

# Reconstructing Surfaces Using Anisotropic Basis Functions

Huong Quynh Dinh and Greg Turk  
Georgia Institute of Technology

College of Computing  
Graphics, Visualization, and Usability Center  
quynh@cc.gatech.edu, turk@cc.gatech.edu

Greg Slabaugh

Georgia Institute of Technology  
School of Electrical and Computer Engineering  
Center for Signal and Image Processing  
slabaugh@ece.gatech.edu

## Abstract

*Point sets obtained from computer vision techniques are often noisy and non-uniform. We present a new method of surface reconstruction that can handle such data sets using anisotropic basis functions. Our reconstruction algorithm draws upon the work in variational implicit surfaces for constructing smooth and seamless 3D surfaces.*

*Implicit functions are often formulated as a sum of weighted basis functions that are radially symmetric. Using radially symmetric basis functions inherently assumes, however, that the surface to be reconstructed is, everywhere, locally symmetric. Such an assumption is true only at planar regions, and hence, reconstruction using isotropic basis is insufficient to recover objects that exhibit sharp features. We preserve sharp features using anisotropic basis that allow the surface to vary locally. The reconstructed surface is sharper along edges and at corner points. We determine the direction of anisotropy at a point by performing principal component analysis of the data points in a small neighborhood. The resulting field of principle directions across the surface is smoothed through tensor filtering.*

*We have applied the anisotropic basis functions to reconstruct surfaces from noisy synthetic 3D data and from real range data obtained from space carving.*

## I. Introduction

The challenge in reconstructing a smooth and continuous surface from range data is that such data is often noisy, has non-uniform density, contains holes due to occlusion, and is low in resolution when compared to optical and laser range scanners. Simply connecting the points to generate a continuous surface is insufficient because the noise in the data becomes embedded in the reconstruction. Several existing surface reconstruction techniques, including Alpha Shapes [10], the Crust algorithm [1], and the Ball-Pivoting algorithm [3], take exactly this approach. Other techniques, including those of Taubin [27], and Gotsman and Keren [15], attempt to fit a global algebraic function to the data with minimal error. The primary limitation of global algebraic

methods is their inability to reconstruct complex models because very high order polynomials would be required for such surfaces. Increasing the degree of the polynomial increases the search space and the computational complexity required to find the best fit polynomial. Methods that perform reconstruction by deforming an initial model to fit the data points are topologically limited by the initial model. Consequently, reconstruction of complex models often requires the use of multiple primitives. Such approaches include the works of Pentland and Sclaroff [21,23] and Terzopoulos and Metaxas [29].

Variational implicit surfaces solve the problem of surface reconstruction through volumetric regularization [9,30]. This approach is akin to surface regularization, which has been used by many for reconstructing height fields and parametric curves, including Terzopoulos [28], Boulton and Kender [5], and Fang and Gossard [11]. Similar to level set approaches [12,19], variational implicit reconstruction can handle complex shapes of arbitrary topology. In addition, through volumetric regularization, the implicit surface can approximate, rather than interpolate, surface points, resulting in a surface that is globally smooth.

Typically, an implicit function is formulated as a sum of weighted basis functions that are radially symmetric. In this respect, variational implicit surfaces are closely related to Blinn's blobby model which uses Gaussian functions [4]. The success of using radially symmetric basis functions, such as Gaussians, to fit dense and precise Cyberware range data has been proven by Muraki [20] and Yngve and Turk [31]. The primary advantages of the variational implicit surface technique are that it uses energy-minimizing basis functions instead of Gaussians to construct a smooth surface, does not assume that the topology of the shape to be reconstructed is known a priori, produces a continuous surface that is inherently seamless and manifold due to the nature of implicit functions, smoothly interpolates the surface where there is little or no data, and can either approximate or interpolate the data. The last two aspects are especially important with regard to vision-based data sets that are noisy and often have holes due to occlusion. Note that this method should not be confused with smoothing opera-

tors as applied to meshes or images because those methods require an initial mesh to be reconstructed from the data set. The variational implicit approach constructs a smooth model in one step.

Previous work in implicit reconstruction using basis functions use radially symmetric bases which force the implicit function to be locally similar everywhere. Such behavior is, however, erroneous at non-planar regions of the surface. For example, in the neighborhood around a point on an edge, the surface is smooth along the edge but not across it. The local behavior of the surface at such points is distinctly different from a point in a planar region. The variational implicit surface algorithm used in [9] and [30], which uses isotropic basis functions, fails to model the asymmetric nature of surface points near sharp features.

Our new approach introduces sharp features such as edges and corners into the smooth surface using anisotropic basis functions that enforce less smoothness across edges than along them. Our basis functions are radial basis functions that have been scaled non-uniformly, so their iso-contours are ellipsoids rather than spheres. The orientation of the anisotropy is determined by categorizing points as being embedded in a planar region, on an edge, or at a corner. The categorization of the data points is obtained through principle component analysis, which is also used in region growing and propagation methods such as Lee, Tang and Medioni's work on tensor voting [17, 26] and Hoppe's work on surface reconstruction [14]. These principle directions form a tensor field across the surface. We low-pass filter this tensor field to combat the noise inherent in the data. We then use the principle direction at each surface point to orient the anisotropic basis functions. The reconstructed surface is obtained by solving for the weights of the basis functions in a closed form solution, unlike region growing methods which must iterate to form a complete surface.

We have verified our approach using synthetic 3D data that has been injected with uniform noise and on real range data obtained from generalized voxel coloring [7], an extension of the voxel coloring method of Seitz and Dyer [24].

## II. Variational Implicit Surfaces

Our new approach to surface reconstruction draws upon the work in variational implicit surfaces, which uses volumetric regularization to construct a three dimensional surface that is smooth and seamless [9,30]. A similar approach was developed by Savchenko [22] to reconstruct contours and solids. In the next section, we discuss the framework behind the variational implicit approach using isotropic basis as presented by Turk and O'Brien. In Section II-B, we describe the radial basis function for multiple orders of smoothness that was used in [9]. We present our new approach using anisotropic basis functions in Section III, discuss our method of determining the direction for anisotropy in Section IV, and show reconstruction results in Section V.

### A. Volumetric Regularization

In [30], Turk and O'Brien present variational implicit surfaces as a solution to the problem of shape interpolation by minimizing a desired energy functional while interpolating data constraints. The variational implicit approach is based on the calculus of variation and is similar to surface regularization in that it defines an energy functional to be minimized. Unlike surface regularization, the energy functional is defined in  $\mathbf{R}^3$  rather than  $\mathbf{R}^2$ . Hence, the energy functional does **not** act on the space of surfaces, but rather, on the space of 3D functions. Turk and O'Brien argue that the iso-surface of a function that minimizes such an energy is also smoothly varying. A typical cost functional for regularization includes a data fitness term and a prior term:

$$H[f] = \sum_{j=1}^n \frac{1}{\lambda_i} (y_i - f(\vec{x}_i))^2 + \beta[f] \quad (1)$$

In the above equation,  $f$  is the unknown surface function,  $n$  is the number of constraints, or observed data points;  $y_i$  are the observed values of the data points at locations  $\vec{x}_i$ ;  $\beta[f]$  is the prior; and  $\lambda_i$  is a parameter to weigh between fitness to each data point and smoothness of the surface. The term  $\lambda_i$  is often called the *regularization parameter*, and is used to specify how closely to approximate the data set. The surface more closely fits a constraint point  $\vec{x}_i$  as  $\lambda_i$  approaches zero, and more loosely approximates  $\vec{x}_i$  when  $\lambda_i > 0$ . The  $\lambda_i$  value for each constraint may be determined according to the noise distribution of the data acquisition technique. The ability to pass close to, but not necessarily through, data points is especially applicable for imprecise data, such as that from voxel coloring. A derivation is presented in [13] which shows that the cost functional,  $H$ , is minimized by a function consisting of a sum of weighted basis functions:

$$f(\vec{x}) = \sum_{i=1}^n w_i \phi(\vec{x} - \vec{c}_i) + P(\vec{x}) \quad (2)$$

In the above equation,  $\phi(\vec{x} - \vec{c}_i)$  is the basis function centered at  $\vec{c}_i$ ;  $n$  is the number of constraint points (each constraint corresponds to a basis);  $w_i$  are the weights for the basis functions; and  $P(\vec{x})$  is a polynomial term. Constraints are placed at surface points, points in the interior of the object, and exterior points surrounding the object. The polynomial term in Equation 2 spans the null space of the basis function. The unknowns,  $w_i$  and the coefficients of  $P(\vec{x})$ , are found by solving the following linear system:

$$\begin{bmatrix} \phi(r_{11}) + \lambda_1 & \cdot & \phi(r_{1n}) & \vec{c}_1 \\ \vdots & & \vdots & \vdots \\ \phi(r_{n1}) & \cdot & \phi(r_{nn}) + \lambda_n & \vec{c}_n \\ \vec{c}_1 & \cdot & \vec{c}_n & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_n \\ \vec{p} \end{bmatrix} = \begin{bmatrix} f(\vec{c}_1) \\ \vdots \\ f(\vec{c}_n) \\ 0 \end{bmatrix} \quad (3)$$

$$r_{ij} = |\vec{c}_i - \vec{c}_j| \quad (4)$$

The function value,  $f(\vec{c}_i)$ , at each constraint point is known since we have defined the constraint points to be on

the surface, or internal or external to the object. In the case of an implicit function that evaluates to zero on the surface, the known function value for each surface constraint is zero. We place all exterior constraints at the same distance away from the surface and assign them a function value of -1.0. All interior constraints are assigned a function value of 1.0. The matrix consists of the evaluation of the basis function at the Euclidean distance between each pair of constraints. For surfaces, constraints are specified by 3D coordinates,  $\vec{c} = (c_x, c_y, c_z)$ . Once the solution to the unknown weights is found, the 3D implicit function is completely defined by Equation 2. The implicit surface is a level-set of the 3D implicit function where it evaluates to zero.

## B. A Radial Basis Function for Multiple Orders of Smoothness

The prior in Equation 1 may take on a variety of forms, such as a thin-plate term. In [9], the authors use a prior that is a combination of first, second and third order energies. The associated functional is similar to Laplace's equation,  $-\Delta f = 0$ , but also has higher order terms:

$$-\delta \Delta f + \Delta^2 f - \tau \Delta^3 f = 0 \quad (5)$$

In the above equation the Laplacian operator in 3D is:

$$\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2} \quad (6)$$

In equation 5, the amount of first order smoothness is specified by  $\delta$ , and  $\tau$  controls the amount of third order smoothness. The balance between  $\delta$  and  $\tau$  controls the amount of second order smoothness. The radial basis that inherently minimizes the above prior is derived in [6] and is given below:

$$\phi(r) = \frac{1}{4\pi\delta^2 r} \left( 1 + \frac{we^{-\sqrt{v}r}}{v-w} - \frac{ve^{-\sqrt{w}r}}{v-w} \right) \quad (7)$$

$$v = \frac{1 + \sqrt{1 - 4\tau^2\delta^2}}{2\tau^2} \quad w = \frac{1 - \sqrt{1 - 4\tau^2\delta^2}}{2\tau^2} \quad (8)$$

In the above equations,  $r$  is the distance from an arbitrary point to the center of the radial basis function. The basis is isotropic due to the directionally unbiased energy functional defined in Equation 5.

The only free parameters in defining the basis function are  $\delta$  and  $\tau$ . By plotting the basis for various values of  $\delta$  and  $\tau$ , one finds that although the basis has infinite support, it quickly falls off towards zero. The basis function falls off more rapidly as  $\delta$  is increased. As  $\tau$  is increased, the center of the basis becomes increasingly smooth. In [9], the authors show through measures of fitness and curvature that values of  $\lambda$  between 0.001 to 0.003,  $\delta$  values between 10.0 to 40.0 and  $\tau$  values between 0.005 to 0.01 are appropriate for a variety of vision-based data sets. Note that although

the appropriate values for  $\delta$  and  $\tau$  depend on the distance between the constraint points (or, basis centers), the parameters are quite robust and require little tuning as discussed in [9].

In this work, we apply the variational implicit surface approach using the multi-order basis to more challenging vision-based data sets that include sharp edges and corners. The results shown in [9] are reconstructions of organic data sets that exhibit smooth features and contain less local detail. Our new approach modifies the isotropic radial basis function described earlier to create an anisotropic basis that treats edges and corners differently than planar, smooth regions. In addition, we adaptively set the regularization parameter,  $\lambda$ , to force the surface to tightly fit the data near edges and corners and more loosely approximate the data in planar regions.

## III. Anisotropic Basis Functions

Surface points can be characterized as embedded in a planar region, at an edge, or at a corner. The surface is radially symmetric around planar points. For a point on the edge, the surface is smooth along the edge but falls off sharply across the edge. At a corner, the surface falls off sharply in many directions. Sharp changes in the surface are associated with discontinuous derivatives which are minimized by the prior described in Section II-B. Adaptively specifying  $\delta$  and  $\tau$  to be smoother at planar points and sharper at edges and corners does not appropriately model the asymmetric nature of the surfaces, however, because the basis remains radially symmetric. We had found, in practice, that spatially varying the smoothing parameters,  $\delta$  and  $\tau$ , failed to maintain smoothness along edges while reducing continuity across them.

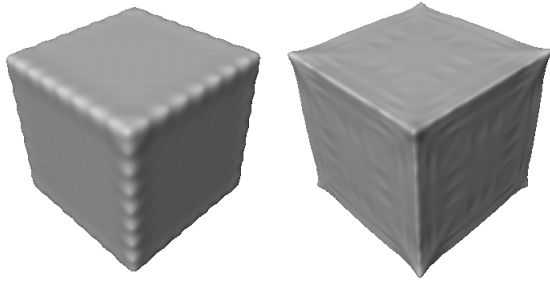
Our new approach is to model the asymmetric nature of surface points by making the multi-order basis function fall off to zero anisotropically. This is analogous to anisotropic support for basis that have finite support (the basis of Equations 7 and 8 have infinite support). Ideally, the support should be reduced across edges and at corners where gradient and curvature is discontinuous, while it should be maintained at planar regions and along edges where the continuity must be preserved. This is achieved by an anisotropic basis function that falls off more rapidly in one direction than another. We construct such a basis  $B(\vec{x})$  by non-uniformly scaling the distance from the center  $\vec{c}$  of the basis to a point  $\vec{x}$  along the direction of anisotropy:

$$B(\vec{x}) = \phi(|M(\vec{x} - \vec{c})|) \quad (9)$$

In the above equation,  $M$  is the scaling function. In practice, the direction of anisotropy is along the principle axes of a corner or edge point, which we later describe in Section IV. Note that  $\phi(r)$  remains unchanged, and Equations 7 and 8 are directly applied to the scaled distance to obtain the anisotropic basis,  $B(\vec{x})$ . The non-uniform scaling allows the basis to approach zero more rapidly in some directions

than others. As a result, for surface constraints on an edge, the basis is oriented so that it approaches zero more rapidly across the edge than along the edge. For corner constraints, the basis approaches zero more rapidly in all directions to allow a break in continuity at that point. For planar regions, the anisotropic basis reduces to an isotropic, radially symmetric basis function.

Figure 1 is a synthetic example of the reconstruction of a cube from 770 sample points. 3D surface constraints were uniformly sampled from a cube of dimensions  $2.0 \times 2.0 \times 2.0$ . Values of  $\delta = 5.0$  and  $\tau = 0.05$  were used in the reconstruction to span the average distance of 0.19 between sample points. The reconstruction using isotropic basis functions is shown on the left, while the reconstruction on the right was created using anisotropic basis functions. The isotropic function is radially symmetric and produces a rounded surface even at corner points. In contrast, the anisotropic basis is able to reproduce sharp edges and corners. Notice how the anisotropic basis provides more support along an edge than across it, while the basis at the center of the cube faces behave isotropically. The reconstruction using the anisotropic basis exhibits sharper corners and edges, and more faithfully reproduces the polygonal cube.



**Fig. 1. Isotropic (left) and anisotropic (right) reconstructions of a synthetic cube from 770 point samples using values of  $\delta = 5.0$  and  $\tau = 0.05$ .**

figure

As shown in Figure 1, the direction of the anisotropy is essential to the reconstruction. In the next section, we discuss how we determine the direction of anisotropy using principle component analysis.

#### IV. Direction of Anisotropy

In order to determine whether a surface point,  $\vec{p}$ , is part of a planar region, part of an edge, or occurs at a corner, we look at the number of dominant principle axes spanned by the neighborhood of points  $\vec{v}_i$  within radius  $r$  of  $\vec{p}$ . To identify the principle axes of the local data set, we perform eigenvalue decomposition of the covariance matrix. The covariance matrix,  $C(\vec{p})$ , is a  $3 \times 3$  symmetric tensor that describes the distribution of points in the neighborhood, and is computed as the sum of the outer product between each

surface point (minus the centroid of the neighborhood) and its transpose:

$$C(\vec{p}) = \sum_{i=1}^n (\vec{v}_i - \vec{c})(\vec{v}_i - \vec{c})^T \quad (10)$$

In the above equation,  $n$  is the number of points in the neighborhood. Each  $\vec{v}_i$  is the location of a surface point in the neighborhood, and  $\vec{c}$  is the centroid of the neighborhood. Our application of principle component analysis differs from that used by Lee, Tang and Medioni in tensor voting [17, 26] and by Hoppe in his surface reconstruction method [14] in that  $C(\vec{p})$  is **not** computed using the estimated normal vector of the surface points. Instead, we use the distribution of the local surface points themselves. Robust estimation of the surface normal at the sample points requires that the data set be very precise, which we do not assume in our approach.

The resulting eigenvectors of  $C$  are the three principle axes of the set of points in the neighborhood. The eigenvalues indicate the strength of the corresponding eigenvector and help characterize the local point set. When all three eigenvalues are nearly equal, then there is no one or two dominant axes along which all the points span. In such a case, the point set within the neighborhood is likely to be a corner, as long as the local point set is a thin-shelled surface. If there is one strong eigenvalue, then there is one dominant principle axis, and all the points in the neighborhood span that axis. Such a situation is likely to occur along edges, and the dominant axis is the orientation of the edge. If all the points lie on a plane, then they span the space of two vectors (or axes). In such a case, there will be two eigenvalues that are nearly equal and larger than the third. The two corresponding eigenvectors form the plane along which the points span, and the eigenvector corresponding to the weakest eigenvalue is the estimated normal vector to the plane.

Once the surface points have been characterized as part of a planar region, part of an edge, or at a corner, the non-uniform scaling is applied to the basis function as described in Section III so that it behaves anisotropically. In particular, for an edge point, the direction of anisotropy is aligned with the dominant principle axis. The scaling function,  $M$ , of Equation 9 is applied to the vector between the edge point and an arbitrary point in 3D space to be evaluated, denoted by  $\vec{x} - \vec{c}$  in Equation 9. The scaling function contracts the vector in the direction of the principle axis by a factor of  $0.5 \times$  the ratio between the eigenvalues associated with the minor axis and the major axis. This contraction causes the basis function in Equations 7 and 8 to be evaluated at a smaller radius than before scaling. Since the basis,  $\phi(r)$ , approaches zero as  $r$  increases, a smaller  $r$  evaluates to a larger basis value. Hence, the anisotropic basis approaches zero less rapidly in the direction of the principle axis, but more rapidly across the edge which is orthogonal to the principle axis.

### A. Filtering the Tensor Field

Each covariance matrix, defined by Equation 10 is a tensor. We obtain a field of tensors over the entire surface by performing the principle component analysis on each surface point in the voxel coloring data set (not just the surface constraints that we uniformly sample as described later in Section V-A). Principle component analysis is sensitive to noise in the data. For example, if a neighborhood of points is **not** a thin-shelled surface due to noise, then the categorization of the surface point by the method described in the previous section may be incorrect. In addition, the juxtaposition of very different tensors will often result in a rough or wrinkled surface. Figure 2 are reconstructions of a synthetic example of a cube whose surface points have been injected with uniform noise that deviates from the noise-free surface by up to ten percent of the cube dimensions. The left panel of Figure 2 shows the reconstruction of the noisy cube using the anisotropic, multi-order basis without additional filtering of the tensor field. The edges and corners are fairly sharp, but the faces are quite rough.

In order to combat the noise present in the tensor field, we apply a low pass filtering of the tensors using a Gaussian-like kernel. We position the kernel at each surface point. The resulting tensor at the center of the filter is the summation of weighted tensors of all surface points within the support of the filter. Additionally, we modify the support of the filter so that it is anisotropic. We align the filter so that it has greater support along the dominant principle axis of the original tensor at the surface point. This anisotropy has little effect for points in planar regions since the tensors at such points do not have one dominant principle axis, and thus, the support is equal along both dominant axes. The weight of a neighboring tensor is dependent on the distance of the neighboring surface point to the center of the filter. This distance is applied to the anisotropic Gaussian kernel to obtain the weight for the unfiltered tensor. The filtered tensor,  $C'(\vec{p}_i)$ , at a surface point,  $\vec{p}_i$ , is as follows:

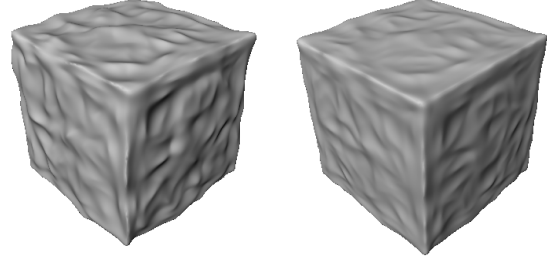
$$C'(\vec{p}_i) = \frac{\sum_{j=1}^n w_{ij} C(\vec{p}_j)}{\sum_{j=1}^n w_{ij}} \quad (11)$$

$$w_{ij} = e^{-\frac{|\vec{p}_i - \vec{p}_j|^2}{2r^2}} \quad (12)$$

In the above equation,  $n$  is the number of surface points within the filter radius  $r$ ;  $\vec{p}_j$  are the coordinates of the surface points; and  $C(\vec{p}_j)$  is the unfiltered tensor of  $\vec{p}_j$ . The right panel of Figure 2 shows the reconstruction of the noisy cube after tensor filtering.

### B. Basis, Neighborhood, and Filter Radii

The span of the basis function, the neighborhood radius used in calculating the unfiltered tensor at each surface point, and the filter radius are all dependent on the sampling density of the data set. The basis function needs to be large enough to span the samples. For noisy data sets, the radius



**Fig. 2. Reconstructions of the noisy cube without tensor filtering (left) and with tensor filtering (right).**

figure

for calculating the tensor at each surface constraint needs to be large in order to be insensitive to the noise. For example, voxel coloring data sets tend to be aliased due to voxelization. A radius of four or more voxels is required to avoid biases due to the voxelization. The filter radius depends on the noise level of the initial tensors at each surface point. We have used a radius of 0.6 to calculate the initial tensors and a filter radius of 0.4 for the synthetic data set sampled from a  $2.0 \times 2.0 \times 2.0$  cube injected with uniform noise that deviates up to 0.2 away from the noise-free surface.

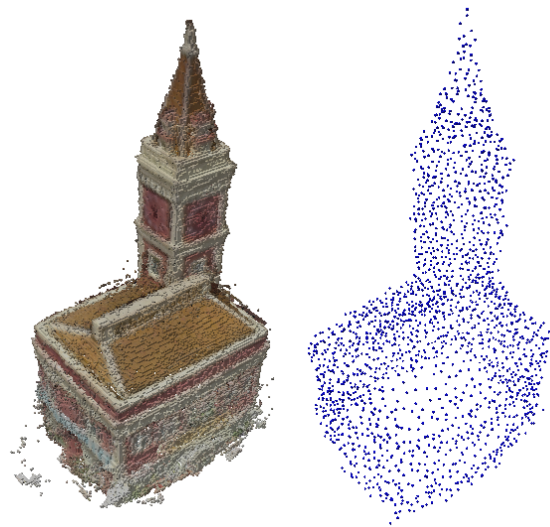
## V. Results

We have applied our approach of surface reconstruction using anisotropic basis functions to a real data set obtained through generalized voxel coloring [7]. The data set is an eight inch tall, souvenir model of Ghirardelli Square. Figure 4 shows actual images of the model. The thin-shelled, volumetric data set consists of 72,101 surface voxels, carved from a volume of  $168 \times 104 \times 256$  voxels using 17 images.

### A. Constraints from Voxel Coloring Data

The generalized voxel coloring algorithm is a space carving approach that begins with a 3D volume and a set of images from arbitrary viewpoints around the object to be reconstructed [7]. Voxels are carved out of the initial volume by splatting each voxel towards each camera and determining the consistency of the voxel color across the images. If the variance in color intensity is below a specified threshold, the voxel is kept as part of the object surface. Otherwise, it is cast out and assigned a zero value. The resulting data is a thin-shelled, volumetric representation of the surface, and consists of red, green, and blue channels for each voxel.

As mentioned in Section II-A, constraint points are specified to be on the surface, or internal or external to the object. Surface constraints are not generated from the entire data set because the system matrix in Equation 3 would become too large, and the reconstructed surface would overfit the data, resulting in overshoots. Instead, we uniformly sample the data set to reduce it to several thousand surface points (examples in this section use only 2000 surface points). Our results show that this reduced data set is suffi-



**Fig. 3. Voxel coloring data set of the eight inch Ghirardelli Square model (left) and the 2000 surface points used in reconstructing the variational implicit surface (right).**

figure  
 cient to generate detailed surfaces using our reconstruction algorithm. In selecting constraints, we use the method described in [9] to uniformly sample the surface, interior, and exterior regions of the object. This method is based on the notion of *free space* as introduced in [8]. Free space is the region of space swept out by the collection of rays emanating from a camera towards surface points visible from that camera. Using the a priori information about the cameras used in generating the voxel coloring data set, we can define exterior points all around the object. Surface points are defined by the voxel coloring data set itself. Interior points are determined by traversing the volumetric data set along each major axis. During the traversal, all points occurring between pairs of surface voxels are marked as interior. Only voxels which are marked as interior by two or more traversals are kept as interior constraints. The left panel in Figure 3 shows the entire voxel coloring data set, and the right panel shows the 2000 surface constraints that were uniformly sampled from the data set and used in the anisotropic variational implicit surface reconstruction.

## B. Reconstructions

The voxel coloring data set, shown in Figure 3, is aliased due to voxelization and contains floating surface points due to noise and non-textured regions of the model. We have extracted the largest single connected component to remove the floating voxels. We uniformly sample the surface, interior, and exterior to obtain 2000 surface, 50 interior, and 200 exterior constraints. All constraints were centered and normalized so that the reconstructed model is within a bounding box spanning -1.0 to 1.0 along each axis.

Figure 5 is a comparison of reconstructions using the



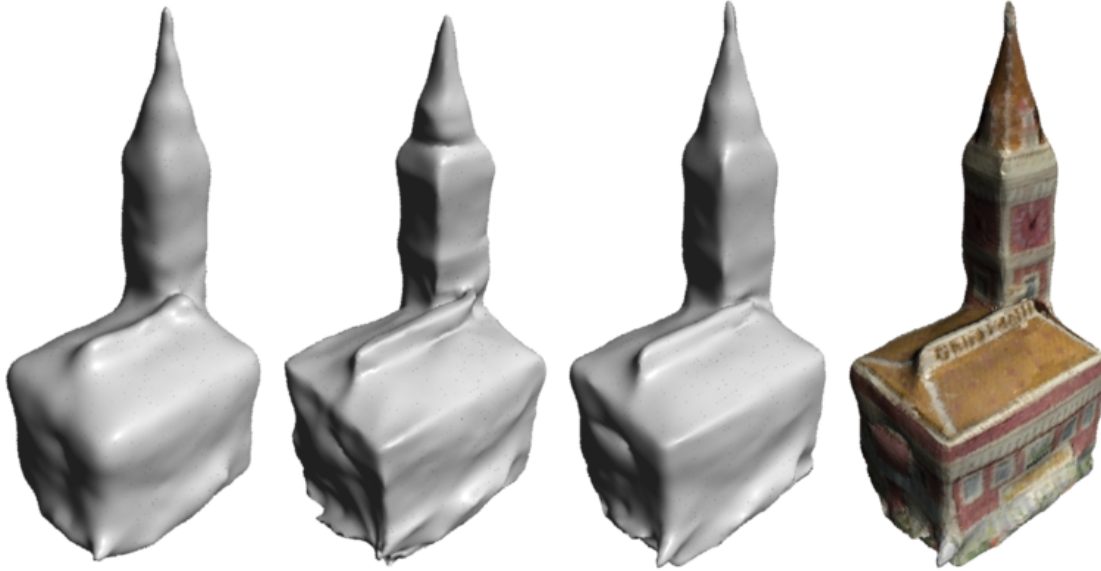
**Fig. 4. Actual images of the Ghirardelli souvenir model (left), and reconstructed model rendered at the same camera viewpoints (right).**

figure

isotropic basis(left panel), using the anisotropic basis without tensor filtering (second panel), and using the anisotropic basis with tensor filtering (third panel). The last panel is a textured version of the reconstruction shown in the third panel. In all four reconstructions,  $\lambda = 0.001$  at sharp features and  $\lambda = 0.01$  in planar regions,  $\delta = 10.0$ ,  $\tau = 0.01$ , the neighborhood radius for the initial tensors is 8.0, and the filtering radius when tensor filter was used is 8.0. The average Euclidean distance error measured on all four models from a random sampling of 10,000 surface points are 0.0137 for the isotropic reconstruction, 0.0123 for the anisotropic reconstruction without tensor filtering, and 0.0120 for the anisotropic reconstruction with tensor filtering.

Visually, the reconstructions using an anisotropic basis with tensor filtering (shown in the third and fourth panels) are the best at capturing sharp features while smoothing out





**Fig. 5. Comparison of reconstructions using isotropic basis (first), using anisotropic basis without additional tensor field filtering (second), using anisotropic basis with additional tensor filtering (third), and the final textured reconstruction (fourth). In all four reconstructions,  $\lambda = 0.001$  for surface points at an edge or corner, and  $\lambda = 0.01$  for planar points.**

figure

planar regions of the model. The differences between the reconstructions using the isotropic (first panel) versus the anisotropic basis is especially apparent at the edges and corners. The isotropic reconstruction has rounded edges and corners, while those of the anisotropic reconstruction are sharp and well-defined. The differences between the unfiltered (second panel) and filtered (third panel) reconstructions using the anisotropic basis is quite subtle. It is clear, however, that the planar regions of the unfiltered reconstruction are more wrinkled. This is also evident in the average error measurements, which show a larger difference between the isotropic and anisotropic reconstructions than between the filtered and unfiltered reconstructions. The error measurements do, however, support the visual notion that the reconstruction using the anisotropic basis with tensor filtering is the best of the three, the reconstruction using the anisotropic basis without additional filtering is a close second best, and the reconstruction using the isotropic basis is the worst of the three types of reconstructions.

The final step in our reconstruction is to texture the reconstructed model. We first extract a polygonal model from the implicit surface using *Marching Cubes* [18]. The polygonal model is then subdivided until all triangles of the mesh project to less than a pixel in area for all the input images. We assign color to each triangle of the polygonal model by projecting each triangle to the input image whose corresponding camera has the best unobscured view of the triangle. Typically several cameras will have an unobscured view of any one triangle, but the one with the best view is the one that has a view most nearly parallel to the trian-

gle normal. This is determined by taking the dot product between the triangle normal vector and the view direction vector of the camera. Finally to reduce aliasing, we filter the color of each triangle by taking the average of the colors of the triangle and its neighbors. Figure 4 is a comparison of two of the original images used in voxel coloring with the textured reconstruction using anisotropic basis functions. The last panel of Figure 5 shows the textured reconstruction from a novel viewpoint.

## VI. Conclusion and Future Work

We have described a method for reconstructing surfaces using anisotropic basis functions that are guided by a tensor field. This method well captures sharp features such as corners and edges because the anisotropic basis more appropriately models the asymmetric nature of the surface near such features than does the isotropic basis. The anisotropy allows the surface to be sharper along different directions across a surface point. Additionally, our method is insensitive to noise that is often present in 3D data sets obtained from images. The variational implicit surfaces are allowed to approximate rather than interpolate the data through volumetric regularization. We have also filtered the tensor field that determines the directions of anisotropy to reduce the effects of noise in the data set. Reconstructions of synthetic data injected with uniform noise and real voxel coloring data show that our approach preserves sharp edges and corners better than isotropic reconstruction, and succeeds in smoothing planar regions.

We are currently looking at improved methods for filtering the tensor field, including anisotropic diffusion. Additionally, we are looking into potential methods for incorporating the variational implicit surface approach into the space carving algorithm. Currently, there is no notion of a surface as voxels are carved away. Voxels are kept as a surface voxel only if its color is consistent with the images of the object. We propose that the reconstructed surface be used as an additional guide in rejecting voxels and in refining the threshold variance above which voxels are rejected.

## VII. Acknowledgements

We thank James O'Brien for stimulating discussions about alternate basis functions for surface creation. We also acknowledge Bruce Culbertson, and Tom Malzbender of *Hewlett-Packard* for use of the generalized voxel coloring algorithm. This work was funded by the *Graphics, Visualization, and Usability Center Seed Grant* in collaboration with Ronald Schafer of the *Center for Signal and Image Processing, Georgia Institute of Technology*.

## References

- [1] Amenta, N., M. Bern, and M. Kamvyselis. A New Voronoi-Based Surface Reconstruction Algorithm. *SIGGRAPH Proceedings*, Aug. 1998, pp. 415–420.
- [2] Bajaj, C. L., F. Bernardini, and G. Xu. Automatic Reconstruction of Surfaces and Scalar Fields from 3D Scans. *SIGGRAPH Proceedings*, Aug. 1995, pp. 109–118.
- [3] Bernardini, F., J. Mittleman, H. Rushmeier, and C. Silva. The Ball-Pivoting Algorithm for Surface Reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4), Oct.-Dec. 1999, pp. 349–359.
- [4] Blinn, J. A Generalization of Algebraic Surface Drawing. *ACM Transactions on Graphics*, 1(3), July 1982, pp. 135–256.
- [5] Boulton, T.E. and J.R. Kender. Visual Surface Reconstruction Using Sparse Depth Data. *Computer Vision and Pattern Recognition Proceedings*, 1986, pp. 68–76.
- [6] Chen, F. and D. Suter. Multiple Order Laplacian Spline - Including Splines with Tension. *MECE 1996-5, Dept. of Electrical and Computer Systems Engineering Technical Report, Monash University*, July 1996.
- [7] Culbertson, W. B., T. Malzbender, and G. G. Slabaugh. Generalized Voxel Coloring. *Vision Algorithms: Theory and Practice*. Part of Series: *Lecture Notes in Computer Science*, 1883, Sept. 1999, pp. 67–74.
- [8] Curless, B. and M. Levoy. A Volumetric Method for Building Complex Models from Range Images. *SIGGRAPH Proceedings*, Aug. 1996, pp. 303–312.
- [9] Dinh, H. and G. Turk. Reconstructing Surfaces by Volumetric Regularization. *Technical Report 00-26, Graphics, Visualization, and Usability, Georgia Institute of Technology*, November, 2000.
- [10] Edelsbrunner H. and E.P. Mücke. Three-Dimensional Alpha Shapes. *ACM Transactions on Graphics*, 13(1), Jan. 1994, pp. 43–72.
- [11] Fang, L. and D. Gossard. Multidimensional Curve Fitting to Unorganized Data Points by Nonlinear Minimization. *Computer-Aided Design*, 27(1), Jan. 1995, pp. 48–58.
- [12] Faugeras, O. and R. Keriven. Complete Dense Stereovision Using Level Set Methods. *Proceedings Fifth European Conference on Computer Vision*, 1998.
- [13] Girosi, F., M. Jones, and T. Poggio. Priors, Stabilizers and Basis Functions: From Regularization to Radial, Tensor and Additive Splines A.I. Memo No. 1430, C.B.C.L. Paper No.75, Massachusetts Institute of Technology Artificial Intelligence Lab, June, 1993.
- [14] Hoppe, H., T. DeRose, and T. Duchamp. Surface Reconstruction from Unorganized Points. *SIGGRAPH Proceedings*, July 1992, pp. 71–78.
- [15] Keren, D. and C. Gotsman. Fitting Curve and Surfaces with Constrained Implicit Polynomials. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(1), pp. 21–31, 1999.
- [16] Kutulakos, K.N. and S.M. Seitz. A Theory of Shape by Space Carving. *International Journal of Computer Vision*, 38(3), July 2000, pp. 198–218.
- [17] Lee, M. and G. Medioni. Inferring Segmented Surface Description from Stereo Data. *Computer Vision and Pattern Recognition Proceedings*, 1998, pp. 346–352.
- [18] Lorensen, W.E. and H.E. Cline. Marching Cubes. *SIGGRAPH Proceedings*, July 1987, pp. 163–169.
- [19] Malladi, R., J.A. Sethian, and B.C. Vemuri. Shape Modeling with Front Propagation: A Level Set Approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2), Feb. 1995.
- [20] Muraki, S. Volumetric Shape Description of Range Data Using Blobby Model. *SIGGRAPH Proceedings*, 1991, pp. 227–235.
- [21] Pentland, A. Automatic Extraction of Deformable Part Models. *International Journal of Computer Vision*, Vol.4, 1990, pp.107–126.
- [22] Savchenko, V.V., A.A. Pasko, O.G. Okunev, and T.L. Kunii. Function Representation of Solids Reconstructed from Scattered Surface Points and Contours. *Computer Graphics Forum*, Vol. 14, No. 4, 1995, pp.181–188.
- [23] Sclaroff, S. and A. Pentland. Generalized Implicit Functions for Computer Graphics. *SIGGRAPH Proceedings*, July 1991, pp.247–250.
- [24] Seitz, S.M. and C.R. Dyer. Photorealistic Scene Reconstruction by Voxel Coloring. *International Journal of Computer Vision*, 35(2), 1999, pp. 151–173.
- [25] Suter, D. and F. Chen. Left Ventricular Motion Reconstruction Based on Elastic Vector Splines. To appear in *IEEE Transactions on Medical Imaging*, 2000.
- [26] Tang, C. and G. Medioni. Inference of Integrated Surface, Curve, and Junction Descriptions From Sparse 3D Data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11), Nov. 1998.
- [27] Taubin, G. An Improved Algorithm for Algebraic Curve and Surface Fitting. *Proceedings Fourth International Conference on Computer Vision*, May 1993, pp. 658–665.
- [28] Terzopoulos, D. The Computation of Visible- Surface Representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(4), July 1988.
- [29] Terzopoulos, D. and D. Metaxas. Dynamic 3D Models with Local and Global Deformations: Deformable Superquadrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 7, July, 1991.
- [30] Turk, G. and J.F. O'Brien. Shape Transformation Using Variational Implicit Functions. *SIGGRAPH Proceedings*, Aug. 1999, pp. 335–342.
- [31] Yngve, G. and G. Turk. Creating Smooth Implicit Surfaces from Polygonal Meshes. *Technical Report 99-42, Graphics, Visualization, and Usability, Georgia Institute of Technology*, 1999.