Reconstructing Surfaces by Volumetric Regularization

Huong Quynh Dinh[1], Greg Turk[1], and Greg Slabaugh[2]

[1]Graphics, Visualization, and Usability Center
College of Computing
Georgia Institute of Technology

[2]DSP
Department of Electrical and Computer Engineering
Georgia Institute of Technology

Abstract

We present a new method of surface reconstruction that generates smooth and seamless models from sparse, noisy, and non-uniform range data. Data acquisition techniques from computer vision, such as stereo range images and space carving, produce three dimensional point sets that are imprecise and non-uniform when compared to laser or optical range scanners. Traditional reconstruction algorithms designed for dense and precise data cannot be used on stereo range images and space carved volumes. Our method constructs a three dimensional implicit surface, formulated as a summation of weighted radial basis functions. We achieve three primary advantages over existing algorithms: (1) the implicit functions we construct estimate the surface well in regions where there is little data; (2) the reconstructed surface is insensitive to noise in data acquisition because we can allow the surface to approximate, rather than exactly interpolate, the data; and (3) the reconstructed surface is locally detailed, yet globally smooth, because we use radial basis functions that achieve multiple orders of smoothness.

1 Introduction

The computer vision community has developed numerous methods of acquiring three dimensional data from images. Some of these techniques include shape from shading, depth approximation from a pair of stereo images, and volumetric reconstruction from images at multiple viewpoints. The advantage of these techniques is that they use cameras, which are inexpensive resources when compared to laser and optical scanners. Because of the affordability of cameras, these vision-based techniques have the potential to enable the creation of digital models by home computer users who may not have professional CAD training. The data that is obtained is, however, comparatively more noisy, more non-uniform, and more sparse than data from laser and optical scanners. Most reconstruction methods that work with such data create a polygonal model from the three dimensional data. Techniques that are used to perform the reconstruction include Alpha Shapes [11], Crusts [1], and region growing algorithms. Noise from data acquisition becomes quite apparent in the resulting model because these methods assume dense and precise data. In the case of volumes carved from multiple viewpoints, the reconstructed models remain in the volumetric domain, and thus have artifacts due to the voxel discretization.

Currently, models in popular use in the entertainment industry (animation and gaming applications), video and image editing, and computer graphics research come from dense laser scans or medical scans, not from vision-based techniques. We believe that this is not because vision-based techniques generate poor three dimensional data, but rather, that the reconstruction of vision-based data has not generated detailed or practical representations. Most rendering, modeling, and animation algorithms as well as hardware speed ups apply only to polygonal models. In order for vision-based models to be more widely used, a representation that is free of noise and easily convertible to a polygonal description must be obtained. We have developed such a representation to address the problem of constructing detailed, smooth, and polygonizable models from vision-based data sets.

Some of the most popular surface reconstruction approaches are based on image processing, computational geometry, and algebraic data fitting. Methods based on computational geometry assume precise and dense data in that they generate polygonal models whose vertices consist of the original data points. Similarly, image processing techniques, such as region growing, rely on dense data to define 3D structure. Approaches based

on algebraic data fitting are limited in their ability to reconstruct complex models. Our approach addresses limitations in these categories of surface reconstruction by generating approximating surfaces and by using radial basis functions centered at data points to overcome the complexity limitation.

The new approach presented in this paper constructs a three dimensional implicit function from range data. We use an implicit function representation because implicit functions can smoothly interpolate the surface where there is little or no data, the surfaces are inherently manifold, compact, smooth and seamless, and they can either approximate or interpolate the data. In addition, implicit surfaces can accurately model soft and organic objects and can easily be converted to a polygonal model by isosurface extraction. They are well-suited for operations such as collision detection, morphing, blending, and modeling with constructive solid geometry because they use a single analytical function, as opposed to a piecewise formulation such as a polygonal model. In effect, our approach retains the flexibility of an analytical representation but can also produce the widely used polygonal representation.

We construct an implicit surface from the data set using three dimensional regularization. This approach is based on the variational implicit surfaces of Turk and O'Brien [27]. Our implicit function consists of a summation of weighted radial basis functions that are placed at surface, interior, and exterior constraint points defined by the data set. The weights of the basis functions are determined by solving a linear system of equations. If so desired, we can approximate the data set by relaxing the linear system. The ability to choose whether to approximate or interpolate the data is especially advantageous in the presence of noise. Surface detail and smoothness is obtained by using radial basis functions that achieve multiple orders of smoothness.

Our main contributions are: (1) introducing the use of variational implicit surfaces for surface reconstruction from range data, (2) the application of a new radial basis function which achieves multiple orders of smoothness, (3) enhancement of fine detail and sharp features that are often smoothed-over by the variational implicit surfaces, (4) construction of approximating, rather than interpolating surfaces to overcome noisy data, and (5) development of validation techniques to guide the selection of parameters that control the smoothness of the surface versus the fitness to the data.

The remainder of the paper is organized as follows: in Section 2, we review related work in surface reconstruction. In Section 3, we show how a priori knowledge about the vision-based data set is used in reconstructing the surface. We describe the variational implicit surface representation in Section 4. In Section 5, we describe our approach to constructing approximating surfaces. In Section 6, we introduce a radial basis function that achieves multiple orders of smoothness. Results from synthetic range images and from real space carved data sets are shown in Section 7.

2 Related Work

The large number of published methods for performing surface reconstruction makes it nearly impossible to perform a comprehensive survey. Instead, we describe some of the more popular aproaches, with a bias towards those more closely related to our own approach. The methods we describe include region growing techniques, algorithms based on computational geometry, algebraic fitting methods, and surface regularization. Techniques based on computational geometry and region growing methods require precise and dense data. Consequently, these methods may fail when applied to non-uniform and noisy data. Methods based on algebraic fitting are limited by the computational complexity required to reconstruct complex models because simple, low order polynomials are insufficient for representing complex objects.

Examples of region growing techniques include Hoppe's work on surface reconstruction [14] and Lee, Tang and Medioni's work on tensor voting [19, 23]. In Hoppe's approach, a plane is fit to a neighborhood around each data point, and these planes provide an estimate of the surface normal for each point. A graph is then constructed in which each data point is a node, and the nodes are connected by arcs if the respective points are in each other's neighborhood. Hoppe assigns a cost to each arc corresponding to the difference in the estimated surface normals of the respective data points, and then surface normals are propagated by traversing the graph as a minimal spanning tree, starting at a seed point. Lee and Medioni's tensor voting method is a similar approach in that neighboring points are used to estimate the orientations of data points. The tensor is the covariance matrix of the normal vectors of a neighborhood of points. Lee and Medioni decompose the tensor into three basis that correspond to three features: planar surfaces, edges where two surfaces meet, and

points where three or more surfaces intersect. Each data point votes for the orientation of other points in its neighborhood using its tensor field. In [23], the surface is reconstructed by growing planar, edge, and point features until they encounter a neighboring feature. Both methods described above are sensitive to noise in the data points because they rely on good estimates for the normal vector at each data point.

Several algorithms based on computational geometry construct a collection of simplexes that form the shape or surface from a set of unorganized points. These methods exactly interpolate the data — the vertices of the simplexes consist of the given data points. A consequence of this is that noise and aliasing in the data become embedded in the reconstructed surface. Of such methods, three of the most successful are Alpha Shapes [11], the Crust algorithm [1], and the Ball-Pivoting algorithm [5].

The Alpha Shapes technique consists of three steps: (1) triangulation of the point set using Delaunay triangulation, (2) selection of an alpha radius, and (3) identification of the simplicial complexes that are to be included in the reconstructed shape. The shape is carved out by removing edges, triangles, and tetrahedrons whose circumscribing sphere is larger than the alpha ball. The output of the Alpha Shapes technique is not a manifold surface, but a collection of simplicial complexes which approximate the 3D shape. The interior of the shape consists of tetrahedrons. The shape boundary may be considered the surface. Selection of bounding complexes to obtain a manifold surface is a non-trivial task.

The Crust algorithm, also called Voronoi filtering, consists of four steps: (1) Delaunay triangulation of the point set, (2) addition of Voronoi vertices, (3) retriangulation, and (4) identification of the simplicial complexes to be included in the reconstructed surface. Voronoi vertices that approximate the medial axis of the shape are retriangulated along with the original set of points. The resulting triangulation distinguishes triangles that are part of the object surface from those that are on the interior because interior triangles have a Voronoi vertex as one of their vertices. Both the Alpha Shapes and Crust algorithms need no other information than the locations of the data points and perform well on dense and precise data sets. The object model that these approaches generate, however, consists of simplexes which occur close to the surface. The collection of simplexes is not a manifold surface, and extraction of such a surface is a non-trivial post-processing task.

The Ball-Pivoting algorithm is a related method that avoids the creation of overlapping triangles and non-manifold junctions by growing a mesh from an initial seed triangle that is correctly oriented. Starting with the seed triangle, a ball of specified radius is pivoted across edges of each triangle bounding the growing mesh. If the pivoted ball hits vertices that are not yet part of the mesh, a new triangle is instantiated and added to the growing mesh. Triangles whose edges are all adjacent to other triangles of the mesh are no longer bounding triangles.

All three algorithms described above depend on dense and precise data, and as a result, any noise inherent in the data becomes embedded in the reconstructed surface. In Figure 1, the Crust algorithm is applied to real range data obtained from the generalized voxel coloring method of [8]. Although, the general shape of the toy dinosaur is recognizable, the surface is rough due to the noisy nature of the real range data.

Algebraic methods avoid creating noisy surfaces by fitting a smooth function to the data points, and by not requiring that the function pass through all data points. The reconstructed surface may consist of a single global function or many functions that are pieced together. Two examples of reconstruction by global algebraic fitting are the works of Taubin [24, 25], and Gotsman and Keren [15, 16]. Taubin fits a polynomial implicit function to a point sets by attempting to minimize the distance between the given point set and the implicit surface. He points out that calculating Euclidean distances for implicit functions requires an iterative process because implicit functions are not often Euclidean distance functions. Consequently, it is necessary to formulate an approximation to the Euclidean distance. In [24], Taubin develops a first order approximation of the Euclidean distance and improves the approximation in [25]. Gotsman and Keren's approach is to create parameterized families of polynomials that satisfy certain desirable properties, such as fitness to the data or continuity preservation. The aim of their work is to find an analytical parameterization of a sub-family of polynomials that already satisfy desirable properties. This family must be as large as possible so that it can include as many functions as possible. This technique leads to an over-representation of the subset, in that the resulting polynomial will often have more coefficients for which to solve than the simpler polynomials included in the subset, thus requiring additional computation. Their examples include simple 2D and 3D shapes such as stars, pentagons, and cubes fitted to polynomials of four, five, and six degrees. Their most

Fig. 1. The data set is a space carved volume of a toy dinosaur. The reconstruction was obtained from the Crust algorithm.

impressive result is fitting an eight degree polynomial to the silhouette of a violin. The primary limitation of global algebraic methods is their inability to reconstruct complex models. These methods become too computationally intensive for the high degree polynomials that are necessary to represent complex objects.

In [4], Bajaj overcomes the complexity limitation by constructing piecewise polynomial patches (called A-patches) that combine to form one surface. There are two primary steps in the algorithm: (1) dividing the data set into tetrahedrons, and (2) obtaining the coefficients of the Bernstein polynomial that is fitted to the data points within each tetrahedron. Delaunay triangulation divides the data points into groups delineated by tetrahedrons. Vertices and interior points of a tetrahedron form the control points of the patch, and then an A-patch is formed by fitting a Bernstein polynomial to the data points within each tetrahedron. If the error between the data points and the A-patch is too large, the tetrahedron is divided into a series of smaller interior tetrahedra and patches are created to each new tetrahedra. C^1 continuity is preserved between patches by forcing several coefficients of each polynomial to be equivalent between neighboring patches. By constructing a piecewise surface, Bajaj's approach loses the compact characteristic of a global representation, and operations such as collision detection, morphing, blending, and modeling with constructive solid geometry become more difficult to perform since the representation is no longer a single analytical function.

Our approach is similar to global algebraic fitting in that we construct one global implicit function. Our method cannot be categorized as algebraic fitting, however, because the implicit function we construct is not a polynomial. Instead, it is a summation of non-polynomial basis functions whose domain is a scalar value obtained from the distance between sample points. Previous work that is most closely related to the work we present in this paper are methods based on *regularization*. Surface reconstruction from a point set is an ill-posed inverse problem because there are infinitely many surfaces which may pass through a given set of points. *Surface regularization* restricts the class of permissible surfaces to those which minimize a selected energy functional. The work of [6] and [26] are examples of regularization applied to height-field surfaces, and [12] is an example of regularization applied to parametric curves. Terzopoulos pioneered finite-differencing techniques to compute approximate derivatives used in minimizing the thin-plate energy functional of a height-field. He developed computational molecules, or masks, from the discrete formulations of the partial derivatives. Regularization is performed by iterating between coarse and fine levels in a multi-resolution hierarchy. Boult and Kender compare classes of permissible functions and discuss the use of basis functions to minimize the energy functional associated with each class. Using synthetic data, they show examples of overshooting surfaces that are often encountered in surface regularization. As exemplified by these two methods, many approaches based on surface regularization are restricted to height fields because surface derivatives are required in the process of regularization. Derivatives with respect to the major axis are naturally defined for height fields.

In [12], Fang and Gossard reconstruct piecewise continuous parametric curves. The advantage of parametric curves and surfaces over height-fields is the ability to represent closed curves and surfaces. Each curve in their piecewise reconstruction minimizes a combination of first, second, and third order energies. Unlike the examples above, the derivative of the curve in this method is evaluated with respect to the parametric variable. Each curve is formulated as a summation of weighted basis functions. Fang and Gossard show examples using Hermite basis. Their examples are synthetic 2D curves, although their method can be applied to higher dimensions. The approach we present in this paper has similar elements. We also use basis functions to reconstruct a closed surface which minimizes a combination of first, second, and third order energies.

We differ from the previous work in that we reconstruct complex 3D objects using a single implicit function; we perform volumetric rather than surface regularization; and we use energy-minimizing basis functions as primitives.

3 Constraint Specification

In this section, we introduce the a priori knowledge we have of vision-based data sets and how this information can be used to reconstruct a surface. Later we will formalize our surface reconstruction algorithm and discuss its relationship to volumetric regularization.

The computer vision community has developed many methods to acquire 3D positional information from photographic images taken by cameras. These techniques include depth from a pair of stereo images, shape from shading, space carving, and structure from motion. The goal of all these methods is to determine a collection of 3D points that lie on a given object's surface. When such a collection of points is acquired using cameras, the camera position and direction provides additional information that can be used for surface reconstruction. In particular, if a point that can be seen from a particular camera is found to be on the surface of an object, there are no surfaces between the camera and the point. This is shown in Figure 2. We call the region between the camera and the surface point *free space*. Some space carving approaches to surface reconstruction make use of this information [9]. We can use this a priori knowledge about the object surface locations and the free space to define constraints that lie on or outside of the object. This is shown in the lower panel in Figure 2, in which points on the surface are shown as asterisks and points outside the surface are marked with dashes. The *exterior* constraints are those locations where we want our implicit function to be negative, and the *surface* constraints are where the implicit function should evaluate to zero. Later, we will give more details on how these constraints are used in our surface reconstruction algorithm. Note that we do not have any knowledge about the space behind the surface locations with respect to one camera position and direction. However, if we have images and surface locations from viewpoints surrounding the object, we can completely define the *existence* space - surface, exterior, and interior - of the object. Suppose we have range images and camera positions from viewpoints on a sphere around the object. The interior of the object is known by virtue of surface enclosure.

In this paper we demonstrate our reconstruction technique with models both from synthetic range images and actual data created using voxel coloring [21] [8]. We use this data to define exterior and surface constraints as described above in order to guide the construction of an implicit function. In the case of voxel coloring, the initial data sets are dense. For example, the toy dinosaur used in the Crust reconstruction shown in Figure 1 consists of 19,641 surface voxels. Notice that the data set is also highly discretized as shown by the Crust reconstruction. As we will explain in the following sections, the computational complexity of our approach prohibits the use of the entire range data set. Instead, we uniformly sample the data set to reduce it to less than one-third of the original size. We show in our results that this reduced data set is sufficient to generate detailed surfaces using our reconstruction algorithm. Using the entire data set is not only computationally expensive, but may also result in overfitting (or overshoots), which commonly occurs in algebraic fitting when a function is forced to interpolate all data points. In Section 7.4.2, we describe in more detail the real range data sets and how the above method for constraint specification is applied to it.

In the next section we introduce variational implicit surfaces. These surfaces are created by regularizing the volume in which the surface, exterior, and interior constraints are defined.

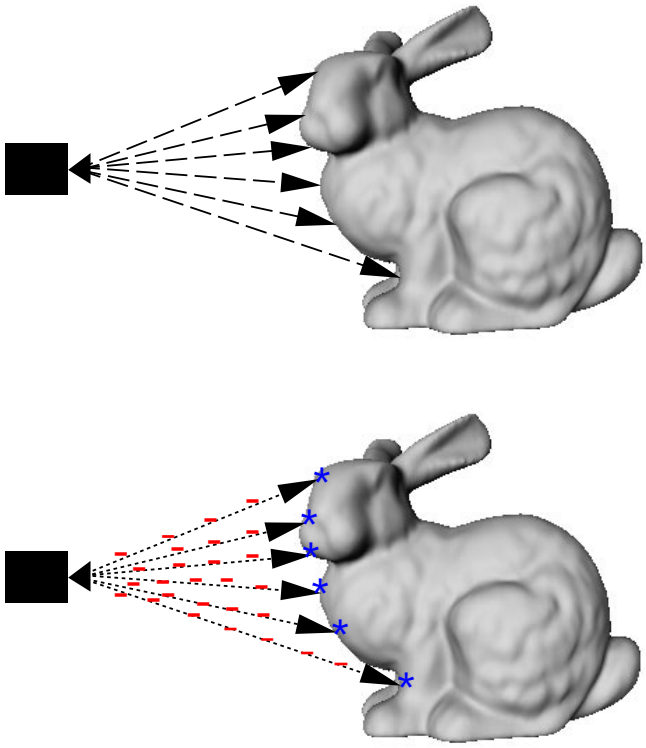

Fig. 2. The upper panel shows the free space carved out by rays projecting from the camera to the object surface. The lower panel shows surface (*) and exterior (-) constraints defined by the free space.

4 Variational Implicit Surfaces

The surface reconstruction technique that we present in this paper is an extension of the variational implicit surfaces of [27]. This 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, however, the energy functional is defined in R^3 rather than R^2. Hence, the functional does **not** act on the space of surfaces, but rather, on the space of 3D functions. In [27], the functional that was minimized is the thin-plate energy in 3D. The total thin-plate energy for this function is calculated by integrating the square of the second partial derivative of the function over the region of interest, Ω, where Ω is in R^3:

$$E = \int_{\Omega} \|f''\|^2 ds \tag{1}$$

Turk and O'Brien used the following radially symmetric basis function which inherently minimizes the above energy in 3D:

$$\phi(r) = |r|^3 \tag{2}$$

r is the distance to the center of the basis function. In this paper we will make use of other basis functions, but the basic formulation of our implicit functions is the same. We will return later to the question of which basis function to use. Regardless of which is used, each basis function is centered at each constraint point, and the resulting implicit formulation is the summation of weighted radial basis functions and a polynomial:

$$f(\vec{x}) = \sum_{i=1}^{n} w_i \phi(\vec{x} - \vec{c_i}) + P(\vec{x}) \tag{3}$$

In the above equation, n is the number of constraint points; $\vec{c_i}$ are the locations of the constraint points corresponding to the centers of the radial basis; and w_i are the weights for the basis functions. The constraints may be surface, interior, or exterior points as defined in the previous section. The polynomial term spans the

null space of the basis function. In the case of the thin-plate energy, the polynomial term consists of linear and constant terms because the thin-plate energy consists of second order derivatives ($P(\vec{x}) = p_0 + \vec{p_d} \cdot \vec{x}$ where d is the dimension of $\vec{x}$). The unique implicit function is found by solving for the weights of the radial basis functions and the coefficients of the null space.

The unknowns, w_i and the coefficients of $P(\vec{x})$, are solved by constructing a linear system. Each constraint, $\vec{c_i}$ can be applied to Equation 3 to form one equation of the system. 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. All exterior constraints are placed at the same distance away from the surface constraints towards the camera viewpoints and are assigned a function value of -1.0. All interior constraints are assigned a function value of 1.0. The basis function, $\phi(r)$, can be evaluated for $r = |\vec{c_i} - \vec{c_j}|$.

The result is a system of equations that is linear with respect to the unknowns. The linear system can be formulated as a matrix equation, $Mw = v$, where v is an array of the function values at each constraint point ($v = 0$ for surface points), w is an array of the unknown weights, and M is a matrix which consists of the evaluation of the basis function at the Euclidean distance between each pair of constraints.

$$Mw = v \tag{4}$$

$$\begin{bmatrix} \phi(|\vec{c_1} - \vec{c_1}|) & \cdots & \phi(|\vec{c_1} - \vec{c_n}|) & 1 & \vec{c_1} \\ \vdots & & \vdots & 1 & \vdots \\ \phi(|\vec{c_n} - \vec{c_1}|) & \cdots & \phi(|\vec{c_n} - \vec{c_n}|) & 1 & \vec{c_n} \\ 1 & 1 & 1 & 0 & 0 \\ \vec{c_1} & \cdots & \vec{c_n} & 0 & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_n \\ p_0 \\ \vec{p_d} \end{bmatrix} = \begin{bmatrix} f(\vec{c_1}) \\ \vdots \\ f(\vec{c_n}) \\ 0 \\ 0 \end{bmatrix} \tag{5}$$

For surfaces, constraints are specified by 3D coordinates, $\vec{c} = (c_x, c_y, c_z)$, and the polynomial that spans the null space for the thin-plate basis function consists of constant and linear terms, $P(\vec{c}) = p_0 + p_1 c_x + p_2 c_y + p_3 c_z$. The basis function $\phi(r) = |r|^3$ tends toward zero as the distance between constraints approaches zero, and it tends toward infinite as the distance approaches infinite. The above matrix is symmetric, and all elements of the diagonal are zero because the distance between a constraint point and itself is exactly zero. The system matrix is dense, and hence, sparse matrix techniques, such as biconjugate gradient descent, do not reduce the time to obtain a solution. Instead, the system can be solved using LU decomposition. Once the solution to the unknown weights are found, the 3D implicit function is completely defined by Equation 3. The implicit surface is a level-set of the 3D implicit function where it evaluates to zero. A polygonal representation of the implicit surface may be obtained by iso-surface extraction using *Marching Cubes* [20].

In the *unorganized points problem*, the interior and exterior of a point set is not defined, and the orientation of the surface reconstructed from such a point set is not known a priori. Turk and O'Brien restrict the surface orientation by pairing each surface constraint with a *normal constraint* that is interior to the surface and such constraint points are given a function value of one. For implicit surfaces constructed from a polygonal representation of the object, the constraints are defined by the vertices of the polygons, and normal constraints are defined by vertex normals. Radial basis functions are centered at each surface and normal constraint locations. In the case of vision-based data sets such as those considered in this paper, there are no surface vertices or normals to be used as constraints. Instead, we specify surface, exterior, and interior constraints in the manner described in Section 3, thus defining the orientation of the surface.

Figure 3 is an example of a surface reconstructed from a vision-based data set using the variational technique of Turk and O'Brien using the basis function $\phi(r) = |r|^3$. The figure shows a front view flanked by two side views of the reconstruction of the toy dinosaur. The constraints for this reconstruction are from range images using the ray-casting method described in Section 3. There were 3000 surface constraint points, 264 exterior constraints, and 100 interior constraints specified. The resulting surface exactly interpolates the constraint points, but overshoots are apparent between the arms and face which are fused, as well as between the feet. These overshoots are the cause of exact interpolation which forces the surface to pass through constraint

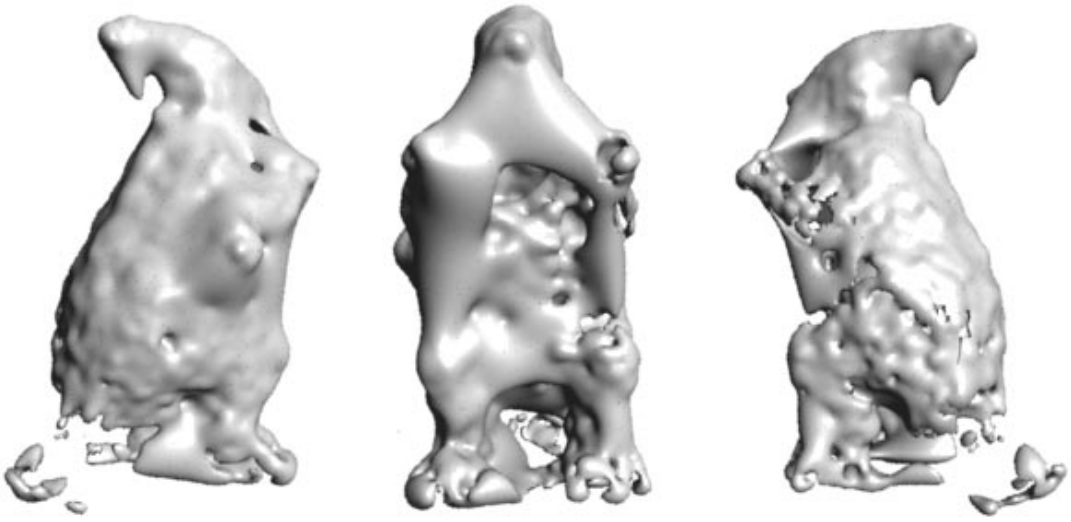

Fig. 3. The data set consists of 3000 surface, 100 interior, and 264 exterior constraints, sampled from a space carved volume of a toy dinosaur. The reconstruction is a variational implicit surface generated using the basis, $\phi(r) = |r|^3$. The reconstruction exactly interpolates the constraint points.

points. In addition, there are cavities which penetrate the tail and form small pockets within the surface. This reconstruction is evidence that exact interpolation algorithms are sensitive to noise in the data which is especially prevalent in vision-based data sets. In this paper we will improve upon these results in several ways. In the next section, we show that better results may be obtained by introducing a regularizing parameter that allows the surface to pass close to, rather than exactly through, the constraint points. A later section will demonstrate that using different basis functions also improves upon the surfaces that are created.

5 Approximation vs. Interpolation

Scattered data interpolation is the process of estimating previously unknown data values using neighboring data values that are known. In the case of surface reconstruction, the surface passes exactly through the known data points and is interpolated between the data points. Data interpolation is appropriate when the data values are precise. In the case of vision-based data, however, there is some uncertainty in the validity of the data points. Using data interpolation to construct the surface is no longer ideal because the surface may not actually pass exactly through the given data points. This is precisely the problem with algorithms from computational geometry that generate polygonal meshes using data points as the vertices of the mesh. If the uncertainty of the data points is known, a surface that better represents the data would pass close to the data points rather than through them. Constructing such a surface is known as *data approximation.* Many vision-based techniques for capturing 3D surface points have an associated error distribution or confidence range for the data points. In this section we discuss how data approximation is achieved in our framework.

We can allow the surface to pass close to, but not necessarily through, the known data points by relaxing the constraints of the linear system. We use the formulation discussed in [13]. A simple derivation is presented therein which shows that a summation of weighted radial basis functions as given in Equation 3 is the solution to minimizing a cost functional, H of the following form:

$$H[f] = \frac{1}{\lambda}\sum_{j=1}^{n}(y_i - f(\vec{x}_i))^2 + \beta[f] \tag{6}$$

In the above equation, f is the unknown surface function, n is the number of constraint points, or observed data points; y_i are the observed values of the data points at locations $\vec{x}_i$; $\beta[f]$ is the smoothness functional, such as thin-plate; and λ is a parameter to weigh between fitness to the data points and smoothness of the

surface. λ is often called the *regularization parameter*. We use λ to specify how closely we want to approximate the data set. In our approach, constraint points are interpolated when $\lambda = 0$ and are approximated when $\lambda > 0$. In the above equation, λ is defined as a global parameter. However, λ need not be global, since it is simply a parameter which controls the trade-off between fitness to the data versus strength of the smoothness assumption. We can assign such a parameter to individual constraints, so that the trade-off is between the fitness to one particular constraint versus the fitness to all other constraints plus the strength of the smoothness assumption. We can modify Equation 6 by repositioning λ as follows, and defining it for each constraint:

$$H[f] = \sum_{j=1}^{n} \frac{1}{\lambda_i}(y_i - f(\vec{x}_i))^2 + \beta[f] \tag{7}$$

λ is a known value that can be assigned according to the noise distribution of the data acquisition technique. It is included in the system matrix of Equation 5 in the following manner:

$$\begin{bmatrix} \phi(|\vec{c_1} - \vec{c_1}|) + \lambda_1 & \cdots & \phi(|\vec{c_1} - \vec{c_n}|) & 1 & \vec{c_1} \\ \vdots & & \vdots & 1 & \vdots \\ \phi(|\vec{c_n} - \vec{c_1}|) & \cdots & \phi(|\vec{c_n} - \vec{c_n}|) + \lambda_n & 1 & \vec{c_n} \\ 1 & 1 & 1 & 0 & 0 \\ \vec{c_1} & \cdots & \vec{c_n} & 0 & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_n \\ p_0 \\ \vec{p_d} \end{bmatrix} = \begin{bmatrix} f(\vec{c_1}) \\ \vdots \\ f(\vec{c_n}) \\ 0 \\ 0 \end{bmatrix} \tag{8}$$

The ability to assign distinct λ values to individual constraints is especially important when we use exterior and interior constraints because they are added only to provide orientation to the surface but do not represent real data. A large λ value such as 2.0 is often used for exterior and interior constraints, while small values such as 0.001 is often used for surface constraints. We can also use λ as a local fitting parameter by assigning the λ value for each surface constraint based on the confidence measurement of the data point. This may not be possible, however, if individual confidence measurements are not available, or if the confidence measurements themselves are imprecise. In practice, we have found that λ works well as a semi-global fitting parameter, where one λ value is used for all surface constraints, and another for all interior and exterior constraints. Figure 4 shows the result of applying different λ values on the same data set. As λ approaches zero, the surface becomes rougher because it is constrained to pass closer to the data points. At $\lambda = 0$, the surface interpolates the data, and overshoots are much more evident. At larger values of λ, the reconstructed model is smoother and approaches an amorphous bubble. We have found that a λ value of 0.001 for surface constraints generates visually pleasing reconstructions. In Section 7 we validate this choice of λ using measures of fitness and curvature.

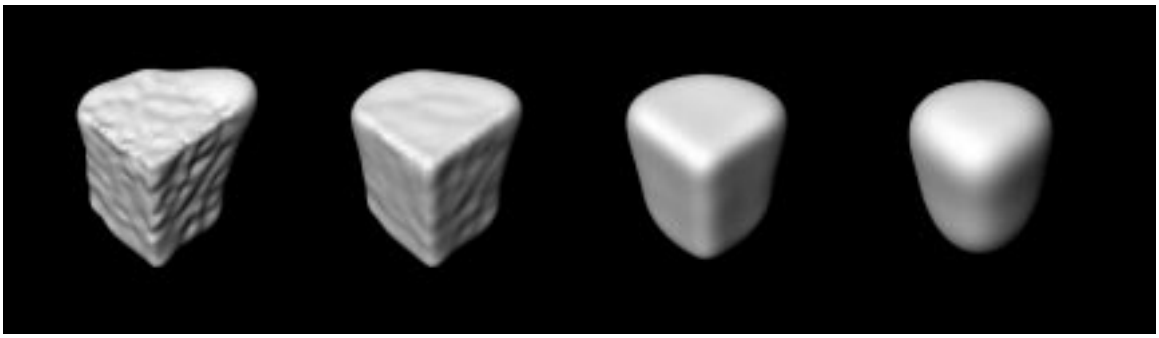

Fig. 4. The data set is from a synthetic range image of a corner of a cube. λ is varied from 0 at the left to 2.0 at the right. $\lambda = 0.001$ for the second reconstruction from the left.

Figure 5 shows the reconstruction of the toy dinosaur using the same 3000 surface, 200 exterior, and 100 interior constraint as that of Figure 3 but with λ set to 0.001. The surface is much smoother, the overshoots are less apparent, and there are fewer protruding bumps and fewer small pockets embedded in the surface. Unfortunately, the toy dinosaur's features are blobby and amorphous, especially at the feet and hands. Distinct limbs, such as the feet and tail, are fused together. It is apparent from this result that the radial basis function

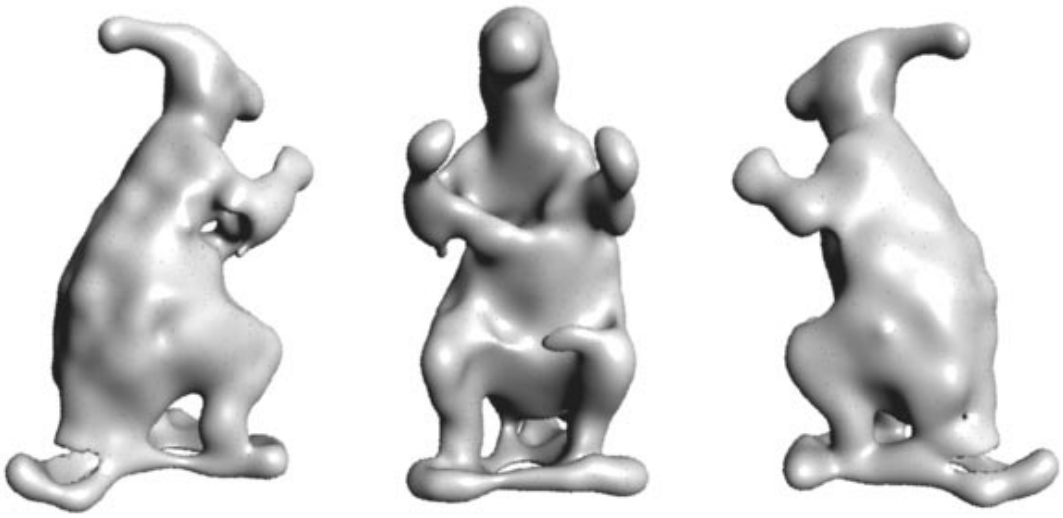

Fig. 5. The reconstruction is a variational implicit surface generated using the basis, $\phi(r) = |r|^3$ and $\lambda = 0.001$ for surface constraints. The data set is the same as that of Figure 3.

used by Turk and O'Brien generates models which are too blobby. In addition, the dense matrix produced by the thin-plate radial basis function is computationally intensive to solve. Computational time increases significantly as more constraints are specified because the complexity of LU decomposition is $O(n^3)$, where n is the number of constraints. The coefficients of the implicit function for the model in Figure 5 were solved in 36.7 minutes on an SGI Origin with 195 MHz MIPS R10000 processor. In the next section, we explore the use of a radial basis function that minimizes multiple orders of smoothness, not just thin-plate energy. This basis function turns out to be amenable to sparse matrix solutions (enabling the specification of many more constraints) and is significantly less blobby in nature.

6 A Radial Basis Function for Multiple Orders of Smoothness

The results of Figures 1 and 5 show that a balance is needed between a tightly fitting, or *shrink-wrapped*, surface, and a smooth surface. A tightly fitting surface separates the features of the model but is prone to jagged artifacts. For example, the Crust reconstruction, shown in Figure 1, is an exact fit to the data with no smoothness constraint. On the other hand, a smooth surface may become too blobby as in Figure 5, which shows that minimizing the thin-plate energy alone is not sufficient to produce a surface that separates features well and is locally detailed.

In [7], Chen and Suter derive the radial basis functions for the family of Laplacian splines, of which the first, second, and third order energy-minimizing splines are members. Thin-plate energy is equivalent to second order energy, and membrane to first order energy. For the first three dimensions, the basis are comprised of r^k, $r^k log|r|$, exponential, and Bessel function terms. r is the distance from the center of the radially symmetric basis. The value of k depends on the dimension and order of smoothness. Turk and O'Brien use $\phi(r) = |r|^2 log|r|$ for two dimensional thin-plate interpolation, and $\phi(r) = |r|^3$ for three dimensional thin-plate interpolation. One dimensional plots of these radial basis functions are shown in Figure 6.

The plots show that the functions exhibit global influence because the value of the function tends toward infinite as the distance from its center increases. The system matrix, which consists of the evaluation of the basis function at distances between pairs of constraints, is dense because constraint points are sparse and uniformly spread across the region of interest.

Surprisingly, a more complex radial basis function has compact support, yielding a better conditioned system matrix. In [22], Suter and Chen used basis functions that minimize multiple orders of smoothness (beyond the first and second order) to reconstruct human cardiac motion. They found that a model minimizing third and

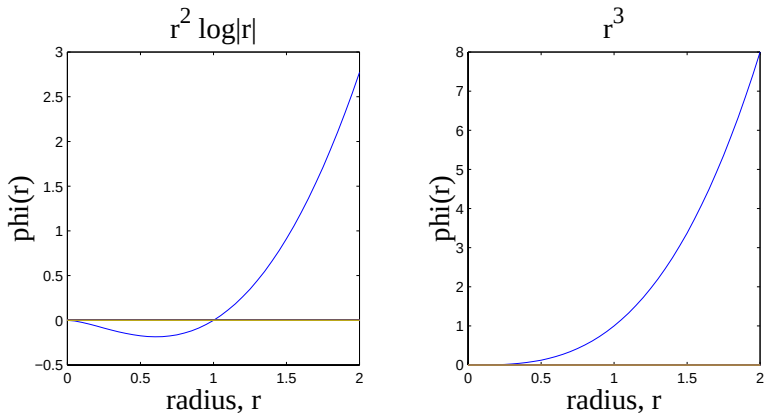

Fig. 6. Cross section of radially symmetric basis functions for thin-plate spline in two and three dimensions.

fourth order energy resulted in the smallest RMS error. They concluded that basis which minimize more than just the first and/or second order energy generate more accurate reconstructions. In particular, as the space dimension increases, the order of continuity of the thin-plate spline at data points decrease. For example, Suter and Chen show that in three dimensions, the thin-plate spline basis has discontinous first order derivative at the data points. We chose to use a basis that achieves first, second, and third order smoothness because, unlike motion, object surfaces may contain sharp features which are C^1 discontinuous. The additional third order, however, enhances smoothness since it's order of continuity does not break down in three dimensions. Chen and Suter present a derivation for such a basis in [7] by calculating the inverse Fourier transform of the Laplacian operator associated with the partial differential equation for minimizing a combination of first, second, and third order energy. The basis function as given in [7] for obtaining a combination of first, second, and third order smoothness in three dimensions is as follows:

$$\phi(r) = \frac{1}{4\pi\delta^2 r}(1 + C_1 e^{-\sqrt{v}r} - C_2 e^{-\sqrt{w}r}) \tag{9}$$

$$v = \frac{1 + \sqrt{1 - 4\tau^2\delta^2}}{2\tau^2} \tag{10}$$

$$w = \frac{1 - \sqrt{1 - 4\tau^2\delta^2}}{2\tau^2} \tag{11}$$

$$C_1 = \frac{w}{v - w} \tag{12}$$

$$C_2 = \frac{v}{v - w} \tag{13}$$

r is the distance from the center of the radial basis function. δ controls the amount of first order smoothness, and τ controls the amount of third order smoothness. The amount of second order smoothness is inherently controled by the balance of δ and τ. δ and τ are the only free parameters in defining the basis function. Figures 7 and 9 are one dimensional plots of the above function. Unlike the plot for $\phi(r) = |r|^3$, these plots show that the value of the basis function tends toward zero as the distance from its center increases.

In Figure 7, δ is increased from a value of 10 to 40 while τ is kept at 0.01. Note how the radius of influence decreases as δ increases. In other words, the basis tends to zero faster for larger values of δ. As the radius of influence shrinks, the basis contributes less and less global influence. In the limit, the basis are simply spikes, and the resulting 3D implicit function would contain steep gradients between spiky points. A zero-valued level-set of such a function would also exhibit sharp features because the spikes are centered at constraint points, including surface constraint points where the implicit function evaluates to zero.

The reconstructions of the toy dinosaur in Figure 8 correspond to the different values of δ in the plots of Figure 7. The models were reconstructed using 3000 surface constraints, 100 interior, and 264 exterior constraints. The surface becomes increasingly pinched at the constraint points as δ is increased. Visually, the pinching appears as surface detail that is often missing in overly smooth surfaces. Despite the pinching, the surface remains globally smooth. This reconstructions are substantial improvements over those shown in Figures 3 and 5.

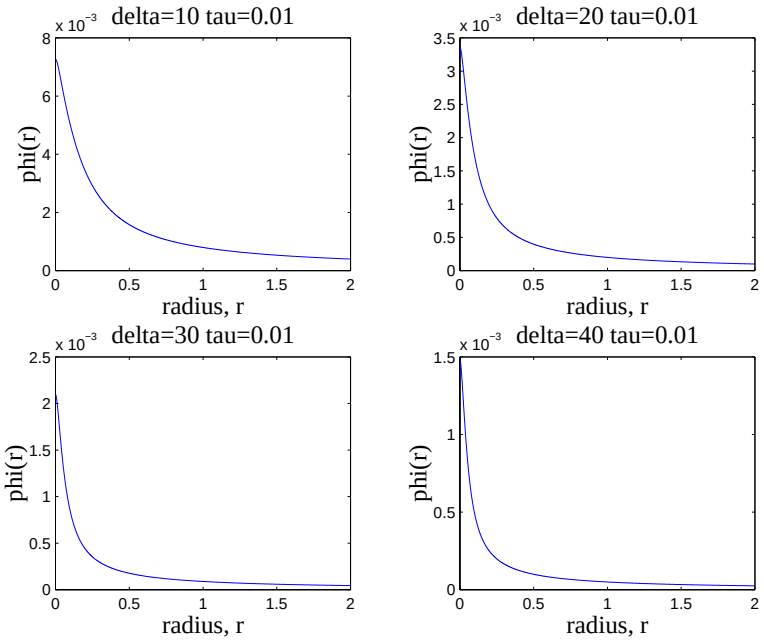

Fig. 7. Cross section of radially symmetric basis functions for a combination of first, second, and third order smoothness. δ varies from 10 to 40, $\tau = 0.01$.

In Figure 9, τ is increased from 0.01 to 0.75 while δ is maintained at 0.25. The radius of influence also shrinks. In this case, however, the value of the function near the center flattens out, instead of becoming spiky. In the limit, the basis becomes parabolic and enforces homogeneous curvature around constraint points. The resulting 3D implicit function and its iso-surface becomes increasingly smooth as τ is increased. Figure 10 shows the reconstruction of the toy dinosaur corresponding to the increasing values of τ for the same data set as that of Figure 8. In the limit, the reconstructed surface becomes amorphous, similar to those obtained from large values of λ.

The system matrix formed by Equations 9 through 13 is diagonally dominant and is especially amenable the biconjugate gradient method of solving linear equations. Timing results show that the system matrix of Equation 5 was solved in 1.7 minutes using the multi-order basis function with $\delta = 0.01$ and $\tau = 10$, while the system matrix generated for the same set of 3300 constraints using the thin-plate radial basis function required 36.7 minutes. Not only are these new results superior in terms of quality, but they also require considerably less time to create than the original formulation of Turk and O'Brien.

The range of weights used in Figures 8 and 10 for δ and τ were chosen to span the search space of possible values. Selection of the weights is constrained by Equations 10 and 11, in which both δ and τ appear under the square root. Values for δ and τ were chosen to avoid imaginary values for v and w. Figure 13 is a plot of the range of allowable values for δ and τ.

Determining the values of δ and τ that correspond to the best reconstruction of a surface is an important issue. By visual inspection, we can see that the reconstructions in Figure 8 are the more pleasing in terms of trade-off between surface detail and smoothness than those of Figure 10. The arms, legs, and tail of the dinosaur in these reconstructions are better separated, and the back more detailed than that of Figure 5 which was produced by the thin-plate radial basis function. Judging between the four reconstructions of Figure 8 is, however, more difficult. Deciding that one is better than another amongst these four is quite

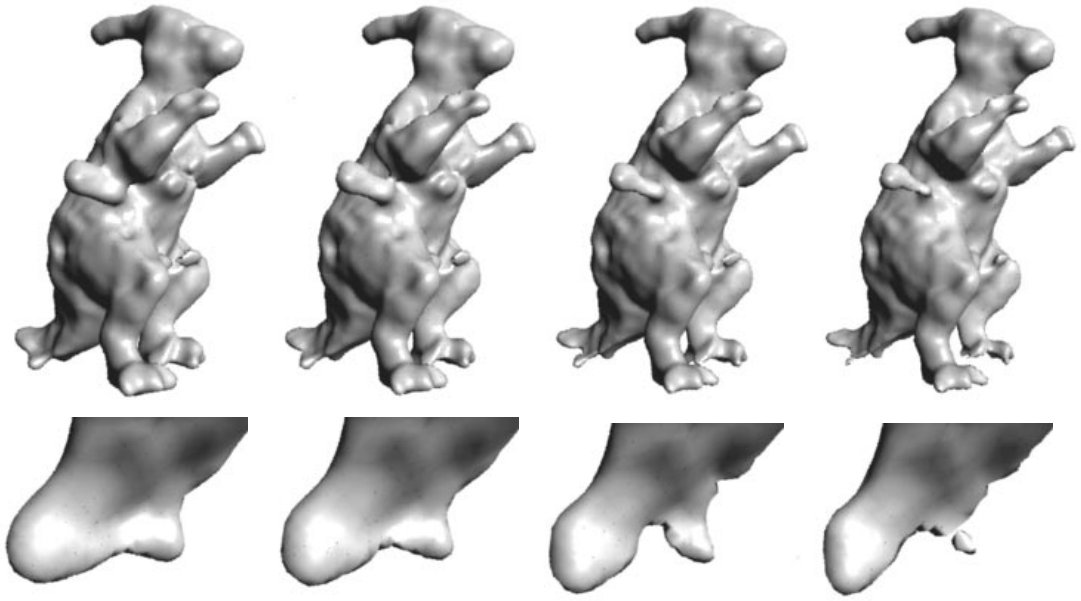

Fig. 8. From left to right, δ is 10, 20, 30, and 40, while τ is kept at 0.01. The lower panels are closeups of the dinosaur tail which show an example of the differences between the reconstructions. The data set is the same as that of Figure 3.

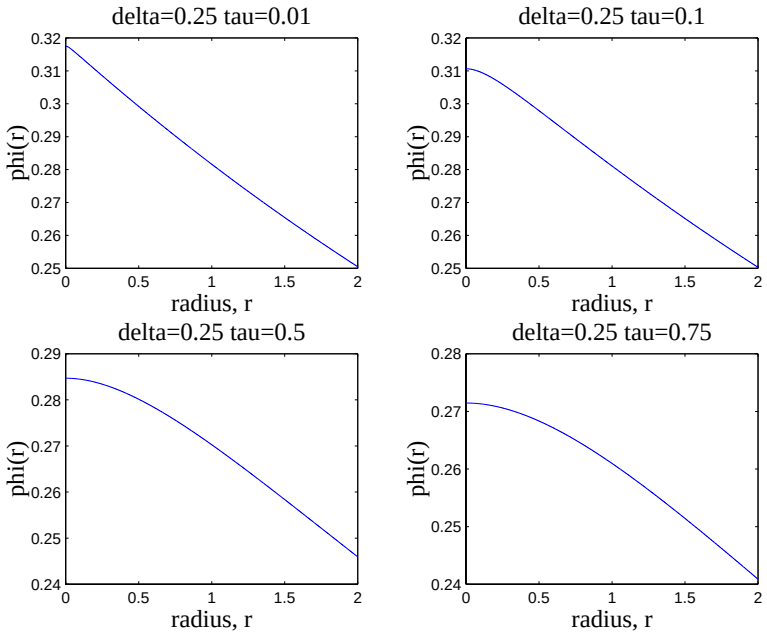

Fig. 9. Cross section of radially symmetric basis functions for a combination of first, second, and third order smoothness. $\delta = 0.25$, τ varies from 0.01 to 0.75.

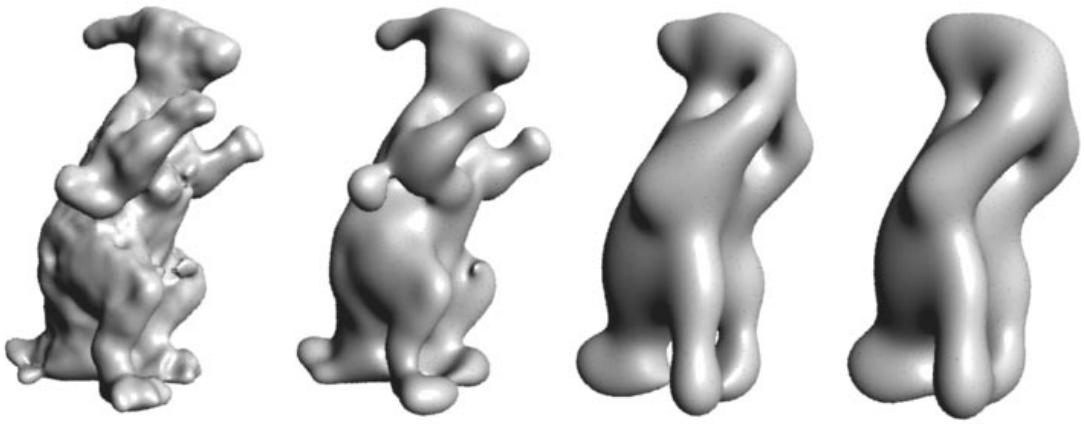

Fig. 10. From left to right, τ is 0.01, 0.1, 0.5, and 0.75, while δ is kept at 0.25. The data set is the same as that of Figure 3.

arbitrary and objective. In the next section, we discuss two methods we have used for validation and as a tool for comparison between the reconstructed models. These methods include a measure of fitness error and a measure of aggregate curvature. They are exactly the attributes that the basis functions and the parameters, λ, δ and τ, regulate.

7 Results

Figure 11 is a visual comparison of different reconstructions of the toy dinosaur. This figure shows the original data (far left), the Crust reconstruction (middle left), the implicit surface using the thin-plate basis function (middle right). At the far right of the figure is the new reconstruction using the techniques of this paper. Note that the round protrusion beneath the arm is the turn key of the actual toy dinosaur (see Figure 20 for the original images of the toy dinosaur). We show further evidence in this section, using new data sets as well as synthetic data sets, that reconstruction by volumetric regularization generates globally smooth, yet detailed, surfaces. Finally, we discuss the addition of color to the models.

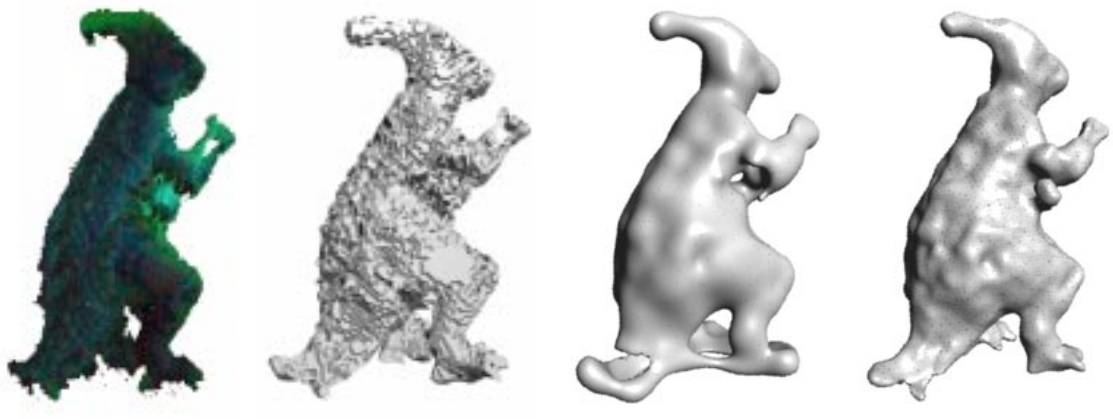

Fig. 11. From left to right: original voxel data set from space carving, Crust reconstruction from entire data set, implicit surface reconstruction using the thin-plate radial basis function, and our new implicit surface reconstruction using the multi-order radial basis function. The last two reconstructions were generated using 3000 surface, 100 interior, and 264 exterior constraints sampled from the space carved data set.

We have introduced three parameters, λ, δ and τ, in our reconstruction algorithm. We now show that for

the purpose of surface reconstruction from sparse point sets, there are a limited number of possible values for these parameters that will produce desirable results. We validate and compare the reconstructions based on visual inspection, a measure of fitness, and a measure of curvature. By delimiting the space of values for these parameters, we show that reconstruction by volumetric regularization requires minimal parameter tuning.

7.1 Selecting λ

Recall from Section 5 that λ controls the trade-off between fitness to the data points and the smoothness assumption. The following measures of fitness and curvature corroborate this fact, as well as help guide the selection of appropriate values for λ.

7.1.1 Fitness Error

We define fitness error to be the aggregate distance between the original data points and the reconstructed surface. We measure this distance by first constructing a polygonal model from the implicit function using Marching Cubes [20], and then finding the closest vertex of the polygonal model to a given data point. This vertex is an intial starting point on the surface from which we can then search for even closer surface locations to the given data point. We find closer surface points crawling along the surface in small increments until a small increment in any four directions along the surface does not yield a location that is closer to the constraint. When this stopping location is found, the Euclidean distance is calculated and accumulated.

Figure 12 shows a plot of the total fitness error for the dinosaur reconstructions using thin-plate and multi-order basis functions at varying values of λ. As expected, small values of λ correspond to less error in data fitness. Note that the error is an accumulation of the Euclidean distance measured at all the original data points, not just the constraint points used in the reconstruction. Consequently, the error is not zero even when $\lambda = 0$, corresponding to exact interpolation. Error in fitness rises more sharply for the multi-order radial basis function as λ is increased than for the thin-plate basis function. At lower values of λ (0.003 or less), the aggregate error for both basis are comparable. The sharp rise in fitness error for the multi-order basis provides a practical upper bound of 0.003 for λ (dotted line in the plots of Figure 12 mark the location of $\lambda = 0.003$).

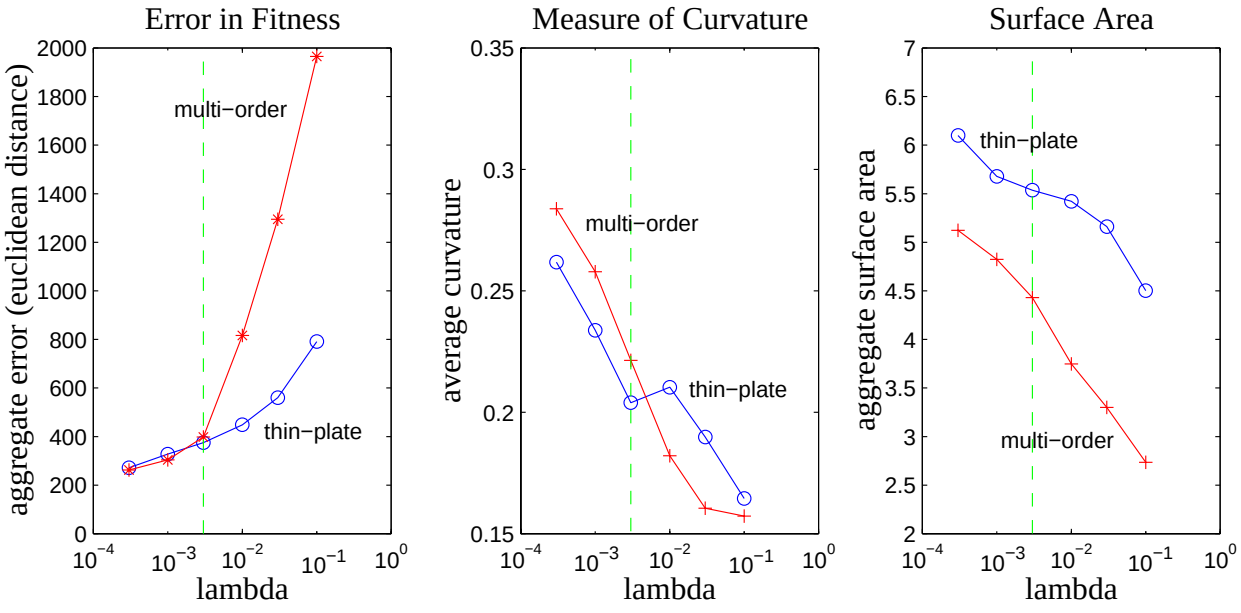

Fig. 12. The plot on the left compares the total error in fitness to the data for the dinosaur reconstructions using the thin-plate (o) and multi- order (*) radial basis functions, for λ =0.0003, 0.001, 0.003, 0.01, 0.03, and 0.1. The center plot compares the total curvature for both basis functions at the same values of λ. The plot on the right compares the total surface area for both basis functions at the same values of λ. The vertical dotted line in all three plots mark the location of $\lambda = 0.003$.

7.1.2 Measure of Curvature

The next quality measure we describe is the aggregate curvature of the surface. We measure the curvature of the surfaces using polygonal models that are extracted from the implicit function using Marching Cubes [20]. We measure curvature at each vertex of these polygonal models using a curvature approximation that was developed for the smoothing operator in [10]. This measure is based on the normal directions of triangles adjacent to each vertex and normalized by the total area of the triangles. High curvature is associated with sharp features in the surface, while low curvature is associated with overshoots and blobby surfaces.

The center plot in Figure 12 shows the total curvature for the thin-plate and multi-order radial basis functions at varying values of λ. As expected, the curvature drops at large λ values since the constraints are no longer interpolated and the influence of the smoothness model is stronger. The plot reveals that the curvature of the surface generated by the multi-order basis is higher than that generated by the thin-plate basis at values of λ less than or equal to 0.003. The thin-plate surface exhibits less curvature at λ values of 0.003 or less, corresponding to the blobbiness seen in Figure 5. When λ is greater than 0.003, however, the surface generated by the multi-order basis exhibits lower curvature than that of the thin-plate basis. This is further evidence that λ should be kept at 0.003 or less.

A measure of the surface area shown in the plot on the right in Figure 12 reveals that the thin-plate basis tends to produce larger surfaces (an indication of overshooting surfaces) across all values of λ. The multi-order basis function achieves a good balance between a tight fit and a smooth surface because they generate surfaces with high curvature and equivalent fitness error in comparison to the thin-plate basis. We have found that λ values between 0.001 and 0.003 work well in practice, and this is consistent with the data of Figure 12. All examples of our reconstruction, save for Figure 3, was created using a value of 0.001 for λ.

7.2 Selecting δ and τ

Recall from Section 6 that δ controls the amount of first order smoothness, while τ controls the amount of third order smoothness. Figures 8 and 10 show reconstructions of the toy dinosaur data set using various values of δ and τ. In selecting appropriate values for these parameters, we began with the values used in Figure 8 rather than those used in Figure 10 because the reconstructions of Figure 8 are, by visual inspection, a better reproduction of the original data set from space carving. Figure 10 shows that a δ value of 0.25 and τ values of 0.01 and higher produce reconstructions which are far too smooth and blobby. Consequently, we constrained our search space to be more along the δ axis than along the τ axis. Figure 13 shows the valid range of values for δ and τ that avoid imaginary values in Equations 10 and 11. The blue line in the graph indicates the limiting values, and the red stars are the parameter values that we tested. Within our selected search space, δ becomes a coarse adjuster, while τ is a fine adjuster for fitness and curvature. We show in the following sections that selection of the δ and τ values is not an extra burden on the user because there is a limited space of values which produce locally detailed and globally smooth surfaces with low fitness error.

7.2.1 Fitness Error

Figure 14 is a plot of the average fitness error for the toy dinosaur reconstructions using various values of δ and τ. Fitness error was measured for δ values of 1.0, 5.0, 10.0, 15.0, 20.0, 30.0, and 40.0, and τ values of 0.0005, 0.001, 0.005, 0.01, and 0.05 (different values were used for this last τ value due to the imaginary value limitation as shown in Figure 13). δ is 40.0 at the near corner of the graph and 1.0 at the leftmost corner along the δ axis. τ is 0.0005 at the near corner and 0.05 at the rightmost corner along the τ axis. The plot shows that δ has a greater influence on fitness to the data since changes in δ cause greater changes in fitness error than changes in τ. δ can be regarded as a course level adjuster, while τ is a fine level adjuster within this search space. In general, larger δ values produce a tighter fit, corroborating the fact that δ controls first order smoothness. Note, however, that at the largest value of δ in the plot ($\delta = 40.0$), changes in τ cause jumps in the fitness error. A δ value of 50.0 was found to result in even greater fitness error. This indicates that 40.0 is the upper bound for appropriate δ values for this data set. We have found by visual inspection that values of 30.0 and 40.0 for δ produced the most detailed, yet smooth surfaces. At such values, the average fitness error is below 0.5 in Euclidean distance. This corresponds exactly with the data set in that the range resolution is

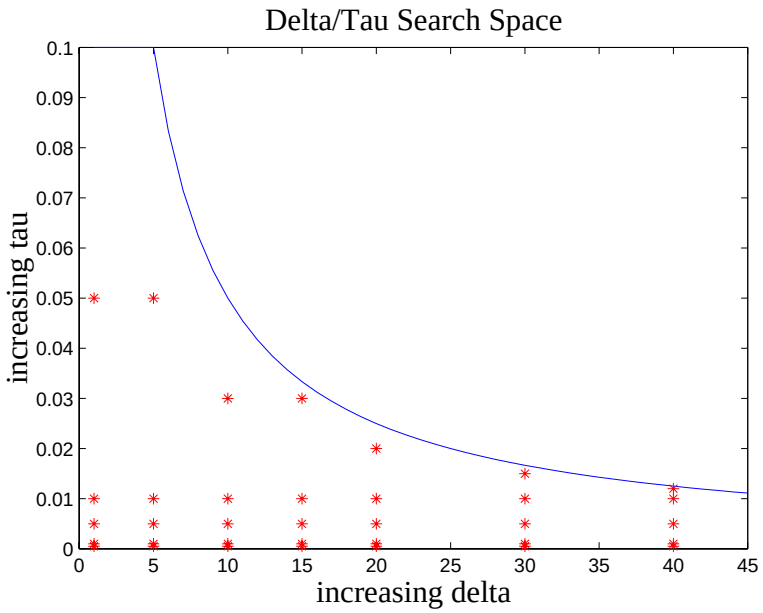

Fig. 13. Limitation placed on δ and τ. The limit is shown as a blue line, and the tested parameters are indicated by the red stars.

0.5, which means that the surface may exist anywhere within a radius of 0.25 from the data points.

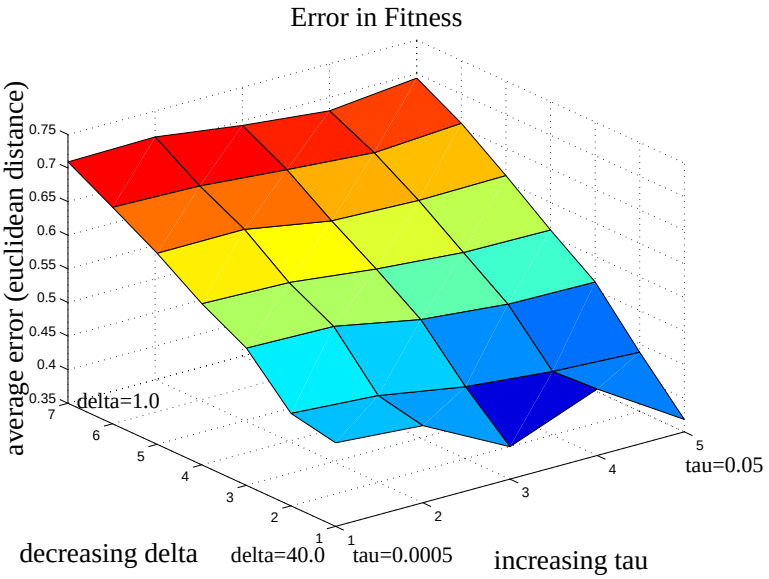

Fig. 14. Average fitness error for varying values of δ and τ. From far to near along the δ axis, $\delta = 1.0$, 5.0, 10.0, 15.0, 20.0, 30.0, and 40.0. From near to far along the τ axis, $\tau = 0.0005$, 0.001, 0.005, 0.01, and 0.05

A value of 0.01 or 0.005 for τ was found to consistently produce the smallest fitness error across all values of δ. These values correspond to the third and fourth gridlines along the τ axis.

7.2.2 Measure of Curvature

Figure 15 is a plot of the average curvature of the reconstructions using the same values of δ and τ as that of the plot in Figure 14. Although a pattern is not as apparent here as in the fitness plot, the plot supports the fact that δ controls first order smoothness, while τ controls third order smoothness. High curvature is maintained at large values of δ and small values of τ, while low curvature is prevalent at large values of τ (values at the far right side of the graph). Low curvature is associated with overshooting surfaces that tend

to be blobby. δ values of 30.0 or 40.0 (second and first grid lines along the δ axis) and τ values of 0.005 or 0.01 (third and fourth grid lines along the τ axis) generate surfaces which exhibit relatively high curvature compared to the other values.

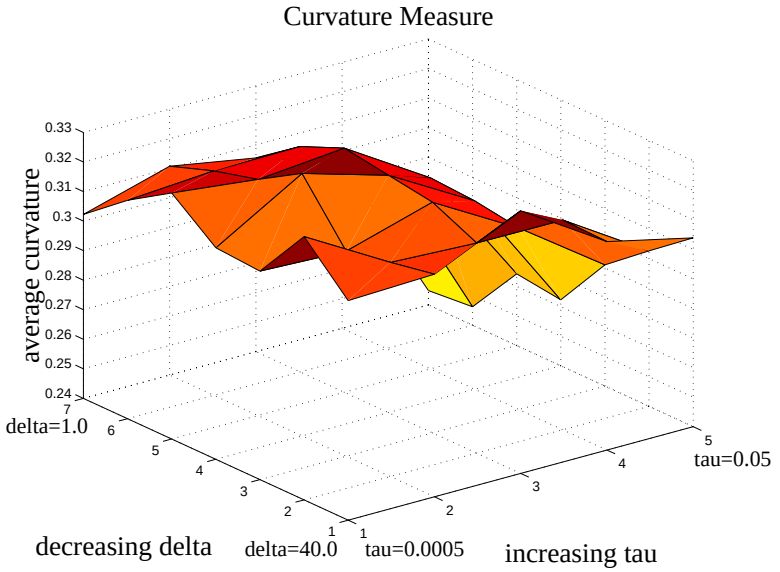

Fig. 15. Average curvature for varying values of δ and τ. From far to near along the δ axis, $\delta = 1.0$, 5.0, 10.0, 15.0, 20.0, 30.0, and 40.0. From near to far along the τ axis, $\tau = 0.0005$, 0.001, 0.005, 0.01, and 0.05

7.3 Generalizing the Parameter Values

We applied the measures of fitness and curvature to the toy dinosaur data set in order to guide selection of appropriate values for λ, δ, and τ. We have found in practice that these same values may be used with other data sets to produce desirable reconstructions. Examples of our reconstructions of other data sets are shown in Figures 17 and 19. Note that although values of 30.0 and 40.0 for δ were found to produce the most visually detailed surfaces for the toy dinosaur data set, values of 10.0 and 20.0 produced surfaces which were only slightly more blobby (see the closeup of Figure 8). By delimiting the space of useful values for λ, δ, and τ, we have shown that the use of three control parameters does not create additional burden on the user. Values of λ between 0.001 to 0.003, δ values between 10.0 to 40.0 and τ values between 0.005 to 0.01 can be used to produce locally detailed, yet globally smooth, reconstructions on a variety of data sets.

7.4 Ground Truth Comparison

The challenge in validating and judging between different surface reconstructions of a data set is due to the fact that no *ground truth* surface exists. We do not know precisely how the surface behaves between data points, and, in the case of vision-based data sets, we cannot be positive that the surface even passes through the observed data points. One way of validating a surface reconstruction algorithm is to compare the results of reconstructing a synthetic data set to the original model from which the data set was obtained. In the next sections, we show results from synthetic data as well as discuss the real range data that was used in previous sections.

7.4.1 Synthetic Range Data

We use a modified ray-tracer [17] to generate synthetic range images as one test of our reconstruction method. We used the Stanford Bunny as our test model, and created three synthetic range images from positions separated by 120 degrees around a circle surrounding the model. We used these three synthetic range images as input data to our reconstruction algorithm. For each range image, surface constraints are created by uniformly downsampling the range image to reduce the size of the data set. For each surface constraint, one exterior negative constraint is created within the free space described in Section 3. Additional

exterior constraints are defined on a sphere surrounding the bounding box of the object at a distance farther away from the object. No additional interior constraints are used because the exterior constraints are sufficient to define the orientation of the surface. The left panel of Figure 16 shows the original Stanford Bunny model consisting of 69,451 triangles, and the remaining panels show the distribution of constraints defined for the Bunny. Surface constraints are drawn as blue squares imbedded in the surface, and negative constraints are drawn as green squares. The model shown in Figure 17 was reconstructed from 2168 surface and 193 exterior constraints. Values of $\lambda = 0.001$, $\delta = 10$, and $\tau = 0.01$ were used to reconstruct the surface. The different views of the reconstructed bunny in Figure 17 show that our model is quite similar to the ground truth. Our reconstruction method produces plausible surfaces even in locations where the data is quite sparse. Note how the model is closed on the top and bottom of the Bunny even though few constraint points were placed in those locations as shown in the third and fourth panels of Figure 16. The model is closed at these places due to the inherently manifold nature of implicit surfaces, and it is smooth at these locations by virtue of minimizing the cost functional.

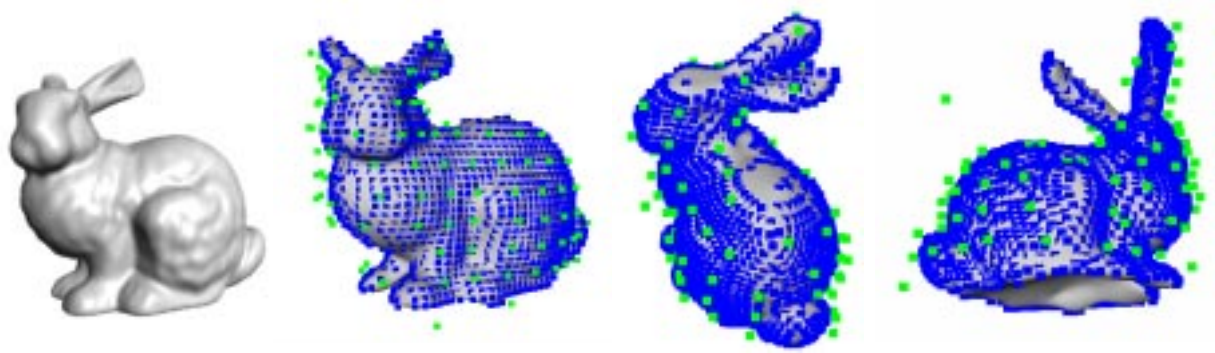

Fig. 16. The original Stanford bunny consisting of 69,451 triangles is shown on the left. On the right, the constraint points are shown embedded in the reconstructed surface. Blue squares are surface constraints. Green squares are negative exterior constraints.

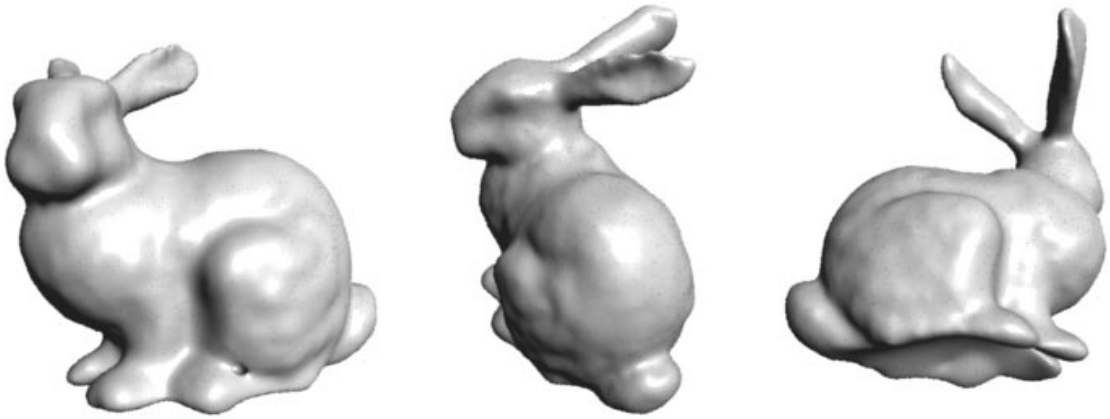

Fig. 17. Reconstruction of the Stanford bunny from synthetic range images using a radial basis function that achieves first, second, and third order smoothness.

7.4.2 Real Volume-Carved Data

Synthetic data does not, of course, have the noisy characteristic of real data. We now describe the real space carved data that we use and how we define the surface, interior, and exterior constraints. We used two datasets from vision data of real objects, a toy dinosaur (from Steve Seitz [21]) and a broccoli stalk (courtesy of Greg Slabaugh). Both sets of data were obtained by taking about 20 images approximately on a circle around

each object. Volumetric surfaces were then constructed using the generalized voxel coloring algorithm [8]. The space is carved by splatting each voxel towards each calibrated 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 opacity value. The data consists of red, green, and blue channels for each voxel. Non-empty voxels represent the presence of a surface, as deduced by the voxel coloring algorithm.

In defining surface constraints, we use the volume as a binary representation in which non-empty voxels are surface locations. We apply the technique described in Section 3 to define the surface, exterior, and interior constraints to reconstruct the object. We do not use the entire set of surface voxels because the system matrix would become too large (19,641 surface voxels for the dinosaur data set), and the reconstructed surface would over fit the data, resulting in overshoots. To obtain a subset of these surface voxels we sample the volume by randomly selecting voxels. Each time a voxel is selected, the neighboring voxels within a radius of influence are eliminated from possible selection in the next round. The elimination process prevents clusters of closely placed constraint points, and resembles a 3D version of Poisson disc sampling. Interior constraints are obtained by traversing the binary volume along the three principal axis. All points occurring between pairs of non-empty voxels are marked as interior. Only voxels which are marked as interior by all three traversals are kept as interior constraints. Again, only a subset of these are selected by the Poisson disc sampling technique described above. Exterior constraints are found by projecting each surface voxel in the volume to the image plane of each camera. If the ray from the surface voxel to a camera intersects other surface voxels, then the view of the voxel is blocked. Otherwise, the camera has an unobscured view, and an exterior constraint can be placed at a small distance away from the surface voxel along the ray towards the camera, as depicted in Figure 2. Once a specified number of constraints have been collected, they are given to the reconstruction algorithm. In this paper, we have used from 800 to 4500 surface constraints. In practice, we have found that 100 or 200 interior and exterior constraints suffice to define the orientation of the surface.

Figure 18 shows the original space carved data set of the toy dinosaur on the left, the implicit surface reconstruction in the middle, and a textured version of the implicit surface on the right. Reconstruction of a broccoli stalk is shown in Figure 19.

Fig. 18. From left to right: volumetric data set of the toy dinosaur from space carving, reconstructed implicit surface generated using the multi-order basis, reconstructed implicit surface textured using the original images.

7.5 Model Coloring

In order to create a color version of the surface, we begin with a polygonal model that was obtained through iso-surface extraction using Marching Cubes [20]. We assign a color to each triangle of the polygonal model

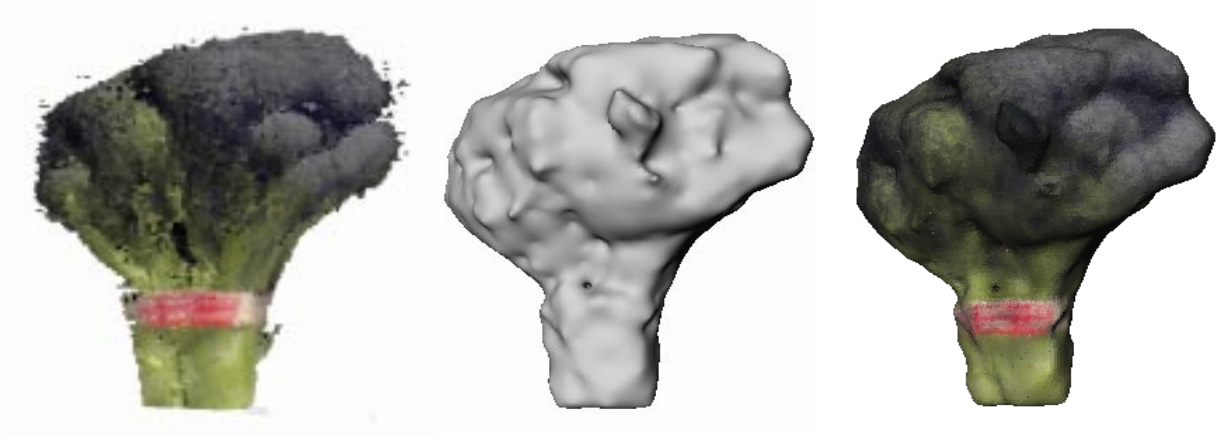

Fig. 19. From left to right: volumetric data set of a broccoli stalk from space carving, reconstructed implicit surface generated using the multi-order basis, reconstructed implicit surface textured using the original images.

by reprojecting the triangles back to the original input images. Each triangle in the polygonal model is subdivided until its projected footprint in the images is subpixel in size, so that it can simply take on the color of the pixel to which it projects. In most cases, a triangle is visible in several of the original images. We combine the colors from the different images using a weighted average. The weight of each color contribution is calculated by taking the dot product between the triangle normal and the view direction of the camera that captured the particular image. Cameras with viewing directions that are nearly perpendicular to the triangle normal contribute less than those with viewing directions that are nearly parallel to the triangle normal. We use z-buffering to ensure that only cameras with an unobscured view of the triangle can contribute to the triangle color. Figures 18 and 19 shows the final models of the toy dinosaur and broccoli once color has been applied. Figure 20 is a comparison of four of the original input images with rendered images of the reconstructed implicit surface from the same camera viewpoints.

8 Conclusion and Future Work

The reconstruction algorithm we have presented in this paper generates models that are smooth, seamless, and manifold. Our method is able to address challenges found in real data sets, including noise, non-uniformity, and holes in the data set. We have compared our technique to an exact interpolation algorithm (Crust), to thin-plate variational implicits, and to the original volumetric reconstruction using the toy dinosaur as a running example. Obvious advantages to the models generated by volumetric regularization are that there are no discretization artifacts as are found in volumetric models, and the surface is not jagged as in the Crust reconstruction. Volumetric regularization is most closely related to the thin-plate variational implicit approach. It compares favorably to the thin-plate variational implicits in computational time as well as in the surfaces that are generated. Using the multi-order radial basis function, volumetric regularization generates locally detailed, yet globally smooth surfaces that properly separate the features of the model.

Through regularization, our method can generate approximating, rather than interpolating, surfaces. This capability is especially advantageous in the presence of noise found in real data. In contrast, popular exact interpolation algorithms such as Crust and Alpha Shapes directly transfer the noise to the reconstructed surface.

We have adapted the variational implicits approach to real range data by developing methods to define surface, interior, and exterior constraints. Although surface points are directly supplied by the range data, we have introduced new methods for creating interior and exterior constraints using information about the camera positions used for capturing the data. We have applied this technique to space carved volumetric data and synthetic range images.

Finally, we have developed and applied measures of fitness error and curvature to the reconstructed surfaces. These measures were used to guide selection of the regularization and smoothness parameters. The measures attest to the benefits of the multi-order radial basis function over the thin-plate radial basis function.

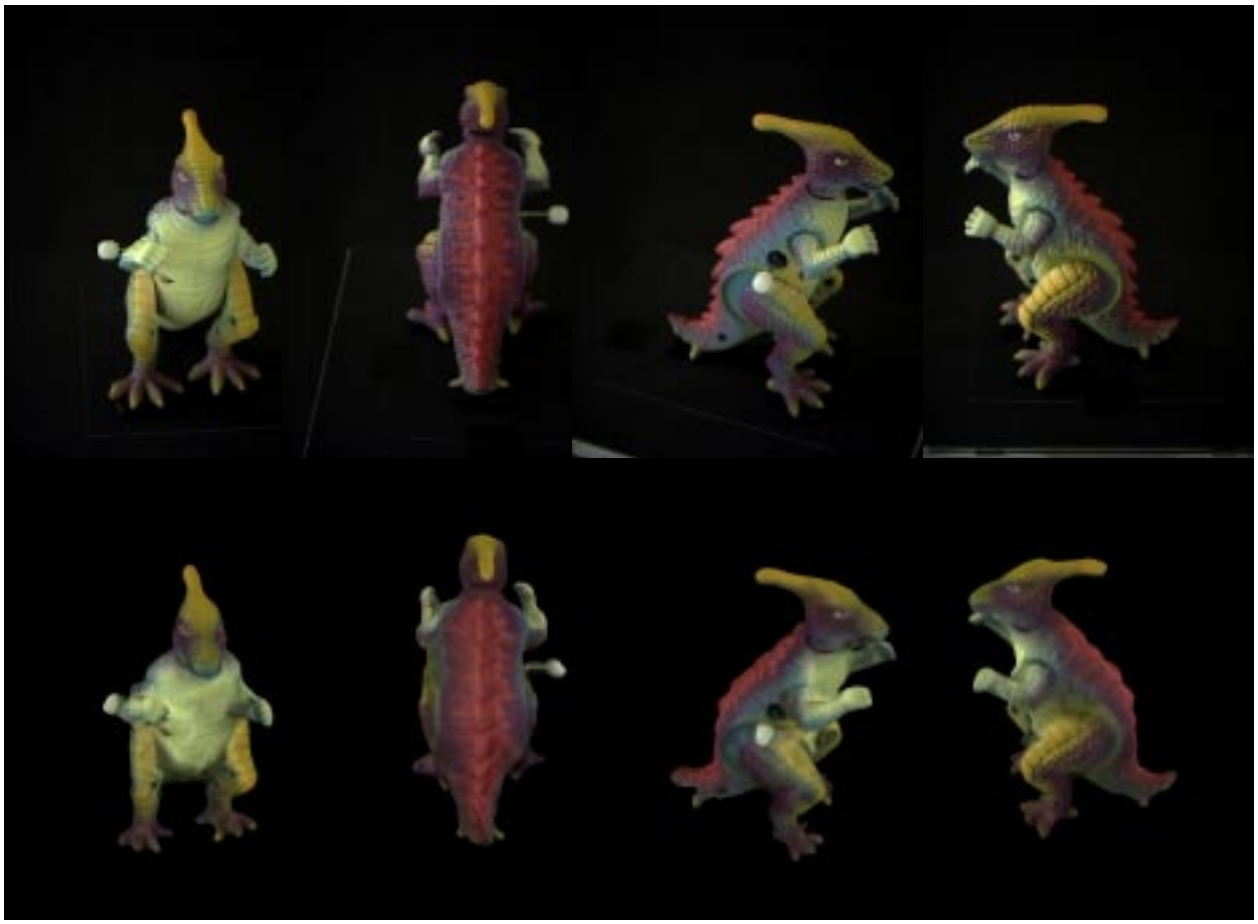

Fig. 20. The top row of images are four of the original input images used to generate the space carved data set of the toy dinosaur. The bottom row are images of the reconstructed implicit model rendered from the original camera viewpoints.

We plan to look at several potential improvements to our approach, including use of confidence measures, adaptive sampling, and adaptively modifying the basis functions locally. The generalized voxel coloring algorithm that generates the input range data also generates confidence measure for each surface point. This confidence measure is based upon the variance of the colors to which each surface voxel projects in the input images. Recall that the λ parameter may serve as a local fitting parameter since a unique value may be assigned for each constraint point. The confidence measure for each surface point could be used to set unique λ values for each surface constraint. Another possible improvement would be to create more constraint points in areas of the model that contain sharp or small features. These areas could be identified by the curvature measure applied to the implicit surface. This would be an adaptive sampling approach in which uniform sampling is used to generate an intial surface, and then, additional sampling would be performed in areas of high curvature. Another alternative is to assign different δ and τ values in different locations for the radial basis used in reconstructing the surface. The curvature measure is an indicator of where the surface may contain sharp features. Consequently, the δ and τ parameters could be defined for each constraint according to the curvature measure at that constraint point. These future directions hold promise of further refining the sharp features of reconstructed surfaces of real world objects.

9 Acknowledgements

We would like to acknowledge Steve Seitz for providing us with the original input images of the toy dinosaur and camera calibration information. A viewer for the surface voxels was provided by Mark Livingston, Bruce Culbertson, and Tom Malzbender of *Hewlett-Packard*. This work was funded by grants from ONR and NSF.

References

[1] Amenta, N., M. Bern, and M.Kamvysselis. A New Voronoi- Based Surface Reconstruction Algorithm. *SIGGRAPH Proceedings*, August 1998, pp.415-420.

[2] Bajaj, C. L. and G. Xu. Data Fitting with Cubic A-Splines. *Computer Graphics International Proceedings*, 1994.

[3] Bajaj, C. L., J. Chen, and G. Xu. Free Form Surface Design with A-Patches. *Graphics Interface Proceedings*, 1994, pp.174-181.

[4] Bajaj, C. L., F. Bernardini, and G. Xu. Automatic Reconstruction of Surfaces and Scalar Fields from 3D Scans. *SIGGRAPH Proceedings*, August 1995, pp.109-118.

[5] Bernardini, F., J. Mittleman, H. Rushmeier, and C. Silva. The Ball-Pivoting Algorithm for Surface Reconstruction. *IEEE Transactions on Visualization and Computer Graphics* , October-December 1999, Vol. 5, No. 4, pp.349-359.

[6] Boult, T.E. and J.R.Kender. Visual Surface Reconstruction Using Sparse Depth Data. *Computer Vision and Pattern Recognition Proceedings* , 1986, pp.68-76.

[7] Chen, F. and D. Suter. Multiple Order Laplacian Spline - Including Splines with Tension. *MECSE 1996-5, Dept. of Electrical and Computer Systems Engineering Technical Report, Monash University* , July, 1996.

[8] 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*, Vol. 1883 September 1999, pp.67- 74.

[9] Curless, B. and M. Levoy. A Volumetric Method for Building Complex Models from Range Images. *SIGGRAPH Proceedings*, August 1996, pp.303-312.

[10] Desbrun, M., M. Meyer, P. Schroder, and A.H. Barr. Implicit Fairing of Irregular Meshes Using Diffusion and Curvature Flow. *SIGGRAPH Proceedings*, August 1999, pp.317-324.

[11] Edelsbrunner H. and E.P. Mucke. Three-Dimensional Alpha Shapes. *ACM Transactions on Graphics*, Vol. 13, No. 1, January 1994, pp. 43-72.

[12] Fang, L. and D. Gossard. Multidimensional Curve Fitting to Unorganized Data Points by Nonlinear Minimization. *Computer-Aided Design*, Vol. 27, No. 1, January 1995, pp.48-58.

[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. Tight Fitting of Convex Polyhedral Shapes. *International Journal of Shape Modeling (special issue on Reverse Engineering Techniques)*, pp.111-126, 1998.

[16] Keren, D. and C. Gotsman. Fitting Curve and Surfaces with Constrained Implicit Polynomials. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol 21, No. 1, pp.21-31, 1999.

[17] Kolb, C., The Rayshade Homepage. *http://www-graphics.stanford.edu/ cek/rayshade/info.html.*

[18] Kutulakos, K.N. and S.M.Seitz. A Theory of Shape by Space Carving. *International Journal of Computer Vision* , August 1999.

[19] Lee, M. and G. Medioni. Inferring Segmented Surface Description from Stereo Data. *Computer Vision and Pattern Recognition Proceedings* , 1998, pp.346-352.

[20] Lorensen, W.E. and H.E. Cline. Marching Cubes. *SIGGRAPH Proceedings*, July 1987, pp.163-169.

[21] Seitz, S.M. and C.R. Dyer. Photorealistic Scene Reconstruction by Voxel Coloring. *Computer Vision and Pattern Recognition Proceedings* , 1997, pp.1067-1073.

[22] Suter, D. and F, Chen. Left Ventricular Motion Reconstruction Based on Elastic Vector Splines. To appear in *IEEE Transactions on Medical Imaging* , 2000.

[23] 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*, Vol. 20, No. 11, November, 1998.

[24] Taubin, G. Estimation of Planar Curves, Surfaces, and Nonplanar Spaces Curves Defined by Implicit Equations with Applications to Edge and Range Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 11, November, 1991.

[25] Taubin, G. An Improved Algorithm for Algebraic Curve and Surface Fitting. *Proceedings Fourth International Conference on Computer Vision*, May 1993, pp.658-665.

[26] Terzopoulos, D. The Computation of Visible- Surface Representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 10, No. 4, July, 1988.

[27] Turk, G. and J.F. O'Brien. Shape Transformation Using Variational Implicit Functions. *SIGGRAPH Proceedings*, August 1999, pp.335-342.