

Reconstructing Surfaces by Volumetric Regularization

Huong Quynh Dinh, Greg Turk, and Greg Slabaugh

Abstract

We present a new method of surface reconstruction that generates smooth and seamless models from sparse, noisy, non-uniform, and low resolution range data. Data acquisition techniques from computer vision, such as stereo range images and space carving, produce 3D 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 vision-based data sets. Our method constructs a 3D 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.

Index terms: regularization, surface fitting, implicit functions, noisy range data

I. 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 region growing techniques, algorithms based on computational geometry, and algebraic fitting methods. In many of these methods, noise from data acquisition becomes quite apparent in the resulting model. In the case of volumes carved from multiple images, 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. Some of the more popular surface reconstruction approaches are based on region growing, 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, region growing techniques rely on dense data to define 3D structure. Approaches based on algebraic data fitting are limited in their ability to reconstruct complex models of arbitrary topology. Our approach addresses the limitations of these surface reconstruction methods 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 3D implicit function from range data. We use an implicit 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. Implicit surfaces are well-suited for operations such as collision detection, morphing, blending, and

modeling with constructive solid geometry because they are formulated as a single analytical function, as opposed to a piecewise representation such as a polygonal model. Implicit surfaces can also accurately model soft and organic objects and can easily be converted to a polygonal model by isosurface extraction. 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 [28]. 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 vision-based 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 II, we review related work in surface reconstruction. In Section III, 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 IV. In Section V, we describe our approach to constructing

approximating surfaces. In Section VI, 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 VII.

II. 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 approaches, with a bias towards those more closely related to our own approach. The methods we describe include region growing, algorithms based on computational geometry, algebraic fitting, and surface regularization. Some of these methods fail in the presence of noisy data, while others are computationally limited to reconstructing simple models.

Examples of region growing techniques include Hoppe’s work on surface reconstruction [13] and Lee, Tang and Medioni’s work on tensor voting [17,23]. In Hoppe’s approach, a plane is fitted to a neighborhood around each data point, providing an estimate of the surface normal for the point. Hoppe constructs a graph that connects neighboring points using arcs weighted by the similarity between the surface normals. The surface normals are then propagated by traversing the graph as a minimal spanning tree. Lee and Medioni’s tensor voting method is similar 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. 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 neighboring features. Both methods described above are sensitive to noise in the data 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 con-

sequence 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 [10], the Crust algorithm [1], and the Ball-Pivoting algorithm [4]. In Alpha shapes, the shape is carved out by removing simplexes of the Delaunay triangulation of the point set. A simplex is removed if its circumscribing sphere is larger than the alpha ball. In the Crust algorithm, Delaunay triangulation is performed on the original set of points along with Voronoi vertices that approximate the medial axis of the shape. 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 non-manifold constructions 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. In Figure 2 (page 13, two left panels) , the Crust algorithm is applied to real range data obtained from the generalized voxel coloring method of [7]. Although, the general shape of the toy dinosaur is recognizable, the surface is rough due to the noisy nature of the real range data.

Many 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 [14]. Taubin fits a polynomial implicit function to a point set by minimizing the distance between the 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. In [24], Taubin develops a first order approximation of the Euclidean distance and improves the approximation in [25]. Gotsman and Keren create parameterized families of polynomials that satisfy desirable properties, such as fitness to the data or continuity preservation. Such a family must be large 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. 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 [3], Bajaj overcomes the complexity limitation by constructing piecewise polynomial patches (called A-patches) that combine to form one surface. Bajaj uses Delaunay triangulation to divide the point set into groups delineated by tetrahedrons. An A-patch is formed by fitting a Bernstein polynomial to the data points within each tetrahedron. 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.

Examples of algebraic methods developed earlier in the vision community that provide both smooth global fitting and accurate local refinement include the works of Terzopoulos and Metaxas on deformable superquadrics [27] and Pentland and Sclaroff on generalized implicit functions [19,20]. Both methods use superquadric ellipsoids as the global shape and

add local deformations to fit the data points. Terzopoulos and Metaxas separate the reconstructed model into global parameters defined by the superquadric coefficients, and local displacements defined as a linear combination of basis functions. The global and local deformation parameters are solved using dynamics. Pentland and Sclaroff define a generalized implicit model that consists of a superquadric ellipsoid, a modal deformation matrix that acts on the ellipsoid, and a displacement map that pushes the implicit surface along the surface normal towards data points. The modal deformation parameters are found by iteratively finding the minimum *RMS* error to the data points. The residual error after the deformation parameters have been found are incorporated into a displacement map that may exactly interpolate or just approximate the data. As with most of the algebraic methods, the primary drawback of these techniques is their inability to handle arbitrary topology. Complex models are constructed by combining multiple superquadrics. Terzopoulos and Metaxas' example of reconstructing a humanoid doll consists of separate deformable superquadrics for the torso, head, arms, and legs.

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 [5] and [26] are examples of regularization applied to height-field surfaces, and [11] is an example of regularization applied to parametric curves. Terzopoulos pioneered finite-differencing techniques to compute approx-

imate derivatives used in minimizing the thin-plate energy functional of a height-field. He developed computational molecules from the discrete formulations of the partial derivatives. Regularization is performed by iterating between coarse and fine levels in a multi-resolution hierarchy. Boulton 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 [11], 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. 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.

III. 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

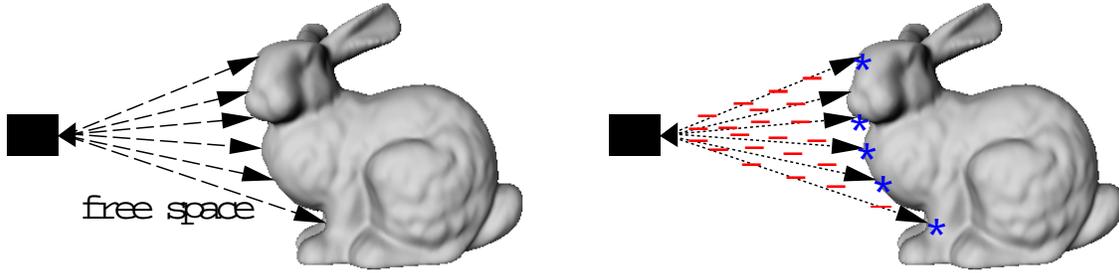


Fig. 1. The left panel shows the free space carved out by rays projecting from the camera to the object surface. The right panel shows surface (*) and exterior (-) constraints defined by the free space.

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. 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. We call the region between the camera and the surface point *free space*. This is shown in the left panel in Figure 1. Some space carving approaches to surface reconstruction make use of this information [8] as well. 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 (shown in the right panel in Figure 1). 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 interior 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, the interior of the object is known by virtue of surface enclosure. Hence, we can completely define the *existence* space - surface,

exterior, and interior - of the object.

In this paper we demonstrate our reconstruction technique with models from both synthetic range images and real data from voxel coloring [21,7]. We use this data to define interior, 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 2 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, which commonly occurs in algebraic fitting when a function is forced to interpolate all data points. In Section VII-D.2, we describe the real range data sets and how we applied the above method for constraint specification 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.

IV. VARIATIONAL IMPLICIT SURFACES

The surface reconstruction technique that we present in this paper is an extension of the variational implicit surfaces of [28]. 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 [28], Turk and O'Brien used the following radially symmetric basis function which inherently minimizes the thin-plate energy in 3D:

$$\phi(r) = |r|^3 \quad (1)$$

In the above equation, 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:

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

In the above equation, n is the number of constraint points; $\vec{\mathbf{c}}_i$ are the locations of the constraint points corresponding to the centers of the basis; and w_i are the weights for the basis. The constraints may be surface, interior, or exterior points as defined in the previous section. The polynomial term, $P(\vec{\mathbf{x}})$, spans the null space of the basis function. For thin-plate energy, the polynomial term consists of linear and constant terms because the thin-plate energy consists of second order derivatives. 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 $P(\vec{\mathbf{x}})$, are solved by constructing a linear system. Each constraint, $\vec{\mathbf{c}}_i$ is applied to Equation 2 to form one equation of the system. The function value, $f(\vec{\mathbf{c}}_i)$, 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. For an implicit function that evaluates to zero on the surface, $f(\vec{\mathbf{c}}_i) = 0$. 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 linear system can be formulated as a matrix equation, $Mb = v$, where v is an array of the function values, $f(\vec{\mathbf{c}}_i)$, at each constraint point, b 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. Explicitly, M , b , and v are defined as follows:

$$\begin{bmatrix} \phi(r_{11}) & \cdots & \phi(r_{1n}) & 1 & \vec{c}_1 \\ \vdots & & \vdots & 1 & \vdots \\ \phi(r_{n1}) & \cdots & \phi(r_{nn}) & 1 & \vec{c}_n \\ 1 & 1 & 1 & 0 & 0 \\ \vec{c}_1 & \cdots & \vec{c}_n & 0 & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_n \\ p_0 \\ \vec{p} \end{bmatrix} = \begin{bmatrix} f(\vec{c}_1) \\ \vdots \\ f(\vec{c}_n) \\ 0 \\ 0 \end{bmatrix} \quad (3)$$

$$r_{ij} = |\vec{c}_i - \vec{c}_j|$$

For surfaces, constraints are 3D coordinates, $\vec{c} = (c_x, c_y, c_z)$. 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. 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 2. 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* [18].

In the *unorganized points problem*, the orientation of a point set is not defined, and thus 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 has a function value of 1.0. For implicit surfaces constructed from a polygonal model, surface constraints are defined by the vertices of the polygons, and normal constraints are defined by vertex normals. Radial basis functions are centered at surface and normal constraint locations. In the case of vision-based data sets

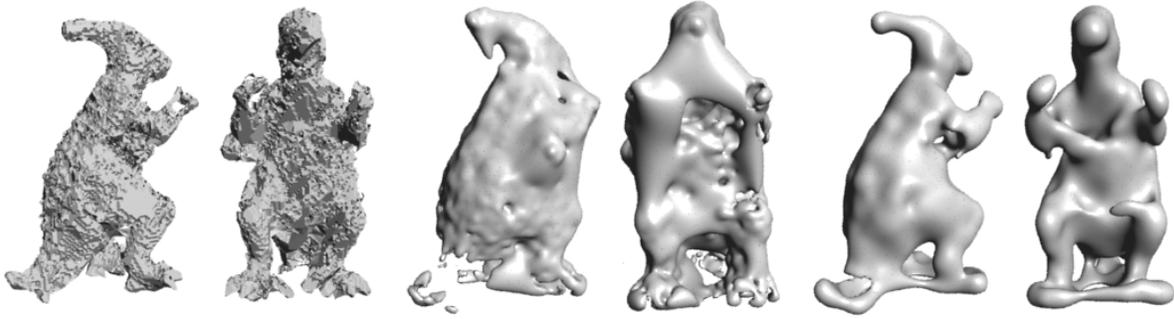


Fig. 2. From left to right: two views of the Crust reconstruction from entire data set, two views of the interpolating variational implicit surface, and two views of the approximating variational implicit surface. The variational implicit surfaces were generated using the basis, $\phi(r) = |r|^3$, centered at 3000 surface, 100 interior, and 264 exterior constraints.

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 III. These constraints define the orientation of the surface.

The reconstruction shown in the two middle panels of Figure 2 is an example of a surface reconstructed from real range data using the variational technique of Turk and O’Brien with the basis function $\phi(r) = |r|^3$. The constraints for this reconstruction were obtained from range images using the method described in Section III. 3000 surface, 264 exterior, and 100 interior constraints were specified. The resulting surface exactly interpolates the constraint points. 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 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 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. In a later section, we demonstrate that using different basis functions also improves

upon the surfaces that are created.

V. 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 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 [12]. A derivation is presented therein which shows that a summation of weighted radial basis functions as given in Equation 2 is the solution to minimizing a cost functional, H of the following form:

$$H[f] = \frac{1}{\lambda} \sum_{i=1}^n (y_i - f(\vec{\mathbf{x}}_i))^2 + \beta[f] \quad (4)$$

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 $\vec{\mathbf{x}}_i$; $\beta[f]$ 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. The term λ is often called the

regularization parameter. We use λ to specify how closely we want to approximate the data set. In our framework, 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 4 by repositioning λ as follows, and defining it for each constraint:

$$H[f] = \sum_{i=1}^n \frac{1}{\lambda_i} (y_i - f(\vec{x}_i))^2 + \beta[f] \quad (5)$$

The λ values may be assigned according to the noise distribution of the data acquisition technique. It is included in the system matrix of Equation 3 in the following manner:

$$\begin{bmatrix} \phi(r_{11}) + \lambda_1 & \cdots & \phi(r_{1n}) & 1 & \vec{c}_1 \\ \vdots & & \vdots & 1 & \vdots \\ \phi(r_{n1}) & \cdots & \phi(r_{nn}) + \lambda_n & 1 & \vec{c}_n \\ 1 & 1 & 1 & 0 & 0 \\ \vec{c}_1 & \cdots & \vec{c}_n & 0 & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_n \\ p_0 \\ \vec{p} \end{bmatrix} = \begin{bmatrix} f(\vec{c}_1) \\ \vdots \\ f(\vec{c}_n) \\ 0 \\ 0 \end{bmatrix} \quad (6)$$

$$r_{ij} = |\vec{c}_i - \vec{c}_j|$$

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 a 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

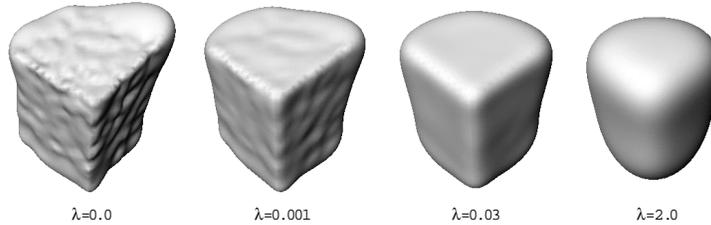


Fig. 3. The data set is from a synthetic range image of a corner of a cube. λ is varied from 0 at the left to 2.0 at the right. $\lambda = 0.001$ for the second reconstruction from the left.

regularizing parameter, where one λ value is used for all surface constraints, and another for all interior and exterior constraints. Figure 3 shows the results of applying different λ values on the same data set. Note that for each reconstruction, the same λ value was used for all surface points. 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 VII we validate this choice of λ using measures of fitness and curvature.

The two far right panels of Figure 2 show the reconstruction of the toy dinosaur using the same 3000 surface, 264 exterior, and 100 interior constraint as that of the two middle panels but with λ set to 0.001 for all surface constraints. 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 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. Computation 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 approximating model in Figure 2 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.

VI. A RADIAL BASIS FUNCTION FOR MULTIPLE ORDERS OF SMOOTHNESS

The results of Figure 2 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 2, is an exact fit to the data with no smoothness constraint. On the other hand, a smooth surface may become too blobby as seen in the variational implicit surfaces of Figure 2, which show that minimizing the thin-plate energy alone is not sufficient to produce a surface that separates features well and is locally detailed.

In [6], 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, where 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 the left plot of Figure 4. 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

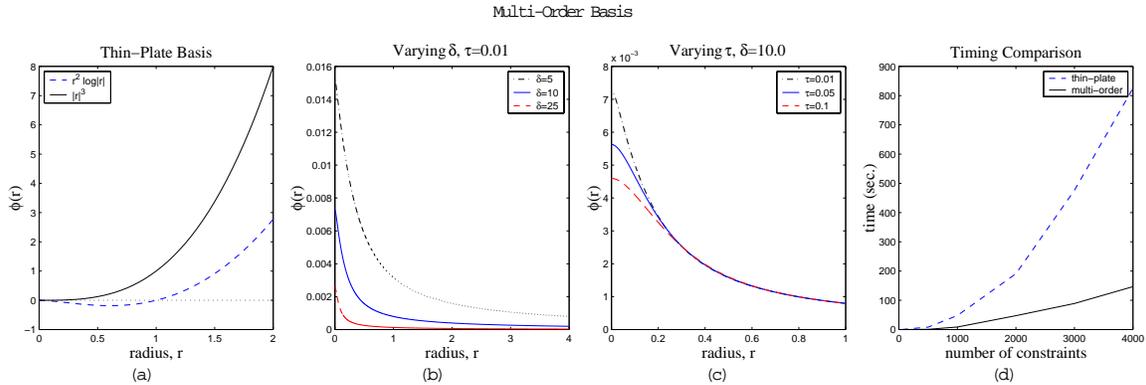


Fig. 4. Cross section of radially symmetric basis functions for thin-plate (left) and for a combination of first, second, and third order smoothness (middle and right). In the middle plot, δ varies from 10 to 30, $\tau = 0.01$. In the far right plot, $\delta = 0.25$, τ varies from 0.01 to 0.5.

constraints, is dense because constraint points are sparse and uniformly spread across the region of interest.

Surprisingly, a radial basis function that minimizes a combination of energies quickly falls toward zero, 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 fourth order energy resulted in the smallest *RMS* error. They concluded that basis functions that 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. Suter and Chen show that in 3D, the thin-plate spline basis has discontinuous first order derivatives 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 its continuity does not break down in 3D.

In [6], Chen and Suter derive such a basis, using a smoothness functional comprised of the first, second, and third order Laplacian operator. The associated partial differential equation is similar to Laplace's equation $-\Delta f = 0$, but also has higher order terms:

$$-\delta\Delta f + \Delta^2 f - \tau\Delta^3 f = 0 \quad (7)$$

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

$$\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2} \quad (8)$$

In equation 7, δ controls the amount of first order smoothness, and τ controls the amount of third order smoothness. The balance between δ and τ controls the amount of second order smoothness. The radial basis that inherently minimizes the above energy functional in 3D as derived in [6] is given below:

$$\phi(r) = \frac{1}{4\pi\delta^2 r} \left(1 + \frac{we^{-\sqrt{v}r}}{v-w} - \frac{ve^{-\sqrt{w}r}}{v-w} \right) \quad (9)$$

$$v = \frac{1 + \sqrt{1 - 4\tau^2\delta^2}}{2\tau^2} \quad w = \frac{1 - \sqrt{1 - 4\tau^2\delta^2}}{2\tau^2} \quad (10)$$

In the above equations, r is the distance from the center of the radial basis function. δ and τ are the only free parameters in defining the basis function. The middle and right plots of Figure 4 show one dimensional plots of the above function for various values of δ and τ . Unlike the plot for $\phi(r) = |r|^3$, these plots show that the value of the basis function quickly falls toward zero as the distance from its center increases.

The center plot of Figure 4 shows that as δ is increased, the basis falls toward zero faster and has less global influence. In the limit, the basis are simply spikes, and the resulting 3D implicit function would have 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 constraints where the implicit function evaluates to zero. The first three reconstructions of the toy dinosaur in Figure 5 correspond to the different values of δ in the middle plot of Figure 4. The surface becomes increasingly pinched at the constraint points as δ is increased. Visually, the pinching appears as surface detail that is

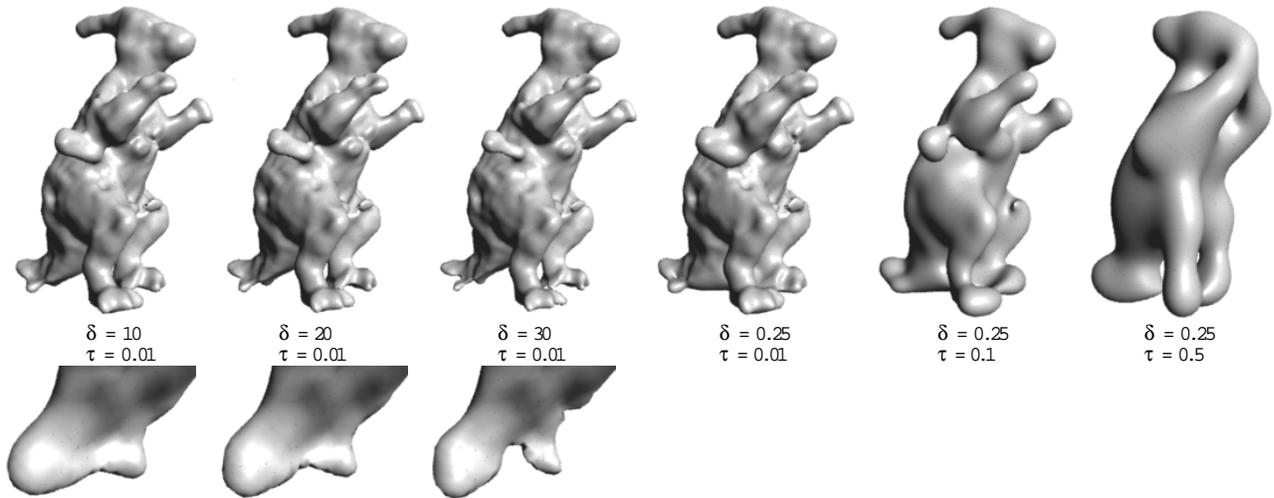


Fig. 5. δ is varied in the first three reconstructions, while τ is varied in the last three. The lower panels are closeups of the dinosaur tail, showing the differences between the reconstructions. The models were generated using the same 3000 surface, 100 interior, and 264 exterior constraints as those of the variational implicit reconstructions in Figure 2.

often missing in overly smooth surfaces. Despite the pinching, the surface remains globally smooth. These reconstructions are substantial improvements over those shown in Figure 2.

The right plot of Figure 4 shows that as τ is increased the center of the basis becomes increasingly smooth. In the limit, the basis becomes parabolic and enforces homogeneous curvature around constraint points. Consequently, the resulting 3D implicit function and its iso-surface becomes increasingly smooth as τ is increased. The last three reconstructions in Figure 5 correspond to the various values of τ plotted in Figure 4. In the limit, the reconstructed surface becomes amorphous, similar to those obtained from large values of λ .

The system matrix formed by Equations 9 and 10 is diagonally dominant and is especially amenable to the biconjugate gradient method of solving linear equations. Timing results show that the unknown weights of Equation 2 were solved in 1.7 minutes using the multi-order basis function with $\delta = 10$ and $\tau = 0.01$, while the system matrix generated for the same set of 3364 constraints using the thin-plate radial basis function required 36.7 minutes to solve. 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 Figure 5 for δ and τ were chosen to span the search space of possible values. Selection of the weights is constrained by Equation 10, 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 8 (page 25) 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. Visually, we can see that the first three reconstructions in Figure 5 are more pleasing in terms of trade-off between surface detail and smoothness than the last three. In addition, the arms, legs, and tail of the dinosaur in these reconstructions are better separated, and the back more detailed than that of the reconstruction shown in the two right panels of Figure 2 which was produced by the thin-plate radial basis function. Judging between the three left reconstructions of Figure 5 is, however, more difficult. Deciding that one is better than another amongst these three is quite arbitrary and subjective. 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 average curvature. They are exactly the attributes that the basis functions and the parameters, λ , δ and τ , regulate.

VII. RESULTS

Figure 6 is a visual comparison of different reconstructions of the toy dinosaur. This figure shows the original data (far left), the Crust reconstruction (middle left), and the implicit surface using the thin-plate basis function (middle). At the middle right and far right of the figure is the new reconstruction using the techniques of this paper (untextured and textured). Note that the round protrusion beneath the arm is the turn key of the actual toy dinosaur (see Figure 12 for the original images). We show further evidence in this section, using new and synthetic data sets, that reconstruction by volumetric regularization generates globally



Fig. 6. 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, our new implicit surface reconstruction using the multi-order radial basis function, and a textured version of our reconstruction.

smooth, yet detailed, surfaces. Finally, we discuss the addition of color to the models.

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.

A. Selecting λ

Recall from Section V 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 .

A.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 [18], and then finding the closest vertex of the polygonal model to a given data point. This vertex serves an initial starting point on the surface from which we can then search for even closer surface locations to the given data

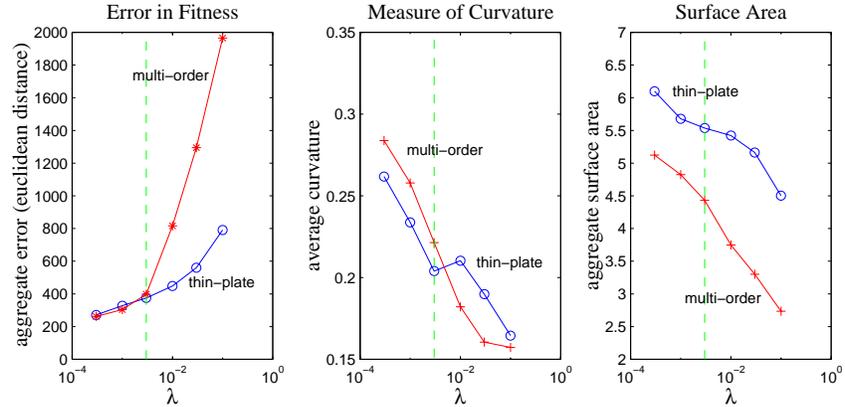


Fig. 7. From left to right: plots of aggregate fitness error, average curvature, and surface area for the toy dinosaur reconstructions using the thin-plate (o) and multi-order (*) radial basis functions, for $\lambda = 0.0003, 0.001, 0.003, 0.01, 0.03,$ and 0.1 for surface constraints. The vertical dashed line in all three plots mark the location of $\lambda = 0.003$.

point. We find a closer surface location by crawling along the surface in small increments until a small increment in any four orthogonal directions along the surface does not yield a location that is closer to the original data point. When this stopping location is found, the Euclidean distance is calculated and accumulated.

The left plot in Figure 7 shows the total fitness error for the dinosaur reconstructions using thin-plate and multi-order basis functions at various 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 (dashed line in the plots of Figure 7 mark the location of $\lambda = 0.003$).

A.2 Measure of Curvature

The next quality measure we describe is the average curvature of the surface. We measure the curvature of the surfaces using polygonal models that are extracted from the implicit function using Marching Cubes [18]. We measure curvature at each vertex of these polygonal models using a curvature approximation that was developed for the smoothing operator in [9]. 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 7 shows the average curvature for the thin-plate and multi-order radial basis functions at various 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 λ values of 0.003 or less. The low curvature of the thin-plate surface at λ values of 0.003 or less corresponds with the blobbiness seen in the thin-plate reconstructions of Figure 2. 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 7 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 7. All examples of our reconstruction were created using a value of 0.001 for λ .

B. Selecting δ and τ

Recall from Section VI that δ controls the amount of first order smoothness, while τ controls the amount of third order smoothness. Figure 5 shows 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 for the first three reconstructions in Figure 5 rather than those used in the last three reconstructions because the first three models are, visually, a better reproduction of the original data set from space carving. The last three reconstructions in Figure 5 span different τ values and show that large τ values produce models 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 8 shows the valid range of values for δ and τ to avoid imaginary values in Equation 10. 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.

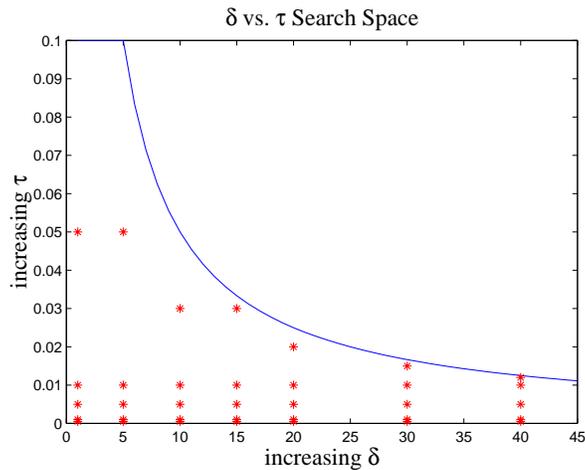


Fig. 8. Limitation placed on δ and τ to avoid imaginary values in Equation 10. The limit is shown as a blue line, and the tested parameters are indicated by the red stars.

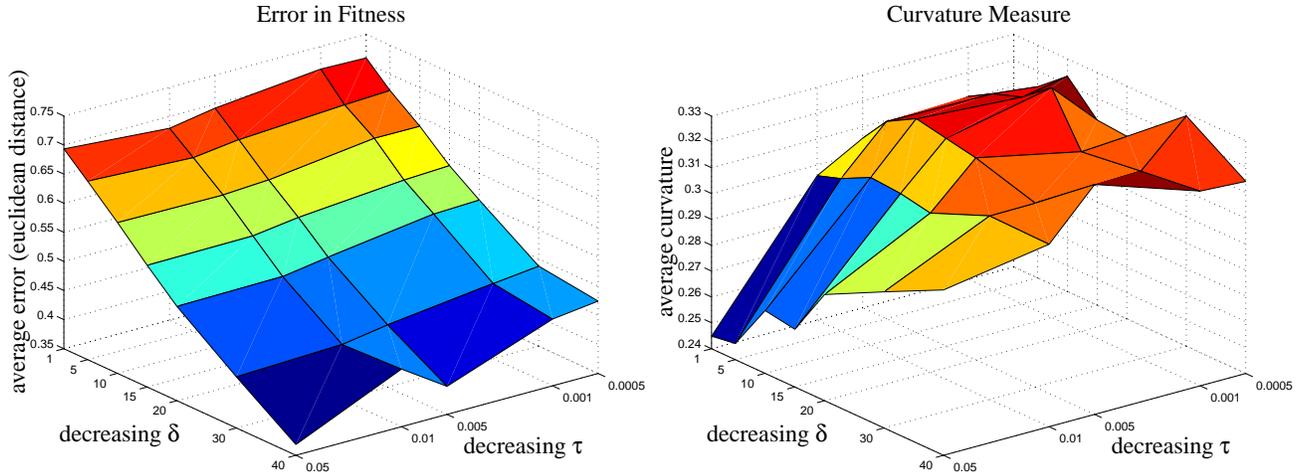


Fig. 9. Average fitness error (left) and average curvature (right) for various values of δ and τ . From near to far along the δ axis, $\delta = 40.0, 30.0, 20.0, 15.0, 10.0, 5.0,$ and 1.0 . From near to far along the τ axis, $\tau = 0.05, 0.01, 0.005, 0.001,$ and 0.0005

B.1 Fitness Error

The left plot of Figure 9 is 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 8). The plot shows that, within this search space, δ has greater influence on fitness to the data since changes in δ cause greater changes in fitness error than changes in τ . In general, larger δ values produce a tighter fit, corroborating the fact that δ controls first order smoothness. Although it is not plotted in the graph, a δ value of 50.0 was found to result in even greater fitness error, indicating that δ should not be set above 40.0. In practice, 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 0.5. 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 second and third gridlines along the τ axis.

B.2 Measure of Curvature

The right plot of Figure 9 is the average curvature of the reconstructions using the same values of δ and τ as that of the left plot in Figure 9. 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 . Recall that low curvature is associated with overshooting surfaces that tend to be blobby. δ values of 40.0 or 30.0 (first and second grid lines along the δ axis) and τ values of 0.01 or 0.005 (second and third grid lines along the τ axis) generate surfaces which exhibit high curvature and low fitness error compared to other values.

C. 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 10 and 11. 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 5). We have found in practice that 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. By delimiting the space of useful values for λ , δ , and τ , we have shown that these three control parameters do not create additional burden on the user.

D. 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

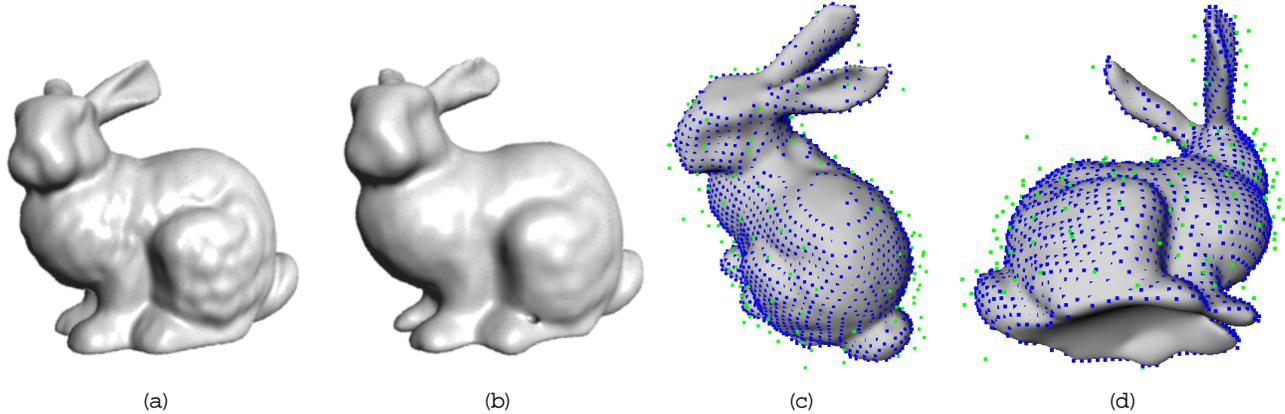


Fig. 10. From left to right: the original Stanford bunny consisting of 69,451 triangles, two views of surface and exterior constraints used to perform reconstruction, and two views of the variational implicit surface reconstructed using the multi-order basis. Blue squares are surface constraints. Green squares are negative exterior constraints.

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.

D.1 Synthetic Range Data

We use a modified ray-tracer [15] 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 on 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 III. 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 far left panel of Figure 10 shows the original Stanford Bunny model consisting of 69,451 triangles, and the middle and middle left panels show the distribution of constraints defined for the Bunny. Surface constraints are drawn as blue squares embedded in the surface, and negative constraints are drawn as green squares. The middle right and far right panels of Figure 10 show the implicit surface reconstructed from 2168 surface and 193 exterior constraints, using the multi-order basis function. Values of $\lambda = 0.001$, $\delta = 10$, and $\tau = 0.01$ were used to reconstruct the surface. The two views of the reconstructed bunny 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. The model is closed on the top and bottom of the Bunny even though few constraint points were placed in those locations (top of the Bunny shown in the third panel of Figure 10). 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.

D.2 Real Volume-Carved Data

Synthetic data does not 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 use two data sets of real objects obtained through methods from computer vision – a toy dinosaur (from Steve Seitz [21]) and a broccoli stalk. Both data sets were obtained by taking about 20 images approximately on a circle around each object. Thin-shelled, voxelized surfaces were then constructed using the generalized voxel coloring algorithm [7]. 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,

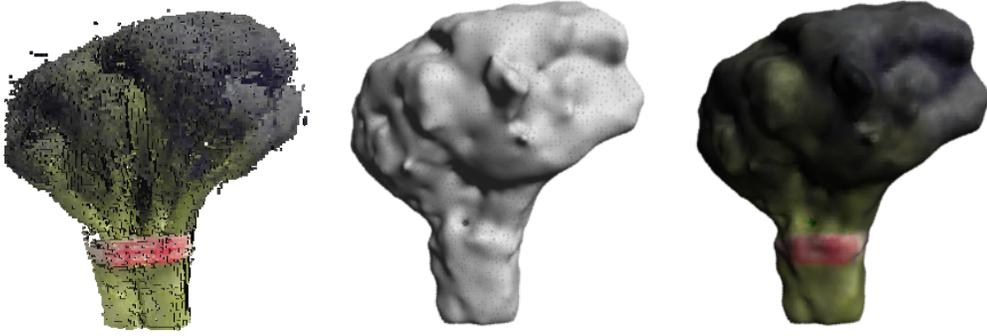


Fig. 11. From left to right: volumetric data set of a broccoli stalk from space carving, reconstructed implicit surface generated using the multi-order basis, reconstructed implicit surface textured using the original images.

and blue channels. 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 III 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 small radius 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 surface voxels are marked as interior. Only voxels which are marked as interior by all three traversals are kept as interior constraints. 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



Fig. 12. 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. A novel viewpoint of the implicit model is shown in Figure 6

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 1. Note that for each surface voxel, an exterior constraint is created for each camera that has an unobscured view of the surface voxel. Again, only a subset of the interior and exterior constraints are selected by the Poisson disc sampling technique described above. 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. Figures 6 and 11 are examples of our reconstructions from space carved data.

E. 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 [18]. We assign a color to each triangle of the polygonal model 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 6 and 11 show the final models of the toy dinosaur and broccoli from novel viewpoints after color has been applied. Figure 12 is a comparison of four of the original input images with rendered images of the reconstructed implicit surface from the same camera viewpoints.

VIII. 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, low resolution, and holes in the data set. We have compared our technique to an exact interpolation algorithm (Crust), to thin-plate variational implicit surfaces, 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 can generate approximating, rather than interpolating, surfaces, and is most closely related to the thin-plate variational implicit approach. It compares favorably to the thin-plate variational implicit surfaces in computation 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.

We have adapted the variational implicit surfaces approach to real range data by developing meth-

ods 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 in 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 guided selection of the regularization and smoothness parameters, and attest to the benefits of the multi-order radial basis function over the thin-plate radial basis function.

We plan to look at several potential improvements to our approach, including use of confidence measurements, adaptive sampling, and adaptively modifying the basis functions locally. Recall that the λ parameter may serve as a local fitting parameter since a unique value may be assigned to each constraint point. For each 3D surface point obtained from the generalized voxel coloring algorithm, the λ value can be assigned based on the variance of the colors to which the surface voxel projects in the input images. Another improvement would be to define more constraint points in areas of the model that contain sharp or small features. These areas can be identified by looking for high curvature regions on the surface. This is an adaptive sampling approach in which uniform sampling is used to generate an initial surface, and then, additional sampling would be performed in areas of high curvature. Another alternative is to assign different δ and τ values for the multi-order basis at different locations in the model. The δ and τ parameters could be assigned 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.

IX. ACKNOWLEDGEMENTS

We thank James O’Brien for stimulating discussions about alternate basis functions for surface creation. We 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*, Aug. 1998, pp.415–420.
- [2] Bajaj, C. L., J. Chen, and G. Xu. "Free Form Surface Design with A-Patches", *Graphics Interface Proceedings*, 1994, pp.174–181.
- [3] 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.
- [4] Bernardini, F., J. Mittleman, H. Rushmeier, and C. Silva. "The Ball-Pivoting Algorithm for Surface Reconstruction", *IEEE Transactions on Visualization and Computer Graphics*, Oct.-Dec. 1999, 5(4), pp.349–359.
- [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", *MECSE 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*, Vol. 1883 Sep. 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] Desbrun, M., M. Meyer, P. Schroder, and A.H. Barr. "Implicit Fairing of Irregular Meshes Using Diffusion and Curvature Flow", *SIGGRAPH Proceedings*, Aug. 1999, pp.317–324.
- [10] Edelsbrunner H. and E.P. Mucke. "Three-Dimensional Alpha Shapes", *ACM Transactions on Graphics*, Jan. 1994, 13(1), pp. 43–72.
- [11] Fang, L. and D. Gossard. "Multidimensional Curve Fitting to Unorganized Data Points by Nonlinear Minimization", *Computer-Aided Design*, Jan. 1995, 27(1), pp.48–58.
- [12] Giroso, 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.
- [13] Hoppe, H., T. DeRose, and T. Duchamp. "Surface Reconstruction from Unorganized Points", *SIGGRAPH Proceedings*, July 1992, pp.71–78.

- [14] 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.
- [15] Kolb, C., The Rayshade Homepage. <http://www-graphics.stanford.edu/cek/rayshade/info.html>.
- [16] Kutulakos, K.N. and S.M.Seitz. "A Theory of Shape by Space Carving", *International Journal of Computer Vision*, Aug. 1999.
- [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] Pentland, A. "Automatic Extraction of Deformable Part Models", *International Journal of Computer Vision*, 1990, Vol.4, pp.107–126.
- [20] Sclaroff, S. and A. Pentland. "Generalized Implicit Functions for Computer Graphics", *SIGGRAPH Proceedings*, July 1991, pp.247–250.
- [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*, Nov. 1998., 20(11).
- [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*, Nov. 1991, 13(11).
- [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*, July 1988, 10(4).
- [27] Terzopoulos, D. and D. Metaxas. "Dynamic 3D Models with Local and Global Deformations: Deformable Superquadrics", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, July 1991, 13(7).
- [28] Turk, G. and J.F. O'Brien. "Shape Transformation Using Variational Implicit Functions", *SIGGRAPH Proceedings*, Aug. 1999, pp.335–342.