finally, *dist*[] is the Euclidean distance function. It is assumed that the ground truth surface is discretized and embedded in the same 3D grid as the reconstructed surface. It is thus possible to define a point on that surface as an individual voxel that belongs to it.

The distance metric *L* was applied to the reconstructions of the hemisphere shown in Fig. 3, using as ground truth the (noise free) hemispherical surface from which the data points shown in Fig. 3(a) were generated. The metric *L* was computed over the entire ground truth surface for the reconstructions shown in Figs. 3(c) and 3(d). For the reconstruction in Fig. 3(c), the mean of L, u_L, is 0.6, with a variance Var_L of 0.3, and a maximum value max_L of 3.7. All units are given in grid voxels, except for the units of variance. For the reconstruction in Fig. 3(d), $u_L = 1.5$, $Var_L = 0.7$, and $max_L = 4.7$. The difference between the values obtained for each reconstruction is due to the fact that, contrary to our algorithm, the algorithm in [33] is not designed to reconstruct surfaces from noisy data. Thus, our algorithm achieves a better reconstruction in the presence of noise.

The distance metric L was also applied in another experiment, presented next. The experiment makes use of the synthetic data shown in Fig. 9(b). The data consists of points sampled from the surface of a thick step with a rectangular hole in it (the original surface is shown in Fig. 9(a)), embedded in a 70×70×70 grid. Fig. 10(a) shows the reconstruction obtained from this dataset using our method, and Fig. 10(b) shows the reconstruction obtained with the algorithm of Zhao *et al.* [33]. One can see that our reconstruction models well the original ground truth (in particular the corners), except that it is thicker. The algorithm in [33], on the other hand, fails to recover some of the features of the surface. The evolving surface, "shrinking" over the data, wraps over the outward-oriented corners of the surface, but fails to enter the inward-oriented corners and smooths them over. If the surface is let to evolve further, it would collapse through the data points and create holes in the reconstruction, still without penetrating the inward-oriented corners. For the reconstruction shown in Fig. 10(a), $u_L = 0.9$, $Var_L = 0.13$, and $max_L = 1.4$. For the reconstruction shown in Fig. 10(b), $u_L = 0.2$, $Var_L = 0.7$, and $max_L = 4.7$. Thus, even though on average the method of Zhao *et al.* [33] provides a better surface localization around the (noise-free) data points, it nevertheless gives a worst-case performance which is a lot worse than the worst-case performance of our algorithm. This effect is apparent with a simple visual inspection of Figs. 9 and 10.

6 Summary and Conclusions

This paper was concerned with the problem of surface reconstruction from 3D sets of points. The paper showed that it is possible to combine the advantages of both parametric and geometric flow approaches into a single hybrid method. The hybrid method presented here is a combination of the curvature consistency algorithm and of the flux maximizing flow. It is designed to produce reconstructions of object surfaces of arbitrary topology. It is designed to preserve structure in the data (edges, ridges, and discontinuities in general), while being robust to noise. These properties are not found in standard geometric flow approaches, such as the method of Zhao *et al.* [33]. Our hybrid algorithm is able to overcome some of the drawbacks of curvature consistency. It also presents a new application of the flux maximizing flow.

Experimental results were presented to demonstrate these properties of the hybrid algorithm. Our algorithm has drawbacks and weaknesses that can be improved with future work. However, it does clearly demonstrate that the inclusion of local parametric information can improve the behavior of geometric flow algorithms, and that it is possible to use the topological modeling power of flow based algorithms together with the structural modeling power of parametric approaches to provide improved reconstruction results.

References

[1] N. Amenta, M. Bern, and M. Kamvysselis. A new Voronoi-based surface reconstruction algorithm. In *Proc. SIGGRAPH'98*, pages 415-421, 1998.

[2] C. Bajaj, F. Bernardini, and G. Xu. Automatic reconstruction of surfaces and scalar fields from 3d scans. In *Proc. SIGGRAPH'95*, pages 193-198, 1995.

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

[4] V. Caselles, F. Catte, T. Coll, and F. Dibos. A geometric model for active contours in image processing. *Numerische Mathematik*, Vol. 66, pages 1-31, 1993.

[5] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic Active Contours. In *Proc. Int. Conf. on Computer Vision (ICCV)'95*, pages 694-699, 1995.

[6] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proc. SIGGRAPH'96*, pages 303-312, 1996.

[7] H. Q. Dinh, G. Turk, and G. Slabaugh. Reconstructing Surfaces Using Anisotropic Basis Functions. In *Proc. Int. Conf. on Computer Vision (ICCV) 2001*, pages 606-613, 2001.

[8] H. Edelsbrunner and E. P. Mücke. Three dimensional α shapes. *ACM Trans. Graphics* 13, pages 43-72, 1994.

[9] F. P. Ferrie, J. Lagarde, and P. Whaite. Darboux frames, snakes, and super-quadrics: Geometry from the bottom up. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 15, pages 771-784, 1993.

[10] P. Fua and P. Sander. Reconstructing surfaces from unstructured 3d points. In *Proc. Image Understanding Workshop*, pages 615-625, 1992.

[11] J. Gomes and A. Mojsilovic. A variational approach to recovering a manifold from sample points. In *Proc. European Conf. on Computer Vision (ECCV) 2002, LNCS 2351*, Vol 2, pages 3-17, 2002.

[12] G. Guy and G. Medioni. Inference of Surfaces, 3D Curves, and Junctions from Sparse, Noisy, 3-D Data. *IEEE Trans. on Pattern Analysis and Machine Intelligence,* Vol. 19, pages 1265-1277, 1997.

[13] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. In *Proc. SIGGRAPH'92*, pages 71-78, 1992.

[14] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: active contour models. *International Journal of Computer Vision*, Vol. 1, pages 321-331, 1988.

[15] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi. Gradient flows and geometric active contour models. In *Proc. Int. Conf. on Computer Vision (ICCV)'95*, pages 810-815, 1995.

[16] R. Malladi, J. A. Sethian, and B. C. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 17, pages 158-175, 1995.

[17] S. Mathur and F. P. Ferrie. Edge Localisation in Surface Reconstruction Using Optimal Estimation Theory. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)'97*, pages 833-838, 1997.

[18] T. McInerney and D. Terzopoulos. A finite element model for 3D shape reconstruction and nonrigid motion tracking. In *Proc. Int. Conf. on Computer Vision (ICCV) '93*, pages 518-523, 1993.

[19] G. Medioni, M. S. Lee, and C. K. Tang. *A Computational Framework for Segmentation and Grouping*, Elsevier, 2000.

[20] S. J. Osher and J. A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, Vol. 79, pages 12-49, 1988.

[21] P. Sander and S. W. Zucker. "Inferring differential structure from 3-D images", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 12, pp. 833-854, 1990.

[22] P. Savadjiev. *Surface recovery from three-dimensional point data*. Master's thesis, Dept. of Electrical Engineering, McGill University, 2003.

[23] K. Siddiqi, Y. Bérubé-Lauzière, A. Tannenbaum, and S. W. Zucker. Area and length minimizing flows for shape segmentation. *IEEE Trans. on Image Processing*, Vol. 7, pages 433-443, 1998.

[24] F. Solina and R. Bajcsy. Recovery of Parametric Models from Range Images: The Case for Superquadrics with Global Deformations. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 12, pp. 131-147, 1990.

[25] D. Terzopoulos. Regularization of Inverse Visual Problems Involving Discontinuities. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 8, pp. 413-424, 1986.

[26] D. Terzopoulos and D. Metaxas. Dynamic 3D models with local and global deformations: deformable superquadrics. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 13, pages 703-714, 1991.

[27] G. Turk and J. F. O'Brien. Modeling with Implicit Surfaces that Interpolate. *ACM Trans. on Graphics,* Vol. 21, No. 4, pages 855-873, 2002.

[28] A. Vasilevskiy and K. Siddiqi. Flux maximizing geometric flows. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 24, pages 1565-1578, 2002.

[29] B.C. Vemuri and Y. Guo. Snake pedals: compact and versatile geometric models with physics-based control. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 22, pages 445-459, 2000.

[30] B.C. Vemuri, Y. Guo, and Z. Wang. Deformable pedal curves and surfaces: hybrid geometric active models for shape recovery. *International Journal of Computer Vision*, Vol. 44, pages 137-155, 2001.

[31] R. T. Whitaker. A level-set approach to 3D reconstruction from range data. *International Journal of Computer Vision*, Vol. 29, pages 203-231, 1998.

[32] C. Xu, A. Yezzi, and J. L. Prince. A summary of geometric level-set analogues for a general class of parametric active contour and surface models. In *Proc. IEEE Workshop on Variational and Level Set Methods*, pages 104-111, 2001.

[33] H. K. Zhao, S. Osher, and R. Fedkiw. Fast surface reconstruction using the level set method. In *Proc. IEEE Workshop on Variational and Level Set Methods*, pages 194-201, 2001.

[34] Stanford University 3D scanning repository: http://graphics.stanford.edu /data/3Dscanrep/