Eurographics Symposium on Geometry Processing (2005)
M. Desbrun, H. Pottmann (Editors)

Reconstruction of Solid Models from Oriented Point Sets

Michael Kazhdan[†]

Abstract

In this paper we present a novel approach to the surface reconstruction problem that takes as its input an oriented point set and returns a solid, water-tight model. The idea of our approach is to use Stokes' Theorem to compute the characteristic function of the solid model (the function that is equal to one inside the model and zero outside of it). Specifically, we provide an efficient method for computing the Fourier coefficients of the characteristic function using only the surface samples and normals, we compute the inverse Fourier transform to get back the characteristic function, and we use iso-surfacing techniques to extract the boundary of the solid model.
The advantage of our approach is that it provides an automatic, simple, and efficient method for computing the solid model represented by a point set without requiring the establishment of adjacency relations between samples or iteratively solving large systems of linear equations. Furthermore, our approach can be directly applied to models with holes and cracks, providing a method for hole-filling and zippering of disconnected polygonal models.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling

1. Introduction

Reconstructing 3D surfaces from point samples is a well studied problem in computer graphics. The ability to reconstruct such surfaces provides a method for zippering samples obtained through scanning, filling in holes in models with degeneracies, and re-meshing existing models. While there has been much work in this area, we provide a novel approach that is based on basic calculus, providing a simple method for reconstructing solid models from oriented point sets (point samples with associated normals).

Our approach takes advantage of the fact that an oriented point set sampled from the surface of a solid model provides precisely enough information for computing surface integrals. Thus, by formulating the solution of the surface reconstruction problem in terms of volume integrals, we can apply Stokes' Theorem to transform the volume integrals into surface integrals and compute a discrete approximation using the oriented point samples.

In practice, our approach provides a method for reconstructing a water-tight model from an oriented point set in three easy steps: (1) The point-normal pairs are splatted into a voxel grid. (2) The voxel grid is convolved with an integration filter. (3) The reconstructed surface is extracted as an iso-surface of the voxel grid.

The advantage of our approach is its simplicity: Splatting the oriented point samples into the voxel grid can be done without necessitating the establishment of adjacency relations between the samples. Convolving with a filter can be efficiently performed using the Fast Fourier Transform [FFTW]. And finally, standard methods such as the Marching Cubes algorithm [LC87] can be used to extract the reconstructed surface, returning a triangulation that is guaranteed to be water-tight.

Figure 1 demonstrates our method for an oriented point set obtained by sampling the surface of a dinosaur head. The original model is shown on the left, the samples are shown in the middle, and the reconstructed model is shown on the right. Note that even though the points were sampled from a model which was not water-tight, (there are holes in the eyes, the nostrils, and the mouth) our method succeeds in returning a seamless mesh that closely approximates the input data, accurately capturing the fine model details.

Our method addresses the surface reconstruction problem by approximating integration by a discrete summation and a direct implementation of our method assumes that the samples are uniformly distributed over the surface of a model. However, in many situations, the input samples are not uni-

[†] Johns Hopkins University

© The Eurographics Association 2005.

Figure 1: *The initial model (left), a non-uniform sampling of points from the model (middle), and the reconstructed water-tight, surface obtained using our method (right).*

formly distributed. To this end, we also present a simple heuristic method that assigns a weight to each point-normal pair corresponding to the sampling density about the sample. We show that this heuristic provides a robust method for assigning weights that approximate the regional sampling density allowing for the reconstruction of surfaces from samples that are not uniformly distributed. Figure 1 shows an example of the use of this method: Though, the samples are not uniformly distributed over the surface of the original triangulation, with denser sampling in regions of higher curvature, our weighting method assigns the correct weights to the samples and the reconstructed surface closely approximates the input sample in regions of both dense and sparse sampling.

The remainder of this paper is structured as follows: In Section 2 we review previous work in surface reconstruction. We present our approach in Section 3 and provide results and discussion in Section 4. Finally, we conclude in Section 5 by summarizing our results and discussing possible directions for future research.

2. Related Work

The importance of the surface reconstruction problem has motivated a large body of research in computer graphics and previous approaches can be broadly grouped into one of three categories: (1) Methods that address the surface reconstruction problem through the use of computational geometry techniques, (2) methods that address the surface reconstruction problem by directly fitting a surface to the point samples, and (3) methods that address the surface reconstruction problem by fitting a 3D function to the point samples and then extracting the reconstructed surface as an iso-surface of the implicit function.

In general, the computational geometry based methods proceed by computing either the Delaunay triangulation of the point samples or the dual Voronoi diagram and using the cells of these structures to define the topological connectivity between the point samples [Bo84, EM94, ABK98, AC*00, ACK01, DG03]. The advantage of these types of approaches is that the complexity of the reconstructed surface is on the order of the complexity of the input samples. Moreover, for many of these reconstruction methods, it is possible to bound the quality of the reconstruction if the sampling density is known. However, these methods suffer from two limitations: First, they require the computation of the Delaunay triangulation which can be inefficient for large point samples. Second, these methods tend to perform less effectively when the point samples are not uniformly distributed over the surface of the model.

Surface fitting methods approach the reconstruction problem by deforming a base model to optimally fit the input sample points [TV91, CM95]. These approaches represent the base shape as a collection of points with springs between them and adapt the shape by adjusting either the spring stiffnesses or the point positions as a function of the surface information. As with the computational geometric approaches, these methods have the advantage of generating a surface reconstruction whose complexity is on the order of the size of the input samples. However, these methods tend to be restrictive as the topology of the reconstructed surface needs to be the same as the topology of the base shape, limiting the class of models that can be reconstructed using this method.

The third class of approaches taken in reconstructing surfaces uses the point samples to define an implicit function in 3D and then extracts the reconstructed surface as an iso-surface of the function, [HD*92, CL96, Wh98, CB*01, DM*02, OBS04, TO04]. The advantage of these types of approaches is two-fold: First, the extracted surface is always guaranteed to be water-tight, returning a model with a well-defined interior and exterior, and second, the use of an implicit function does not place any restrictions on the topological complexity of the extracted iso-surface, providing a reconstruction algorithm that can be applied to many different 3D models. In general, this type of approach has the limitation that the complexity of the reconstruction process is a function of the resolution of the voxel grid, not the output surface, and many of these approaches use hierarchical structures to localize the definition of the implicit function to a thin region about the input samples, thereby reducing both the storage and computational complexity of the reconstruction.

3. Approach

The goal of our work is to provide a method that takes as its input an oriented point set sampled from the surface of a model and returns a water-tight reconstruction of the surface. Our approach is to construct the characteristic function of the solid defined by the point samples – the function whose value is one inside of the solid and zero outside of it – and then to extract the appropriate iso-surface.

To compute the characteristic function of the solid we compute its Fourier coefficients. In practice, one would compute the Fourier coefficients by integrating the complex exponentials over the interior of the model. However, by Stokes' Theorem, we can express this volume integral as a surface integral, only using information about the positions and the normals of points on the boundary. Since this is precisely the information provided as the input to our method we have sufficient information to compute the Fourier coef-

© The Eurographics Association 2005.

ficients of the characteristic function, allowing us to use the inverse Fourier transform to compute its values.

We begin our discussion by reviewing Stokes' theorem. Next, we provide a method for expressing each of the Fourier coefficients of the characteristic function in terms of a surface integral. Then, we provide an efficient method for computing the Fourier coefficients and describe a method for selecting the iso-value to be used for the iso-surface extraction. Finally, we provide a simple heuristic for addressing the problem of non-uniformly sampled input data.

3.1. Stokes' Theorem

Stokes' Theorem is an extension of the Fundamental Theorem of Calculus which provides a method for expressing the integral of a function over the interior of a region as an integral over the region's boundary. In our work, we will be considering a specific instance of Stokes' Theorem known as the *Divergence Theorem* or *Gauss's Theorem*. Specifically, if $M \subset \mathbb{R}^3$ is a three-dimensional solid and $\vec{F} = (F_x, F_y, F_z) : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is a vector-valued function, the Divergence Theorem expresses the volume integral as a surface integral:

$$\int_M \nabla \cdot \vec{F}(p) dp = \int_{\partial M} \langle \vec{F}(p), \vec{n}(p) \rangle dp$$

where $\nabla \cdot \vec{F} = \partial F_x/\partial x + \partial F_y/\partial y + \partial F_z/\partial z$ is the divergence of $\vec{F}$ and $\vec{n}(p)$ is the surface normal at the point p.

Our approach is motivated by the observation that if $\{\vec{p}_i, \vec{n}_i\} \subset M$ is a uniformly sampled point set, the volume integral can be approximated using Monte-Carlo integration:

$$\int_M \nabla \cdot \vec{F}(p) dp \approx \frac{|M|}{N} \sum_{i=1}^{N} \langle \vec{F}(\vec{p}_i), \vec{n}_i \rangle.$$

In the next section, we show that the surface reconstruction problem can be reduced to the computation of volume integrals. Using the above Monte-Carlo approximation, we can compute these volume integrals as a summation over a set of surface samples, providing a method for reconstructing surfaces from oriented point sets.

3.2. Defining the Fourier Coefficients

To reconstruct a surface from a set of samples, we will first construct the *characteristic function* of the solid model. This is a function defined in 3D whose value is equal to one inside the solid and zero outside. Once the characteristic function is obtained, we can obtain the surface by using iso-surface extraction. Instead of computing the characteristic function directly, we first compute its Fourier coefficients and then compute the inverse Fourier Transform of these coefficients. Though less direct, this approach is easy to implement because the Fourier coefficients can be expressed as volume integrals and hence can be computed from an oriented point set with a Monte-Carlo approximation of Stokes' Theorem.

Our goal is to compute the characteristic function by computing its Fourier coefficients. Specifically, if M is a solid model and χ_M is its characteristic function, we would like to compute the coefficients:

$$\begin{aligned} \hat{\chi}_M(l,m,n) &= \int_{\mathbb{R}^3} \chi_M(x,y,z) e^{-i(lx+my+nz)} dxdydz \\ &= \int_{p \in M} e^{-i(lp_x+mp_y+np_z)} dp. \end{aligned}$$

(Note that since the function χ_M is equal to one inside the model and zero outside, integrating the complex exponentials against the characteristic function is equivalent to computing the integral of these functions over the solid model.) Using the Divergence Theorem, we know that if $\vec{F}_{l,m,n} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is a function such that:

$$\left(\nabla \cdot \vec{F}_{l,m,n} \right)(x,y,z) = e^{-i(lx+my+nz)}$$

the volume integral can be expressed as the surface integral:

$$\hat{\chi}_M(l,m,n) = \int_M e^{-i(lp_x+mp_y+np_z)} dp = \int_{\partial M} \langle \vec{F}(p), \vec{n}(p) \rangle dp$$

where $\vec{n}(p)$ is the unit normal of the surface ∂M at p.

Since the input to our algorithm is an oriented point set, we can compute the Fourier coefficients of the characteristic function using a Monte-Carlo approximation. Specifically, given the points $\{\vec{p}_1, \ldots, \vec{p}_N\}$ with associated normals $\{\vec{n}_1, \ldots, \vec{n}_N\}$, we set:

$$\hat{\chi}_M(l,m,n) = \frac{1}{N} \sum_{j=1}^{N} \langle \vec{F}_{l,m,n}(\vec{p}_j), \vec{n}_j \rangle.$$

In order to be able to evaluate the above summation explicitly, we need to choose the functions $\vec{F}_{l,m,n}$ whose divergences are equal to the complex exponentials. Perhaps the most direct way to do this is to set $\vec{F}_{l,m,n}$ to be the function:

$$\vec{F}_{l,m,n}(x,y,z) = \begin{pmatrix} \frac{i}{l+m+n} e^{-i(lx+my+nz)} \\ \frac{i}{l+m+n} e^{-i(lx+my+nz)} \\ \frac{i}{l+m+n} e^{-i(lx+my+nz)} \end{pmatrix}.$$

The limitation of this type of function is that it is anisotropic, treating different directions differently. Thus, the obtained characteristic function does not rotate with the model.

To avoid this problem, we use functions $\vec{F}_{l,m,n}$ that do not depend on the alignment of the coordinate axis:

$$\vec{F}_{l,m,n}(x,y,z) = \begin{pmatrix} \frac{il}{l^2+m^2+n^2} e^{-i(lx+my+nz)} \\ \frac{im}{l^2+m^2+n^2} e^{-i(lx+my+nz)} \\ \frac{in}{l^2+m^2+n^2} e^{-i(lx+my+nz)} \end{pmatrix}. \qquad (1)$$

Figure 2 demonstrates the differences in the characteristic function obtained using the different $\vec{F}_{l,m,n}$ for individual point samples with different normal orientations (left and middle) and a point set consisting of 100 points randomly distributed over the boundary of the square. The middle row

© The Eurographics Association 2005.

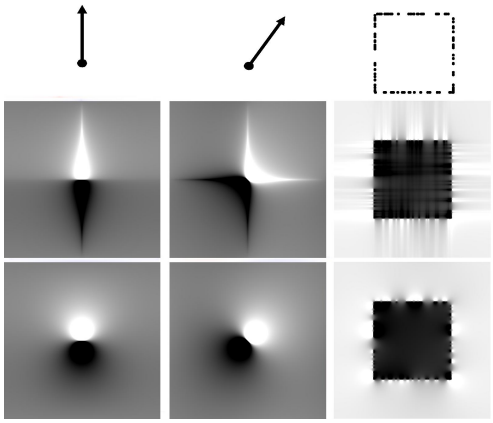

Figure 2: *Characteristic functions obtained from a point with a* 90° *degree normal (left), a point with a* 60° *degree normal (middle), and a point set sampled from the boundary of a square (right). Reconstructions are shown using the functions* $\vec{F}_{l,m,n}$ *whose coordinate functions are equal (middle) and functions defined by Equation 1 (bottom). Points with positive value are drawn in white, points with negative value in black, and points with zero-value in gray.*

shows the characteristic function obtained using the functions $\vec{F}_{l,m,n}$ whose coordinate functions are equal. In this case, the anisotropic nature of the function results in characteristic functions that do not rotate with the normals, resulting in a non-uniform distribution of noise in the reconstruction of the square. In contrast, the characteristic functions obtained using Equation 1 (botom) rotate with the normals and give rise to a smoother reconstruction. (The uniqueness of the functions $\vec{F}_{l,m,n}$ is discussed in the Appendix.)

One should note that computing the Fourier Coefficient of the characteristic function using a Monte-Carlo approximation of Stokes' Theorem defines all the Fourier coefficients of the characteristic function except the constant order term. Thus, the obtained function is well defined up to an additive constant. Furthermore, in the case that we do not know the sampling density (i.e. the surface area associated with each sample point), the resultant characteristic function is only well defined up to a multiplicative constant.

3.3. Computing the Fourier Coefficients

While the method described in the previous section provides a direct way for obtaining the Fourier coefficients of the characteristic function, it requires a summation over all of the input samples to compute a single Fourier coefficient. Thus, in the case that both the number of input samples and the reconstruction band-width are large, explicitly computing the summation becomes prohibitively slow. In this section, we show that this summation can be expressed as a convolution so that the Fast Fourier Transform can be used to compute the characteristic function efficiently. Consequently, if N is the number of input samples and b is the reconstruction band-width, using the FFT for surface reconstruction we obtain a reconstruction algorithm with complexity $O(b^3 \log b + N)$, (as compared to the $O(b^3 N)$ complexity of the explicit summation approach).

Our approach is to represent the oriented point set by a gradient field which is almost everywhere zero except at the sample locations. At these locations the value of the gradient field is equal to the normal of the corresponding point sample. Specifically, we set $\vec{N} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ to be the function:

$$\vec{N}(\vec{p}) = \frac{1}{N} \sum_{j=1}^{N} \delta_{\vec{p}_j}(\vec{p}) \vec{n}_j$$

where $\delta_{\vec{p}}$ is the Kronecker Delta function centered at $\vec{p}$. The advantage of this representation is that its Fourier coefficients are closely related to the Fourier coefficients of the characteristic function. Specifically, if we set $\vec{l} = (l,m,n)$ then the $\vec{l}$-th Fourier coefficients of the characteristic function and the $\vec{l}$-th Fourier coefficients of the gradient field are:

$$\hat{\chi}_M(\vec{l}) = \frac{i}{N\|\vec{l}\|^2} \sum_{j=1}^{N} e^{-i\langle \vec{l}, \vec{p}_j \rangle} \langle \vec{n}_j, \vec{l} \rangle \quad \hat{\vec{N}}(\vec{l}) = \frac{1}{N} \sum_{j=1}^{N} e^{-i\langle \vec{l}, \vec{p}_j \rangle} \vec{n}_j.$$

Thus, the $\vec{l}$-th Fourier coefficient of the characteristic function can be obtained by multiplying the $\vec{l}$-th Fourier coefficient of the gradient field by $i/\|\vec{l}\|^2$ and taking the dot product with the vector $\vec{l}$:

$$\hat{\chi}_M(\vec{l}) = \frac{i}{\|\vec{l}\|^2} \langle \hat{\vec{N}}(\vec{l}), \vec{l} \rangle.$$

(Note that, by abuse of notation, $\hat{\vec{N}}(\vec{l})$ is a 3D complex vector, obtained by computing the Fourier coefficients of each of the coordinate functions of $\vec{N}$ independently.)

In practice, we implement this reconstruction of the characteristic function by "splatting" the sample normals into a voxel grid, (where each voxel stores a 3-vector) and then convolving the "splatting" function with a filter $\vec{F}$ whose (l,m,n)-th Fourier coefficient is:

$$\hat{\vec{F}}(l,m,n) = \frac{i(l,m,n)}{(l^2+m^2+n^2)}.$$

This is an extended notion of the standard convolution. In general, convolving the point samples with a filter results in a function that is the sum of the filters centered at each of the sample points. The result of our extended convolution is a summation of filters that are not only centered at the sample points but also aligned with the normals. Intuitively, this means that the reconstruction of the characteristic function is performed by taking functions shown in the bottom row of Figure 2, translating them to align with the sample points and rotating them to align with the normals.

One can view this reconstruction of the characteristic function as an integration. Specifically, we know that integration acts on the complex exponentials by:

$$\int e^{ik\theta} d\theta = \frac{-i}{k} e^{ik\theta}$$

© The Eurographics Association 2005.

so that integration is equivalent to convolution with a filter whose k-th Fourier coefficient is $-ik/\|k\|^2$. In this context, multiplication of the $\vec{l}$-th Fourier coefficient of $\vec{N}$ by $i\vec{l}/\|\vec{l}\|^2$ can be viewed as an integration of the surface gradient field – a vector field which points in the direction of the surface normals at the surface points and is zero everywhere else. Integrating this gradient field, we obtain a function that is constant almost everywhere, with a sharp change in value at the sample points so that points interior to the model all have the same constant value c_i and points outside all have the same constant value c_o, with $c_o \neq c_i$. (We make the normals inward facing so that the value of the characteristic function is larger inside the model and smaller outside. This gives rise to the change in sign of the integration filter.)

3.4. Extracting the Iso-Surface

In order to extract an iso-surface from the characteristic function, we need to choose an appropriate iso-value. We describe two ways in which this can be done.

First, we can use the fact that the characteristic function has a value of one inside the solid and a value of zero outside of it. This motivates choosing an iso-value equal to 0.5. Since, as discussed in the previous section, the characteristic function is only defined up to an additive constant, we would first need to compute the constant order coefficient, or solid volume. (This can be done, for example, by using a function such as $\vec{F}_{0,0,0}(x,y,z) = \frac{(x,y,z)}{3}$.) However, this type of approach has two limitations: First, it assumes that the sampling density is given so that the ambiguity in the multiplicative scale factor of the characteristic function has been resolved. Second, and more important, this method fails to be robust in the case when the input points are samples from a model that is not water-tight. In this case, the computed volume will vary with the translational alignment of the points and the shape of the reconstructed surface will depend on the coordinate frame of input samples.

Instead, we can compute the average value of the obtained characteristic function at the sample positions $\vec{p}_j$. Since we would like the input points to lie on the reconstructed surface, we simply set the iso-value equal to this average. The advantage of this approach is that it provides a robust iso-surfacing value even in the case that input points are obtained from a model that is not-water-tight.

3.5. Non-Uniform Sampling

In our approach, we compute the characteristic function by using Stokes' theorem to transform a volume integral into a surface integral. We then approximate the surface integral by a discrete summation over points distributed on the surface of the model. In order for the approximation to be robust, the distribution of sample points needs to be uniform. However, in many applications, the samples may be non-uniform.

In the case that the sampling density about each point is given, we can address the problem of non-uniformity in the standard manner – weighing the contribution of each sample point to the overall integral as a function of the sampling density. Specifically, we assign a weight to each sample that is the reciprocal of the regional sampling density and proceed as before: (1) We scale each normal by the weight of its sample and splat the weighted normals into a voxel grid, (2) we convolve the voxel grid with the integration filter, and (3) we extract the iso-surface at the iso-value equal to the weighted average of the iso-function at the sample locations.

For the case that we do not know the sampling density a priori, we propose a simple heuristic for assigning weights to the samples. Our approach is based on the observation that the 3D function obtained by summing Gaussians centered at each of the sample points has the property that the value of the function is proportional to the local sampling density. This motivates the following approach for assigning weights to the samples: We "splat" the samples into a voxel grid – adding a value of one to each voxel whose position corresponds to the position of a sample – and convolve with a Gaussian filter. We then assign a weight to each sample point $\vec{p}$ which is the reciprocal of the value of the convolution at $\vec{p}$. Since the value of the convolution at a point is proportional to the local sampling density, the reciprocal is proportional to the surface area associated to the point, providing the appropriate weighting for the Monte-Carlo integration.

While this method is easy to implement and we find that it works well in practice, we stress that it is only a heuristic as it assigns a weight that is inversely proportional to the 3D sampling density, not the sampling density over the surface.

4. Results and Discussion

In order to determine how well our method works in practice, we ran our reconstruction approach on oriented point sets obtained by sampling triangulated models and compared the obtained reconstruction with the initial model. Specifically, we designed our experiments to evaluate the quality of the reconstruction as a function of the number of samples, the band-width of the reconstruction, the presence of holes and missing data, non-uniform sampling, and the addition of positional and normal noise.

4.1. Experimental Results

To evaluate the effect of sample size and band-width on the quality of the reconstruction, we obtained samples by randomly choosing points on the surface of the Stanford Bunny, where the probability of choosing a position from within a specific triangle was proportional to the area of the triangle (as in [OF*01]) and the normal of the point was set to the normal of the triangle from which the position was sampled. The original model is shown in Figure 3, with an inset showing that the original model has holes in its base and therefore

© The Eurographics Association 2005.

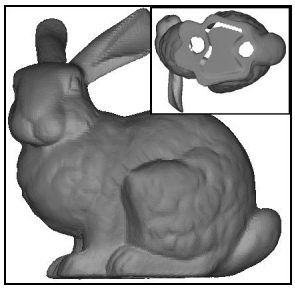

Figure 3: *The original model from which the test samples were obtained. The inset shows a view of the base of the model, indicating that the model is not water-tight.*

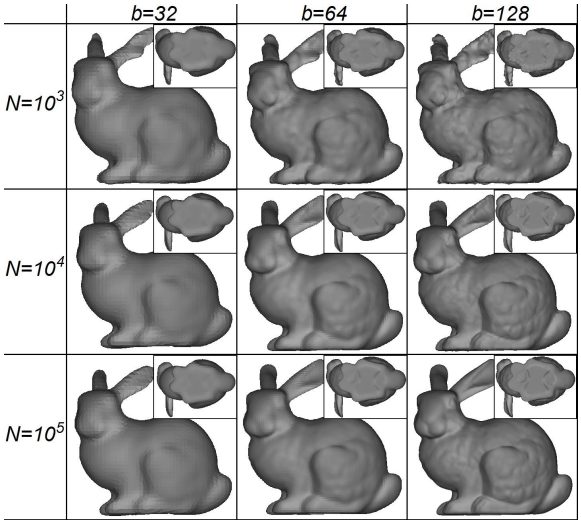

Figure 4: *Reconstructed surfaces of the Stanford bunny with different numbers of surface samples (N) and different band-widths (b).*

does not define a solid volume. The results of reconstruction experiments at different sample sizes (N) and different band-widths (b) are presented in Figure 4 and Tables 1 and 2.

As Figure 4 indicates, our method returns a model that approximates the input and smoothly fills in the holes where data is missing. The image also shows that the accuracy of reconstruction increases with sample size and band-width.

	$b=32$	$b=64$	$b=128$
$N=1000$	T=12,384 s=0.16	T=51,112 s=1.07	T=206,448 s=8.05
$N=10,000$	T=11,952 s=0.18	T=49,776 s=1.11	T=201,804 s=8.04
$N=100,000$	T=11,880 s=0.52	T=49,508 s=1.44	T=200,668 s=8.60

Table 1: *The size T, in triangles, and compute time s, in seconds, for reconstruction of the Stanford bunny as a function of the number of samples (N) and band-width (b).*

	$b=32$	$b=64$	$b=128$	Error
$N=1000$	0.43% 3.11%	0.30% 2.35%	0.29% 2.37%	RMS Maximum
$N=10,000$	0.32% 2.42%	0.12% 1.17%	0.06% 0.68%	RMS Maximum
$N=100,000$	0.31% 2.33%	0.10% 0.70%	0.04% 0.37%	RMS Maximum

Table 2: *The distance, as a percentage of model size, of the initial model from the reconstructed surface as a function of the number of samples (N) and band-width (b).*

The computational complexity and the complexity of the reconstructed surfaces are described in Table 1. Because our reconstruction algorithm runs in $O(b^3\log(b)+N)$ time and because the triangle count is quadratic in the sampling resolution, we find that doubling the band-width results in computation time that is roughly eight times larger and a reconstructed model with four times the number of triangles. Additionally, the table highlights the fact the limiting factor in the reconstruction is the computation of the forward and inverse Fourier transforms, so that changing the number of input samples does not markedly affect computation time.

To evaluate how well our method reconstructs the surface of the model, we randomly sampled the initial model at 100,000 points and computed the distance from these test points to the reconstructed models. Table 2 gives the accuracy of the reconstructed models shown in Figures 4 in terms of the root mean square (RMS) and maximum distance of these models from the test points (with the distance given as a percentage of the voxel resolution). As expected, the table indicates that the quality of the reconstruction improves when the sample size and reconstruction band-width are increased. The table also shows that the RMS errors does not exceed the size of a voxel and, when the surface is reconstructed at sufficiently high resolution, the maximum error is also smaller than a voxel. Thus, our method provides a fast, water-tight reconstruction, with sub-voxel accuracy. (Note that since the bunny model has holes, water-tight reconstructions must introduce surface patches that are not present in the initial model. As a result, symmetrizing the error metric by sampling points on the reconstruction and measuring the distance to the initial model would result in an inaccurate measure of reconstruction accuracy.)

To evaluate the robustness of our method in the presence of larger aberrations, we ran our algorithm on an oriented point set that was uniformly sampled from part of a model of human head. Figure 5 shows the surface from which the samples were taken (top row) and the reconstruction returned by our method (bottom row). In this experiment, the sample size was set to $N=100,000$ and the reconstruction band-width was set to $b=128$. Although the input samples only come from a fraction of the surface, the figure shows that our

© The Eurographics Association 2005.

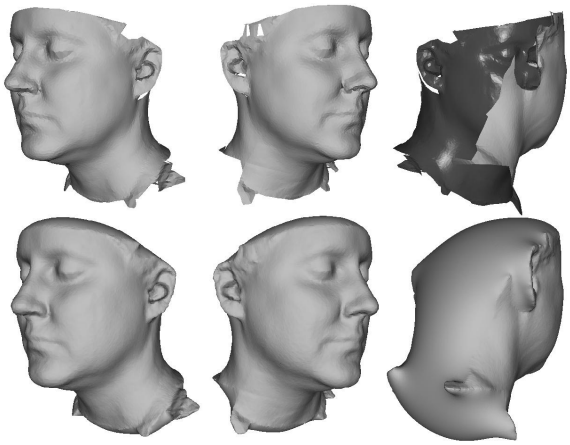

Figure 5: *Reconstructed surfaces of a human face using samples from a 3D model. Views of the original model are shown in the top row. Views of the reconstructed, water-tight model are shown in the bottom row.*

method returns a solid model that accurately fits the input samples while providing a reasonable reconstruction of the surface in the regions where no samples could be provided.

Figure 6 shows the reconstructions for a point set that was uniformly sampled from the surface of the toes of Michelangelo's David model ($N = 100,000$, $b = 128$). The initial model is shown on the left, with a crack between the first two toes resulting from the scanner's inability to see the region. Using our method to reconstruct the surface of the model by assigning uniform weights to each sample point gives rise to the surface shown in the middle column. While this surface accurately approximates the data and results in a water-tight reconstruction, it introduces a topological handle connecting the first two toes. By assigning weights to the samples that are inversely proportional to the regional sampling density, as described in Section 3.5, we obtain a new reconstruction (right column) that gives more weight to the points near the boundary of the crack. This forces the reconstruction to maintain the surface orientation near the missing data and results in a reconstruction that does not have the topological artifact introduced when uniform weights are used.

To evaluate the performance of our method in the presence of non-uniform sampling, we generated an oriented point set by randomly sampling 100,000 points from the surface of the "Happy Buddha" model, Figure 7(a), where the probability of choosing a point was a function of the surface curvature. An image of the point set is shown in Figure 7(b), with sparse sampling in low curvature regions (e.g. the stomach and the base of the pedestal) and dense sampling in high curvature regions. Figure 7(c) shows the reconstruction obtained using uniform weighting which over-integrates the high curvature areas, resulting in a poor reconstruction in planar regions. In contrast, Figure 7(d) shows the reconstruction obtained when we use our weighting method to assign weights to the samples. As the figure indicates,

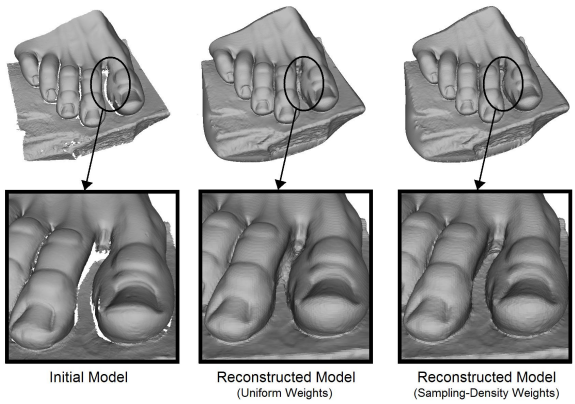

Figure 6: *The toes of Michelangelo's David model (left) and the reconstructions obtained using uniform weights (middle) and weights that are inversely proportional to the sampling density (right).*

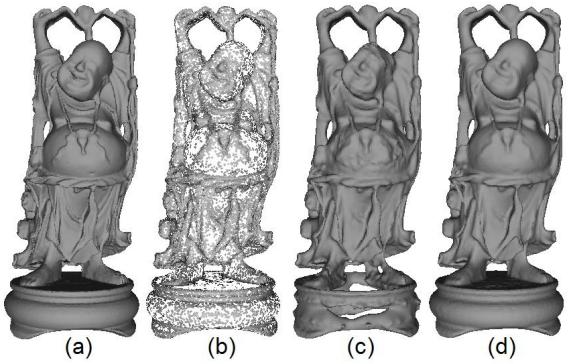

Figure 7: *Reconstructions from a non-uniform point set. The initial Buddha model (a), the point set sampled as a function of surface curvature (b), the reconstructed surface obtained using uniform point weighting (c), and the surface reconstructed using our weight assignment method (d).*

the reconstructions closely approximates the initial surface (Figure 7(a)), indicating that though our weight assignment method is a heuristic, it gives a good approximation to the true sampling density and results in robust reconstructions.

Finally, to test how well our method performs in the presence of noise, we sampled a cow model at 100,000 points and added noise to both the position and normal of each sample. Noise was added to each sample by randomly displacing the position by a fixed distance and randomly changing the direction of the normal by a fixed angle. The reconstructed cow models are shown in Figure 8. The rows show the change in the reconstructed model as the positional noise is increased and the columns show the change as the angular noise is increased. The results in the image indicate that our reconstruction method is robust in the presence of both positional and angular error. In particular, the figure shows that our method reconstructs all but the features of the model that are smaller than the displacement size. For example, when a displacement value of 1/32-nd of the bounding radius is used (right column), the body of the cow is reconstructed,

© The Eurographics Association 2005.

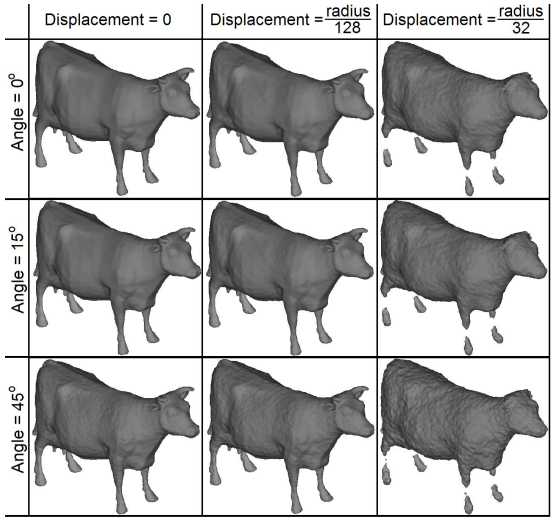

Figure 8: *Reconstructions of a cow model where varying amounts of noise were added to the position and the normals of the samples.*

but the legs and horns, whose widths are smaller the size of the displacement, can no longer be reconstructed. More importantly, these results show that our method remains robust in the presence of significant normal error. This is particularly important in practical settings, because though the positions of the sample points may be accurately obtained with 3D scanners, the normals, which are differential properties of the surface, are often more noisy.

4.2. Discussion

4.2.1. Memory Requirements

One limitation of our approach is its memory requirements. If we would like to obtain a high detail reconstruction of a model, it is necessary to generate a large voxel grid. Specifically, if we would like to reconstruct a model at band-width b it is necessary to perform forward and inverse FFTs on a $2b \times 2b \times 2b$ voxel grid. Assuming that the values of the voxels are stored at floating point precision this implies that, given the memory limitations of commodity computers, our method cannot reconstruct a surfuce using a voxel grid with resolution larger than $512 \times 512 \times 512$.

4.2.2. Translation, Additivity, and Noise

An important feature of our method is that it commutes with translation and is additive. That is, (1) the implicit function obtained from a translated point set is equal to the translation of the implicit function obtained from the initial point set, and (2) the implicit function obtained from the union of two subsets is equal to the sum of the implicit functions obtained from each subset independently. A property that any such method satisfies is that if we can model the noise acting on the samples by a probability distribution, then the implicit function obtained from noisy samples will approximate the convolution of the noise model with the implicit function obtained from noise-free samples. For example, if the sampling noise is Gaussian, the reconstructed characteristic function of the noisy samples will approximate a smoothed version of the characteristic function of the noise-free samples. As a result, the surface reconstructed from samples with Gaussian noise will resemble a smoothed version of the initial model.

4.3. Comparison to Related Methods

Our method differs from much of the previous work in surface reconstruction in that the fitting of the reconstructed surface to the samples requires a simple global optimization. Specifically, due to the additive nature of the reconstruction process, we fit the characteristic function to the sample points on a point-by-point basis without considering the proximity of adjacent points. Then, to extract the iso-surface, we optimize by simply setting the iso-value equal to the average of the characteristic function at the sample points.

In order to evaluate the effects of our global optimization on the accuracy of retrieval, we compared the reconstructions obtained using our method with the reconstructions obtained using Radial Basis Functions [CB*01, RBF] and Multi-Level Partition of Unity Implicits [OB*03, MPU].

To compare these methods we ran the reconstruction algorithm on point sets obtained from water-tight models of a human pelvis and the Armadillo Man. In the first experiment, the point sets were non-uniformly sampled from the surface of the model and in the second experiment, the points were uniformly sampled from the surface of the model and then noise was added to the samples prior to the reconstruction. The results of our experiments are shown in Figures 9 and 10 and the complexity and accuracy of the reconstructions are described in Table 3. In both experiments, the points sets consisted of $N = 100,000$ samples and the surfaces were reconstructed at a band-width of $b = 128$. The accuracy of a reconstruction was measured by uniformly sampling 100,000 points from both the initial model and the reconstructed surface and computing the distances of the points sampled from the initial surface to the reconstructed surface and the distances of the points sampled from the reconstructed surface to the initial surface.

The results from the pelvis experiment demonstrate that all three methods return accurate reconstructions of the model, despite the non-uniformity in the sampling and the large genus of the model. Furthermore, as the results in Table 3 indicate, even though our optimization step is a simple averaging operation that results in faster reconstruction, the lack of explicit local optimizations does not give rise to a less accurate reconstruction. Thus our method returns a reconstruction that has the same resolution and accuracy as competing reconstructions but runs in less time.

The difference in the approaches becomes amplified in the

© The Eurographics Association 2005.

Figure 9: *A non-uniform sampling of points from the human pelvis (a) and the reconstructions obtained using Radial Basis Functions (b), Partition of Unity Implicits (c), and our method (d).*

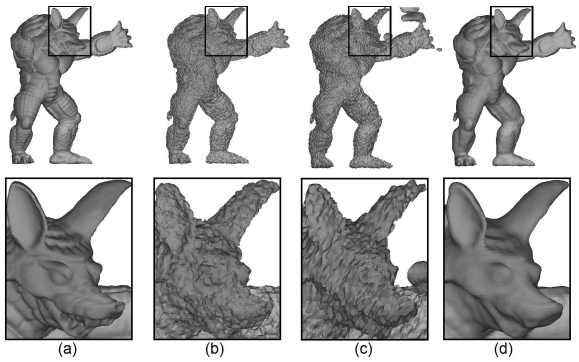

Figure 10: *The armadillo-man model (a) and reconstructions from a noisy point sample obtained using Radial Basis Functions (b), Partition of Unity Implicits (c), and our method (d). The inset shows a magnification of the model's face, showing that the local nature of previous reconstruction approaches results in noisy reconstructions.*

Armadillo Man experiment when noise is added to the samples. In the results of this experiment, the local optimizations of the Radial Basis Function and Partition of Unity approaches give rise to surfaces that strive to interpolate the samples and are themselves noisy. In contrast, the additive nature of our method (as described in the previous section) gives rise to a surface reconstruction that averages out the noise and returns a surface that resembles a smoothed version of the initial model and is, on average, about twice as accurate as the reconstruction of the competing methods.

The local nature of the optimizations in the Radial Basis Function and Partition of Unity approaches is further highlited by the timing results in Table 3. Since the input samples are noisy, the iterative local optimizations converge less efficiently, making the Radial Basis Function reconstruction run five times slower and the Partition of Unity reconstruction run three time slower. In contrast, the efficiency of our method, which only performs a single averaging optimization, is not affected by the noise in the data, returning an accurate reconstruction in the same amount of time.

	Model	Time	Tris.	RMS	Max
Pelvis	RBF	5 : 23	302*K*	0.10%	2.19%
	MPU	0 : 39	288*K*	0.12%	3.37%
	Ours	0 : 12	289*K*	0.11%	1.85%
Armadillo	RBF	24 : 10	200*K*	0.13%	0.50%
	MPU	2 : 14	205*K*	1.16%	12.71%
	Ours	0 : 11	176*K*	0.07%	0.67%

Table 3: *A comparison of the reconstruction time (min:sec), the triangle count, and the accuracy of the reconstructions of the pelvis and Armadillo Man models using Radial Basis Functions, Multi-Level Partition of Unity Implicits, and our method.*

5. Conclusion and Future Work

In this paper we have presented a novel method for reconstructing seamless meshes from oriented point samples. Our method differs from past approaches in that it leverages Stokes' Theorem to provide a method for surface reconstruction that does not require the establishment of topological relations between adjacent points and involves no implicit parameter fitting. Consequently, we provide a surface reconstruction method that is both simple and efficient. We have shown that the method is robust and can be used to reconstruct the surface of 3D solids in the presence of missing data, non-uniform sampling, and noise.

In the future, we would like to consider using a method akin to PolyCube-Maps [TH*04] to decompose the input samples into a collection of regionally localized subsets. This would enable us to run our method on each of the subsets independently, allowing us to overcome the memory bottleneck that restricts the resolution of reconstructable detail. We expect that the additive nature of our reconstruction process should facilitate this task by providing a method for stitching together adjacent reconstructions.

References

[ABK98] AMENTA N., BERN M., KAMVYSSELIS M.: A new Voronoi-based surface reconstruction algorithm. *Computer Graphics (Proceedings of SIGGRAPH 98)* (1998), 415–21. 2

[AC*00] AMENTA N., CHOI S., DEY T., LEEKHA N.: A simple algorithm for homeomorphic surface reconstruction. *ACM Symp. on Computational Geometry* (2000), 213–222. 2

[ACK01] AMENTA N., CHOI S., KOLLURI R.: Power crust. *6th ACM Symp. on Solid Modeling and Applications* (2001), 249–260. 2

[Bo84] BOISSONNAT J.: Geometric structures for three dimensional shape representation. *ACM TOG* (1984), 266–286. 2

[CB*01] CARR J., BEATSON R., CHERRIE H., MITCHEL T., FRIGHT W., MCCALLUM B., EVANS T.: Reconstruction and representation of 3D objects with radial basis functions. *ACM SIGGRAPH* (2001), 67–76. 2, 8

[CL96] CURLESS B., LEVOY M.: A volumetric method for building complex models from range images. *Computer Graphics (Proceedings of SIGGRAPH 96)* (1996), 303–312. 2

© The Eurographics Association 2005.

[CM95] CHEN Y., MEDIONI G.: Description of complex objects from multiple range images using an inflating balloon model. *Computer Vision and Image Understanding* (1995), 325–334. 2

[DG03] DEY T., GOSWAMI S.: Tight cocone: A water tight surface reconstructor. In *Proc. 8th ACM Symp. on Solid Modeling Applications* (2003), 127–134. 2

[DM*02] DAVIS J., MARSCHNER S., GARR M., LEVOY M.: Filling holes in complex surfaces using volumetric diffusion. In *Intl. Symp. on 3D Data Processing, Visualization and Transmission* (2002), 428–438. 2

[EM94] EDELSBRUNNER H., MÜCKE E.: Three-dimensional alpha shapes. *ACM TOG* (1994), 43–72. 2

[RBF] FAST RBF 1.4.1: `http://www.farfield-technology.com/`, 2004. 8

[FFTW] FFTW: `http://www.fftw.org`, 1998. 1

[HD*92] HOPPE H., DEROSE T., DUCHAMP T., MCDONALD J., STUETZLE W.: Surface reconstruction from unorganized points. *ACM SIGGRAPH* (1992), 71–78. 2

[LC87] LORENSEN W., CLINE H.: Marching cubes: A high resolution 3d surface reconstruction algorithm. *SIGGRAPH Conference Proceedings* (1987), 163–169. 1

[MPU] MPU IMPLICITS: `http://www.mpi-sb.mpg.de/~ohtake/mpu_implicits/`. 8

[OB*03] OHTAKE Y., BELYAEV A., ALEXA M., TURK G., SEIDEL H.: Multi-level partition of unity implicits. *ACM TOG* (2003), 463–470. 8

[OBS04] OHTAKE Y., BELYAEV A., SEIDEL H.: 3D scattered data approximation with adaptive compactly supported radial basis functions. *SMI* (2004), 31–39. 2

[OF*01] OSADA R., FUNKHOUSER T., CHAZELLE B., DOBKIN D.: Matching 3d models with shape distributions. *SMI* (2001) 154–166. 5

[TH*04] TARINI M., HORMANN K., CIGNONI P., MONTANI C.: Polycube-maps. *TOG (SIGGRAPH 2004)* (2004), 853–860. 9

[TO04] TURK G., O'BRIEN J.: Modelling with implicit surfaces that interpolate. *ACM TOG* (2004), 855–873. 2

[TV91] TERZOPOULOS D., VASILESCU M.: Sampling and reconstruction with adaptive meshes. In *IEEE CVPR* (1991), 70–75. 2

[Wh98] WHITAKER R.: A level-set approach to 3d reconstruction from range data. *IJCV* (1998), 203–231. 2

Apppendix

To reconstruct a surface from an oriented point set, we compute the Fourier coefficients of the characteristic function by approximating surface integration by a Monte-Carlo summation. To do this we need to define complex vector functions whose divergences are equal to the complex exponentials.

In this appendix, we show that although there are many complex vector functions whose divergences are equal to a given complex exponential, if we would like the reconstruction to satisfy two simple properties, these vector functions have to be unique. In particular, we show that if we would like the reconstruction process to satisfy:

- The contribution of a point to the reconstructed implicit is independent of the contribution of any other point, and
- The reconstruction commutes with translation and rotation (translations and rotations of the point set result in corresponding translations and rotations of the implicit)

then the vector function giving rise to the $\vec{l}$-th Fourier coefficient can only be:

$$\vec{F}_{\vec{l}}(\vec{q}) = \left(i\vec{l}/\|\vec{l}\|^2\right) e^{-i\langle \vec{l},\vec{q}\rangle}.$$

To prove this, we recall that given a point set $\{\vec{p}_i,\vec{n}_i\}$, the $\vec{l}$-th Fourier coefficient of the implicit function is defined by:

$$\hat{F}(\vec{l}) = \sum_i \left\langle \vec{F}_{\vec{l}}(\vec{p}_i),\vec{n}_i \right\rangle.$$

Since we assume that the contribution of any point is independent of any other point, it suffices to prove the uniqueness property for the case when the point set consists of a single oriented point $\{\vec{p},\vec{n}\}$.

The condition that the reconstructed function commutes with translation implies that if $\hat{F}(\vec{l})$ is the $\vec{l}$-th Fourier coefficient of the reconstruction obtained using the oriented point $\{\vec{p},\vec{n}\}$, then $e^{-i\langle \vec{l},\vec{p}_0\rangle}\hat{F}(\vec{l})$ is the $\vec{l}$-th Fourier coefficient of the reconstruction obtained using the oriented point $\{\vec{p}+\vec{p}_0,\vec{n}\}$. Since this must be true for any normal vector $\vec{n}$, it follows that the function $\vec{F}_{\vec{l}}(\vec{p})$ can be factored as the product $\vec{F}_{\vec{l}}(\vec{p}) = \vec{G}(\vec{l})e^{-i\langle \vec{l},\vec{p}\rangle}$ with $\vec{G}(\vec{l}) \in \mathbf{C}^3$.

Moreover, since the divergence of $\vec{F}_{\vec{l}}(\vec{p})$ has to be equal to $e^{-i\langle \vec{l},\vec{p}\rangle}$, it follows that $-i\langle \vec{G}(\vec{l}),\vec{l}\rangle = 1$. Thus, we know that:

$$\vec{G}(\vec{l}) = i\vec{l}/\|\vec{l}\|^2 + \vec{l}^{\perp} \qquad (2)$$

where $\vec{l}^{\perp}$ is a vector perpendicular to $\vec{l}$.

Since we also want the reconstruction to commute with rotation, this implies that if $\hat{F}(\vec{l})$ is the $\vec{l}$-th Fourier coefficient of the reconstruction obtained using the oriented point $\{\vec{p},\vec{n}\}$, then $\hat{F}(R(\vec{l}))$ is the $\vec{l}$-th Fourier coefficient of the reconstruction obtained using the oriented point $\{R(\vec{p}),R(\vec{n})\}$, for any rotation R.

Thus, in order for the reconstruction process to commute with rotation, the function $\vec{G}(\vec{l})$ must satisfy the property:

$$R(\vec{G}(\vec{l})) = \vec{G}(R(\vec{l})).$$

Clearly, the function $\vec{G}(\vec{l}) = i\vec{l}/\|\vec{l}\|^2$ satisfies this property. On the other hand, if we take R to be a rotation about the vector $\vec{l}$, we get $R(\vec{G}(\vec{l})) = \vec{G}(\vec{l})$. Thus, it follows that $\vec{l}^{\perp}$ (in Equation 2) must be zero, and hence the only function satisfying translation and rotation commutativity is:

$$\vec{F}_{\vec{l}}(q) = \left(i\vec{l}/\|\vec{l}\|^2\right) e^{-i\langle \vec{l},\vec{q}\rangle}.$$

© The Eurographics Association 2005.

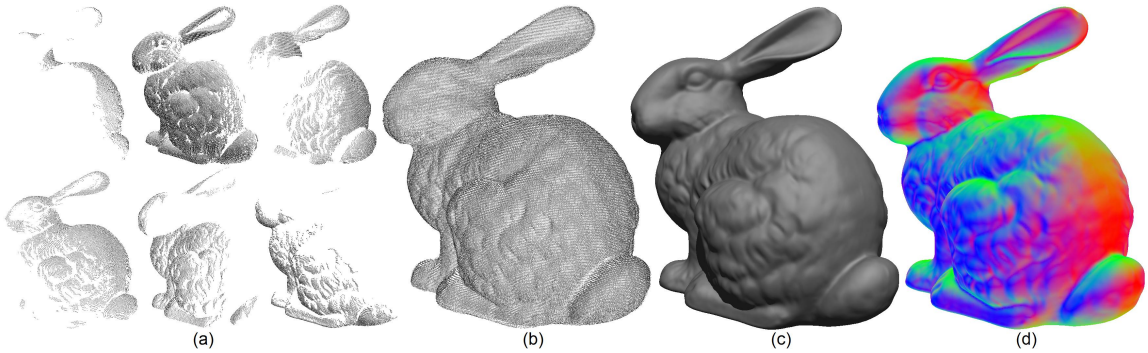

Figure 11: *A reconstruction of the Stanford bunny model from range scans: Some of the initial scans of the model, obtained from different view points, are shown in (a). The complete point set, obtained by merging the scans, is shown in (b). The reconstructed surface is shown in (c). And a visualization of the surface, obtained by mapping the normal coordinates to RGB is shown in (d). Note that although the overlap of different scans results in a non-uniform distribution of points, our weighting scheme correctly assigns sampling densities to the input points, resulting in an accurate and smooth reconstruction of the surface (as demonstrated by the smooth variation of normals seen in (d)). (Model courtesy of Stanford University Computer Graphics Laboratory.)*

© The Eurographics Association 2005.