Reconstructing Curved Surfaces From Specular Reflection Patterns Using Spline Surface Fitting of Normals

Mark A. Halstead[†‡] Brian A. Barsky[‡§] Stanley A. Klein[§] Robert B. Mandell[§]

University of California at Berkeley

ABSTRACT

We present an algorithm that reconstructs a three-dimensional surface model from an image. The image is generated by illuminating a specularly reflective surface with a pattern of light. We discuss the application of this algorithm to an important problem in biomedicine, namely the measurement of the human cornea, although the algorithm is also applicable elsewhere. The distinction between this reconstruction technique and more traditional techniques that use light patterns is that the image is formed by *specular* reflection. Therefore, the reconstruction algorithm fits a surface to a set of normals rather than to a set of positions. Furthermore, the normals do not have prescribed surface positions. We show that small surface details can be recovered more accurately using this approach. The results of the algorithm are used in an interactive visualization of the cornea.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling; J.6 [Computer-Aided Engineering]: Computer-Aided Design.

Keywords: Surface reconstruction, videokeratography, corneal modeling, photogrammetry, normal fitting.

1 INTRODUCTION

A problem of particular interest to the computer graphics community is how to construct accurate computer models of existing objects. One use of these reverse-engineered models is to populate rendered scenes with realistic objects. Another important use is for computer-aided visualization, where the models are displayed in ways that expose information for analysis. An example is medical imaging, in which models of a patient's anatomy are first extracted from measurements made by a variety of scanners (for example MRI or ultrasound [7]). The models are then displayed in various forms for diagnosis. The success of this approach has made medical imaging one of the more important applications of computer graphics and modeling. In addition to displaying the reconstructed models, they can be used to provide a valuable starting point for other operations. For example, in the area of computer-aided design (CAD), an architect can input a model of an existing building, then design and visualize modifications by editing the existing structure. Reverse-engineering is also studied in disciplines other than computer graphics and modeling. For example, in the machine vision community, it appears under the guise of automatic object recognition.

This paper presents a solution to an important reverse-engineering problem in biomedicine: constructing a computer model of the human cornea. The cornea is the outer layer of the eye, and plays the primary role in focusing images on the retina. The algorithm that we have devised to construct a model of the cornea is of interest to the graphics community for the following reasons:

- We derive the surface model by applying backward ray-tracing to simulate reflection at a specular surface and the resulting virtual image.
- We use a variation on the standard B-spline surface representation to increase the efficiency of the backward ray-tracing by at least an order of magnitude.
- We solve a problem of fitting a surface to a set of normals at unprescribed locations.
- The algorithm we present here has significantly advanced the frontier of corneal modeling and visualization.

Building a model of a physical object usually proceeds in two stages: data capture, and construction of the model from the data. Data capture takes many forms. For a survey of techniques, the reader is referred to [6, 14]. A common technique uses correspondences between two or more images of an object taken from different positions [5]. Stereopsis or depth disparity allows the recovery of an approximate depth map for the object. Depth is also recovered by a different class of techniques that use structured light. In this approach, the object is illuminated with a pattern to form an image. The geometric relationship between the light source, the object, and the image recording device is sufficient to determine depth. In some cases, estimates of surface orientation are also provided. Examples include laser rangefinders, slit-ray or grid projectors, and Moiré pattern generators [1, 22, 26].

In most cases, the data is returned as samples of surface position. A model is built by fitting the captured positional data with a surface [8, 11, 12, 20, 21, 23]. Problems faced at this stage include surface discontinuities, and noisy or missing data. Terzopoulos [25] presents a robust algorithm

[†]Apple Computer, 1 Infinite Loop M/S 301-3J, Cupertino, CA 95014.

[‡]Computer Science Division, University of California, Berkeley, CA 94720-1776. www.cs.berkeley.edu/projects/optical/

[§]School of Optometry, University of California, Berkeley, CA 94720-2020.

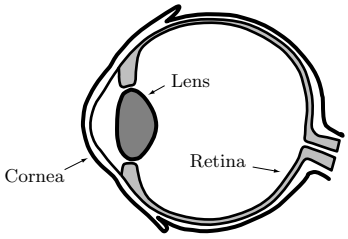


Figure 1: Cross-section of the eye.

which handles these problems using a multigrid approach. In [24], the model is built from implicit algebraic surfaces. The problem of joining positional data taken from different measurements is addressed in [27]. Hoppe [13] discusses an algorithm for recovering surfaces with complex topology from range data.

The important distinction between these fitting problems and the one presented in this paper is that we measure surface *orientation* rather than surface position. This presents a number of interesting challenges and leads to a new approach to surface fitting. The problem differs from those faced by the vision community (such as shape-from-shading [5]) in that we rely on the specular reflection of illumination at the surface in order to reconstruct very small surface features.

As shown in Figure 1, the cornea is the transparent tissue that forms the foremost surface of the eye. Refraction at the air/cornea interface accounts for approximately three-quarters of the focusing power of the eye. Consequently, the exact shape of the cornea is critical to visual acuity. Even subtle deviations of corneal shape can have a significant impact on vision. Measurements must be at the level of submicrons to be meaningful. It is the high precision needed for the measurements that leads us to measure corneal orientation rather than position, and that makes surface reconstruction a challenging problem.

Vision problems arise when the cornea is asymmetric or is too peaked or flat to focus light uniformly on the retina. Eyeglasses and contact lenses address the problem by prerefracting the light so that the overall effect of the corrective lens plus cornea is more uniform. Recent surgical techniques change the degree of refraction by adjusting the shape of the cornea directly. *Radial keratotomy* (RK) and *astigmatic keratotomy* (AK) reduce the curvature of the cornea by weakening the structure with radial cuts; *photorefractive keratectomy* (PRK) and *laser in-situ keratomileusis* (LASIK) achieve a similar result by ablating corneal material with an excimer laser.

All these procedures benefit from having accurate models of corneal shape. Until recently, however, models used in the optometric community have assumed that the cornea is spherical, ellipsoidal, toric (like a section cut from a torus), or has radial symmetry. On a gross scale, most corneas do have these smooth shapes. However, we wish to measure, and model, deviations from these simple shapes that are on the order of microns and that contribute significantly to the refractive power of the surface.

Ideally, we would like to have an algorithm that, in clinical practice, can scan a patient's cornea and display usable results almost immediately. We have developed an interactive program that presents a good approximation to the shape of a patient's cornea within seconds, and then continues to refine the display until micron accuracy is achieved.

2 MEASURING SURFACE SHAPE

Depth from binocular disparity does not work well for measuring corneal shape, since the surface has no distinctive variation in texture. This makes it difficult to identify correspondences between multiple images. Therefore, we consider measurement techniques in which the surface is illuminated by a pattern of light. The resulting image is recorded by a scanner such as a CCD array. There are two ways to apply this technique, which we will refer to as approach A and approach B.

Applications of approach A are more typical. Often referred to as *raster photogrammetry*, they require a pattern of one or more lines to be projected along a known direction onto a diffuse surface [4, 27, 30]. The projected pattern acts as a secondary diffuse source, which is viewed by the scanner along a direction off-axis to the direction of projection. The observed distortion of the lines indicates the variation in surface distance from the projector, and fully determines the position of a set of sample points. This approach is used, for example, by the Cyberware laser scanner [27]. Lately, interest has focused on fitting the data points with surface models, and combining the results from multiple scans [27].

The approach taken by applications of type Bis for a specular surface, rather than diffuse, near a diffusely emitting source pattern. The scanner captures the virtual image of the pattern caused by specular reflection at the surface. As with approach A, the shape of the surface affects the scanned image. However, the relationship between the shape and the image is harder to define, as it now depends on surface orientation as well as position.

One advantage of approach B is that very small deviations in surface orientation cause large changes in the image. There is no such magnification possible in approach A — a change in orientation has no effect on the scanned image, and a change in position causes changes only of the same magnitude. The accuracy is therefore limited by the resolution of the scanning device. We believe that, with a good reconstruction algorithm, approach B allows submicron level detail to be recovered. This paper describes such a reconstruction algorithm.

2.1 Videokeratography

Fortunately, the cornea in its normal state is covered by a thin layer of tears and presents a specular surface. This means that we can conveniently image the surface using approach B. Over the last few years, simple observation devices based on this approach and used in the optometric community have evolved into the *videokeratograph* [15, 16, 19, 29, 31]. This instrument contains a video camera to capture a digital image which is analyzed by an on-board computer. Although there are variations among systems, a standard arrangement is to have the source pattern painted on the inner surface of a cone. The cone has a hole in its apex through which a system of lenses and a CCD array capture the reflected image (Figure 2). The most common source pattern is alternating black and white concentric rings. Recently, however, we have been experimenting with a prototype source pattern, not commercially available, that resembles a dartboard. We show later how such a pattern allows a more accurate reconstruction.

In preparation for measurement, the patient's line of sight is aligned with the axis of the cone and the image is brought into focus. After capturing an image (Figure 3), the videokeratograph performs a processing step to locate prominent

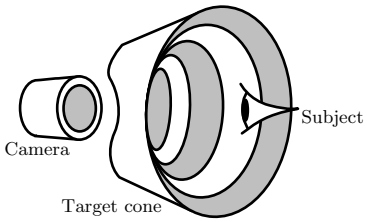


Figure 2: Videokeratograph.

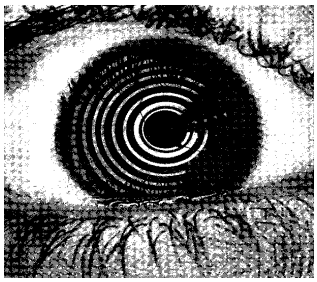


Figure 3: Captured videokeratograph image.

image features. The nature of these features varies between source patterns. For both the concentric ring and the dartboard sources, the feature set includes the edges between regions of black and white in the image. These edges are located with standard image processing techniques and are discretely sampled. For the dartboard source, the crossings created by the junction of four black and white patches form an additional feature set. The aim of our reconstruction algorithm is to generate a model of the corneal surface from the image feature positions and the geometry of the videokeratograph. Previous algorithms have failed to recover continuous, accurate surface models [4, 10, 15, 16, 28, 30]. Our algorithm satisfies these goals.

3 RECONSTRUCTION ALGORITHM

The reconstruction algorithm inputs a set of feature samples from a videokeratograph image. The goal is to find a surface that, if placed in the videokeratograph, would create the same image. With certain assumptions and constraints, we claim that this surface is a good fit to the original cornea.

In order to find the surface, we use a simulation of the videokeratograph. As explained in Section 2.1, light from the source pattern is reflected at the corneal surface and gathered by a system of lenses to form the videokeratograph image. We simulate this process by backward ray-tracing, as illustrated in Figure 4. In the simulation, the system of lenses is replaced by its equivalent nodal point[1], which becomes the center of projection. The video CCD array becomes the image plane, with which we associate the coordinate system (u, v).

In Section 2.1, we noted that image processing techniques are used to extract features such as edges and crossings from the videokeratograph image. The positions of these features in (u, v) are sampled to form the set $\{\mathbf{f}_i : i = 1 \ldots k\}$. Our algorithm relies on the fact that features in the image correspond to features in the source. For example, a sample point taken from an edge in the image is the reflection of some point on an edge in the source. If we trace a ray from the sample point $\mathbf{f}_i$ on the image to the surface (through the nodal point) we can identify a ray that goes from the surface to the "correct" point on the source (that is, some point on the corresponding source feature). In Figure 4, this second ray is shown as a dashed line. These two rays — the incident ray $\hat{\mathbf{i}}_i$ and the modified reflected ray $\hat{\mathbf{r}}_i^*$, where the hat indicates normalization — define a modified surface normal vector:

$$\hat{\mathbf{n}}_i^* = \frac{\hat{\mathbf{r}}_i^* - \hat{\mathbf{i}}_i}{\left|\hat{\mathbf{r}}_i^* - \hat{\mathbf{i}}_i\right|}.$$

The modified reflected ray and the modified normal vector may differ from the actual reflected vector $\hat{\mathbf{r}}_i$ and the actual normal vector $\hat{\mathbf{n}}_i$. In this case, we attempt to adjust the surface so that it interpolates the modified normal.

We determine a modified normal vector for each image sample and fit the set of normals with a new surface. Because the backward rays intersect the new surface at new positions, the set of modified normals we just fit is now incorrect, so we must recompute them and repeat the fitting step. This leads to an iterative process which we initialize by taking a guess at the shape of the cornea. Each iteration consists of a *normal fitting* phase and a *refinement phase*. The normal fitting phase determines the set of modified normals using the current surface and fits them with a new surface. The refinement phase adds degrees of freedom to the surface model as needed for a more accurate fit.

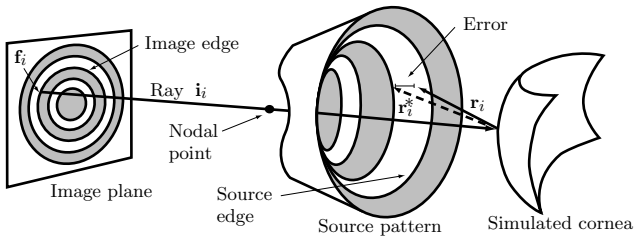


Figure 4: Simulation using ray-tracing.

3.1 The Normal Fitting Phase

For each sample $\mathbf{f}_i$, we determine the modified normal $\hat{\mathbf{n}}_i^*$ using the current surface. We define the *range* of $\mathbf{f}_i$ to be the set of points in the source pattern that could be imaged to $\mathbf{f}_i$. This set may form a curve, such as the entire source edge in the above example, or it may contain a single point. The latter case arises when the dartboard pattern is used, because crossings can be located exactly in the source.

Using backward ray-tracing from $\mathbf{f}_i$ and the current surface, we find where the reflected ray $\hat{\mathbf{r}}_i$ intersects the source pattern. We then determine which point in the range lies nearest the intersection. The ray from the surface to this point is $\hat{\mathbf{r}}_i^*$.

The set $\{\hat{\mathbf{n}}_i^* : i = 1 \ldots k\}$ is viewed as a set of normal vector constraints which, together with a positional constraint discussed in the next section, is fit with a new surface S^*. The details are given in Section 3.4 after we have described the surface representation scheme.

[1]For our purposes, the action of a camera with one or more lenses is adequately modeled by a pinhole camera positioned at the nodal point

3.2 Convergence and Uniqueness of Solution

The search process clearly must be iterative, since each normal fitting phase changes not only the normal vectors of the surface but the position, which causes the rays from the image plane to intersect at new locations. We have no formal proof that our search algorithm will always converge, but in all trial runs we have observed rapid and stable convergence to a good solution.

For a given set of features, there may be zero, one, or many surfaces that generate matching images. The result depends on the distribution of image features and the representation chosen for the surface model. Fortunately, we are working in a restricted problem domain, in which we can assume that the cornea has a smooth, regular shape. This assumption is based on the fact that the cornea is a pliable tissue subject to internal pressure. Based on this assumption, we formulate the search process to favor smoothly varying surfaces. With the further assumption that the number of degrees of freedom in the surface model is related to its smoothness, we begin the search with a surface of few degrees of freedom, and incrementally add degrees of freedom until a satisfactory solution is found.

Although this algorithm may not be a rigorous statement of our goal, it has given more than satisfactory results in practice. Note that we use far more features than degrees of freedom, so that the search for a match is overconstrained.

Section 3.5 will discuss efficient methods for adding degrees of freedom to the model using refinement.

To further reduce the number of possible solutions, we impose one or more interpolation constraints. The constraint that we use most often is to fix the position of the apex of the cornea. Some videokeratograph systems have attachments that can directly measure this position. If the information is not directly available, it can be estimated from the image and the known focal length of the camera lens. The algorithm then generates an initial surface that interpolates this point, and maintains the interpolation constraint for the remainder of the search.

3.3 Surface Representation

The representation of S is carefully chosen to allow the efficient execution of the normal fitting phase. At first glance, conventional CAGD wisdom would suggest the use of a parametric polynomial patch scheme — such as tensor product B-splines with control points in $\mathbb{R}^3$ — to define the position of S in (x, y, z). One drawback of this scheme is that numerically expensive algorithms such as root finding are required to compute the ray/surface intersections during the backward ray-tracing stage.

In contrast, we use a representation scheme that allows the point of intersection of a ray to be determined simply by taking a linear combination of scalar control points. Furthermore, the coefficients of the linear combination remain constant throughout the search, and are computed only once, at initialization.

Figure 5 illustrates how the scheme works. By suitably scaling the feature positions $\mathbf{f}_i$, we can assume that the image plane lies at $z = -1$ and that $u \equiv x, v \equiv y$. We define $D(u, v)$ to be the z-coordinate of the point of intersection between the surface and the ray that originates at (u, v) and passes through the nodal point. The coordinates of this point are

$$x = -uD(u, v) \quad (1)$$
$$y = -vD(u, v) \quad (2)$$
$$z = D(u, v). \quad (3)$$

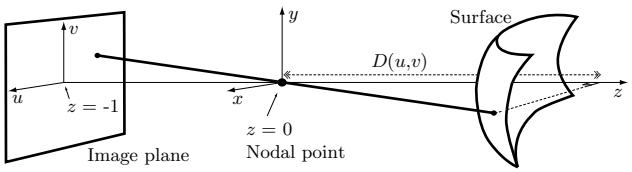


Figure 5: Surface representation scheme.

These equations define the surface S in (x, y, z), parameterized by image plane coordinates (u, v). Furthermore, the representation is ideal for the backward ray-tracing task, since, by definition, the equations directly give the intersection of each ray with S.

In the current implementation, $D(u, v)$ is represented by a set of biquintic tensor product B-spline patches, with uniform knot spacing and no boundary conditions [3]. A surface consisting of $(m-5) \times (n-5)$ patches is defined by an $m \times n$ array of control points $\{P_{1,1}, \ldots, P_{m,n}\}$ in $\mathbb{R}^1$. From the definition of (x, y, z), we find that the x-coordinate patches of S are degree 6 polynomials in u and degree 5 in v; the y-coordinate patches are degree 5 in u and degree 6 in v; and the z-coordinate patches are degree 5 in both u and v. The surface S can be rendered using standard techniques for polynomial patch surfaces.

We have chosen to use biquintic patches rather than a lower degree representation because of the high degree of smoothness exhibited by the cornea. Furthermore, one of our scientific visualization tasks is to display curvature maps of the surface. Since the formula for curvature involves second partial derivatives of surface position, a bicubic representation allows cusps in curvature, whereas a biquintic formulation provides for smoother joins at patch boundaries.

Since the feature positions $\{\mathbf{f}_i : i = 1 \ldots k\}$ are defined by the input image, their (u, v) coordinates, which are used as ray origins for backward ray-tracing, remain fixed throughout the search. For fixed (u, v), the functions x, y, z given by (1)–(3) are linear combinations of the control points $\{P_{1,1}, \ldots, P_{m,n}\}$. This is a consequence of the B-spline function $D(u, v)$ being linear in $\{P_{1,1}, \ldots, P_{m,n}\}$ for fixed (u, v). For each of the k features, we can evaluate the basis functions of $D(u, v)$ at the feature's coordinate and multiply by $-u$ (for x) or $-v$ (for y) to find the coefficients that multiply each of the control points. Collecting these coefficients together in matrix form, we can write

$$\begin{aligned} x &= M_x[P_{1,1}, \ldots, P_{m,n}]^T \\ y &= M_y[P_{1,1}, \ldots, P_{m,n}]^T \\ z &= M_z[P_{1,1}, \ldots, P_{m,n}]^T, \end{aligned}$$

where M_x, M_y and M_z are $k \times mn$ matrices that are precomputed and stored using a sparse matrix representation for efficient evaluation.

Similarly, we derive matrices M_x^u and M_x^v such that $\partial x/\partial u = M_x^u[P_{1,1}, \ldots, P_{m,n}]^T$ and $\partial x/\partial v = M_x^v[P_{1,1}, \ldots, P_{m,n}]^T$. Together with matrices M_y^u, M_y^v, M_z^u, and M_z^v for y and z, these are used to find expressions for the surface tangent vectors in the u and v directions:

$$\begin{aligned} \mathbf{t}_u &= \partial \mathbf{S}(u, v)/\partial u \\ \mathbf{t}_v &= \partial \mathbf{S}(u, v)/\partial v \\ \text{where } \mathbf{S}(u, v) &= [x(u, v), y(u, v), z(u, v)]. \end{aligned}$$

The surface normal required for computing the reflected ray during ray-tracing is given by $\mathbf{t}_u \times \mathbf{t}_v$.

3.4 Solving the Normal Fitting Problem

In Section 3.1, the normal fitting problem was defined as finding the surface S^* that fits the prescribed surface normals $\{\hat{\mathbf{n}}_i^* : i = 1 \ldots k\}$ subject to one or more interpolation constraints.

Let (x_c, y_c, z_c) be a point interpolated by the surface. Using (1)–(3), we find that the constraint is satisfied if

$$D(-x_c/z_c, -y_c/z_c) = z_c.$$

Using the same technique employed to construct the matrix M_z in the previous section, we determine coefficients $\{c_{1,1}, \ldots, c_{m,n}\}$ such that

$$D(-x_c/z_c, -y_c/z_c) = [c_{1,1}, \ldots, c_{m,n}][P_{1,1}, \ldots, P_{m,n}]^T = z_c.$$

This constraint equation is linear in the control points, and some number, say s, of interpolation constraints are expressed in matrix form as

$$C[P_{1,1}, \ldots, P_{m,n}]^T = [z_1, \ldots, z_s]^T,$$

where C is an $s \times mn$ matrix.

Let N be the $mn \times t$ matrix whose columns span the null space of C. Given an input surface S with control points $\{P_{1,1}, \ldots, P_{m,n}\}$ that satisfies the constraints, all other surfaces S^* that satisfy the constraints, and hence are possible solutions to the normal fitting problem, have control points given by

$$[P_{1,1}^*, \ldots, P_{m,n}^*]^T = [P_{1,1}, \ldots, P_{m,n}]^T + N[Q_1, \ldots, Q_t]^T. \quad (4)$$

Now consider a feature $\mathbf{f}_i$ with image plane coordinates (u_i, v_i). The modified surface normal computed for this feature is $\hat{\mathbf{n}}_i^* = (n_{i,x}^*, n_{i,y}^*, n_{i,z}^*)$. We can require S^* to have a scalar multiple of this normal by using the pair of constraints

$$\begin{aligned} \hat{\mathbf{n}}_i^* . \mathbf{t}_u|_{u_i,v_i} &= 0, \\ \text{and } \hat{\mathbf{n}}_i^* . \mathbf{t}_v|_{u_i,v_i} &= 0. \end{aligned}$$

Using the matrices from the previous section, and defining $diag(\alpha_1, \ldots, \alpha_j)$ to be the $j \times j$ matrix with $\alpha_1, \ldots, \alpha_j$ on the diagonal and zeroes elsewhere, we write the pairs of constraints for all features as

$$\begin{aligned} \Big[diag(n_{1,x}^*, \ldots, n_{k,x}^*)M_x^u + diag(n_{1,y}^*, \ldots, n_{k,y}^*)M_y^u + \\ diag(n_{1,z}^*, \ldots, n_{k,z}^*)M_z^u\Big] [P_{1,1}^*, \ldots, P_{m,n}^*]^T &= 0 \\ \Big[diag(n_{1,x}^*, \ldots, n_{k,x}^*)M_x^v + diag(n_{1,y}^*, \ldots, n_{k,y}^*)M_y^v + \\ diag(n_{1,z}^*, \ldots, n_{k,z}^*)M_z^v\Big] [P_{1,1}^*, \ldots, P_{m,n}^*]^T &= 0 \end{aligned}$$

These normal constraint equations are combined to form the simpler expression

$$M[P_{1,1}^*, \ldots, P_{m,n}^*]^T = 0. \quad (5)$$

Substituting (4) into (5), we arrive at a system of linear equations which is solved in a least squares sense for $[Q_1, \ldots, Q_t]$:

$$MN[Q_1, \ldots, Q_t]^T = -M[P_{1,1}, \ldots, P_{m,n}]^T. \quad (6)$$

The final control points $[P_{1,1}^*, \ldots, P_{m,n}^*]$ of the new surface S^* are determined by substitution of $[Q_1, \ldots, Q_t]$ into (4).

The $k \times t$ matrix MN in (6) has the same number of rows as there are image features, and the number of columns is no more than the number of surface control points minus one. To ensure a sensible fit to the data, we overconstrain the system by using as many as 20 to 30 times as many features as control points. This is an attempt to smooth out errors in the feature location process. The least squares solution to the resulting rectangular system is found with standard numerical algorithms. In the current implementation, the sparsity of MN is not used to advantage.

3.5 Refinement

The goal of our interactive system is to rapidly display an initial, approximate solution which is then improved incrementally. Therefore, the initial search iterations should execute quickly at the expense of accuracy. Since each iteration is dominated by the solution of (6), this goal is satisfied by performing the initial iterations with a small number of control points and features. Accuracy is improved at the expense of iteration time by increasing the number of control points and features, which is consistent with the algorithm discussed in Section 3.2 in regard to uniqueness of solution.

In our current implementation, which uses biquintic B-spline patches with uniform knot spacing, the number of control points is increased by subdividing each patch into four subpatches. A new knot is inserted at the midpoint of each knot interval in both u and v. The new control points are derived using simple linear combinations of the old, with coefficients given by an application of the Oslo algorithm [3, 9].

The search begins with a single patch model of the surface. Refinement is performed when the mean change in angle between normals at successive iterations falls below a given threshold. The model moves through representations using $(4, 16, 64, \ldots)$ patches until the maximum change in normal falls below a predetermined threshold at a predetermined level of refinement.

A typical feature set contains over 5000 samples. For low subdivision levels, this is more than we need, so a subset of features is used. This subset is chosen so that the image is uniformly sampled.

4 RESULTS

To test the algorithm, we need to run it on surfaces whose shape is accurately known. Unfortunately, it is difficult to manufacture interesting test cases. For this reason, we have tested the algorithm on both data collected from real objects and data generated synthetically. The synthetic data is generated automatically from various surface definitions by a software simulation of the videokeratograph.

Figures 9a-10d show some frames illustrating the progress of the algorithm. The results of the search process are displayed to the user after each iteration. The algorithm is formulated so that a good approximation to the final answer is reached in a few seconds, so the user can start analyzing the results immediately. A more accurate picture evolves over the next few minutes.

In Figure 9, there are four frames showing the patches converging to a solution. The input data is a simple ellipsoid with axis radii of 8mm, 9mm, and 10mm. The image data for this example was synthetically generated and is shown in Figure 6. For illustration purposes, we show the exact ellipsoid (lower surface) plus the current solution offset above. However, since the difference in shape is so small, we magnify this difference by a factor of 20 in order to better visualize the progress of the algorithm. The surface colors indicate the logarithm (base 10) of the distance in

millimeters between the current solution and the exact ellipsoid. In the final frame, we can see that good convergence has been achieved. We measure the error as the distance in z between the two surfaces, computed at a large number of sample points in the x, y plane. The RMS error for this example was 9.2×10^{-6}mm, which is 0.0092 microns. This extremely high accuracy is typical of all synthetic data sets we have tried.

Figure 10 shows frames from another run of the algorithm. The input data in this example is again synthetically generated, and is shown in Figure 7. The aim here is to simulate *keratoconus*, which is a condition in which the cornea has a local region of high curvature [2, 17, 18]. The surface is generated from a sphere with a rotationally symmetric bump grafted onto it. The bump and the sphere meet with curvature continuity. The curvature at the peak of the bump is significantly greater than the curvature of the sphere. This situation has not been handled very well by existing algorithms. Our algorithm, however, has no difficulty in finding an accurate solution. Note that the bump rises only approximately 20 microns above the sphere. This is a deviation of about 0.2 percent of the radius of the sphere. However, the bump causes large deviations in the image rings (Figure 7), demonstrating that we can record smaller deviations using an image formed by specular reflection than one formed by diffuse reflection.

Rather than color encode the distance between the current solution and the actual surface, in Figure 10 we have color encoded the separation between the current surface and a sphere whose radius is the same as that of the input test surface. This form of rendering illustrates one of our visualization techniques, which is to display the surface separation from a best-fitting ellipsoid. This enhances the deviations so that the bump, which is positionally very close to the sphere, becomes noticeable. In this example, we get extremely high positional accuracy of 0.013 microns.

Figure 8 illustrates the results of the algorithm run on real data taken from a cornea. In this case, we cannot report accuracy information because the true shape is unknown. Nonetheless, we can render it with our in-house scientific visualization software package. Figure 8 shows the surface with pseudo-color representing Gaussian curvature (and the height information in the image is simply the true height of the 3-D surface). The red area on the left depicts a local area of high Gaussian curvature. The vectors correspond to the direction of minimum curvature at each point on the surface. This image demonstrates how effective the use of curvature can be in conveying subtle changes in shape.

We have run the algorithm on real data measured from physical ellipsoids of known radii. In these runs, the final accuracy lies in a range of 0.9-1.5 microns of mean error in z. This is still extremely accurate, but it is significantly larger than the error in the synthetic runs. We conclude that the error is introduced, not by the algorithm, but by the feature extraction algorithm and in the measurements we have made of the physical videokeratograph geometry (such as distance between rings, etc.). We are currently addressing these issues.

In all these runs, the final surface consists of 8×8 patches. This gives adequate accuracy, although there is no reason why we cannot go to the next level of 16×16 patches. Beyond that, we reach the limits of the feature sampling process. In the source patterns we currently use, the features are not uniformly spread across the image but are concentrated along boundaries between areas of black and white rings. This limits how small the patches can be, because if a patch falls between feature clusters it will be unconstrained (except by continuity between adjacent patches).

5 CONCLUSION

We have presented an algorithm that reconstructs the shape of the cornea from a single videokeratograph image. The algorithm is interesting because it fits a surface to a set of normals rather than to a set of positions. Furthermore, the normals are not associated with spatial positions as in standard normal fitting problems. This distinguishes it from more typical surface reconstruction problems. The normal fitting is necessary because the surface imaging technique uses reflection from a specular surface. This improves its ability to detect small variations in surface position because surface orientation is a more sensitive indicator of shape variations than is surface position. This technique can be applied to objects other than the human cornea, and any applications that require high accuracy would be candidates. However, we have made some assumptions about the surface that allow us to proceed with little direct information about the corneal position. For example, we only require a single positional constraint. These assumptions are valid in the case of corneas. For other more general objects where these assumptions could not be made, more positional measurements may be needed to provide additional constraints.

REFERENCES

[1] Altschuler, M., Altschuler, B., and Taboada, J. Laser electro-optic system for rapid three-dimensional (3-D) topographic mapping of surfaces. *Optical Engineering 20*, 6 (1981), 953–961.

[2] Barsky, B. A., Mandell, R. B., and Klein, S. A. Corneal shape illusion in keratoconus. *Invest Opthalmol Vis Sci 36 Suppl.:5308* (1995).

[3] Bartels, R. H., Beatty, J. C., and Barsky, B. A. *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling*. Morgan Kaufmann, 1987.

[4] Belin, M. W., Litoff, D., and Strods, S. J. The PAR technology corneal topography system. *Refract Corneal Surg 8* (1992), 88–96.

[5] Blake, A., and Zisserman, A. *Visual Reconstruction*. MIT Press, 1987.

[6] Bolle, R., and Vemuri, B. On three-dimensional surface reconstruction methods. *IEEE Trans. PAMI 11*, 8, 840–858.

[7] Brinkley, J. Knowledge-driven ultrasonic three-dimensional organ modeling. *IEEE Trans. PAMI 7*, 4, 431–441.

[8] Cheng, F., and Barsky, B. A. Interproximation: Interpolation and approximation using cubic spline curves. *Computer-Aided Design 23*, 10 (1991), 700–706.

[9] Cohen, E., Lyche, T., and Riesenfeld, R. Discrete B-splines and subdivision techniques in computer aided geometric design and computer graphics. *Computer Graphics and Image Processing 14* (1980), 87–111.

[10] Doss, J. D., Hutson, R. L., Rowsey, J. J., and Brown, D. R. Method for calculation of corneal profile and power distribution. *Arch Ophthalmol 99* (1981), 1261–5.

[11] Favardin, C. *Determination automatique de structures geometriques destinees a la reconstruction de courbes et de surfaces a partir de donnees ponctuelles*. PhD thesis, Universite Paul Sabatier, Toulouse, France, 1993.

[12] Goshtasby, A. Surface reconstruction from scattered measurements. *SPIE 1830* (1992), 247–256.

[13] Hoppe, H., DeRose, T., Duchamp, T., Halstead, M., Jin, H., McDonald, J., Schweitzer, J., and W., S. Piecewise smooth surface reconstruction. *Computer Graphics (SIGGRAPH '94 Proceedings)* (July 1994), 295–302.

[14] Jarvis. A perspective on range finding techniques for computer vision. *IEEE Trans. PAMI 5*, 2 (1983), 122–139.

[15] Klyce, S. D. Computer-assisted corneal topography, high-resolution graphic presentation and analysis of keratoscopy. *Invest Ophthalmol Vis Sci 25* (1984), 1426–35.

[16] Koch, D. D., Foulks, G. N., and Moran, T. The corneal eyesys system: accuracy, analysis and reproducibility of first generation prototype. *Refract Corneal Surg 5* (1989), 424–9.

[17] Krachmer, J. H., Feder, R. S., and Belin, M. W. Keratoconus and related noninflammatory corneal thinning disorders. *Surv. Ophthalmol 28*, 4 (1984), 293–322.

[18] Maguire, L. J., and Bourne, W. D. Corneal topography of early keratoconus. *Am J Ophthalmol 108* (1989), 107–12.

[19] Mammone, R. J., Gersten, M., Gormley, D. J., Koplin, R. S., and Lubkin, V. L. 3-D corneal modeling system. *IEEE Trans Biomedical Eng 37* (1990), 66–73.

[20] Moore, D., and Warren, J. Approximation of dense scattered data using algebraic surfaces. Tech. rep., TR 90-135, Rice University, 1990.

[21] Pratt, V. Direct least-squares fitting of algebraic surfaces. *SIGGRAPH '87 Conference Proceedings* (1987), 145–152.

[22] Sato, Y., Kitagawa, H., and Fujita, H. Shape measurement of curved objects using multiple slit-ray projections. *IEEE Trans. PAMI 4*, 6 (1982), 641–649.

[23] Schmitt, F., Barsky, B. A., and Du, W.-H. An adaptive subdivision method for surface fitting from sampled data. *SIGGRAPH '86 Conference Proceedings* (1986), 179–188.

[24] Taubin, G. An improved algorithm for algebraic curve and surface fitting. In *Proc. 4th International Conf. on Comp. Vision, Berlin* (1993), pp. 658–665.

[25] Terzopolous, D. Regularization of inverse visual problems involving discontinuities. *IEEE Trans. PAMI 8* (1986), 413–424.

[26] Topa, L., and Schalkoff, R. An analytical approach to the determination of planar surface orientation using active-passive image pairs. *Computer Vision, Graphics, and Image Processing 35* (1994), 404–418.

[27] Turk, G., and Levoy, M. Zippered polygon meshes from range images. *Computer Graphics (SIGGRAPH '94 Proceedings)* (1994), 311–318.

[28] van Saarloos, P. P., and Constable, I. Improved method for calculation of corneal topography for any photokeratoscope geometry. *Optom Vis Sci 68* (1991), 960–6.

[29] Wang, J., Rice, D. A., and Klyce, S. D. A new reconstruction algorithm for improvement of corneal topographical analysis. *Refract. Corneal Surg 5* (1989), 379–387.

[30] Warnicki, J. W., Rehkopf, P. G., and Curtin, S. A. Corneal topography using computer analyzed rasterographic images. *Am. J. Opt 27* (1988), 1125–1140.

[31] Wilson, S. E., and Klyce, S. D. Advances in the analysis of corneal topography. *Surv. Ophthalmol. 35* (1991), 269–277.

Figure 6: Synthetic ellipsoid image.

Figure 7: Synthetic "bump on sphere" image.

Figure 8: Visualization in 3D of surface recovered from real data.

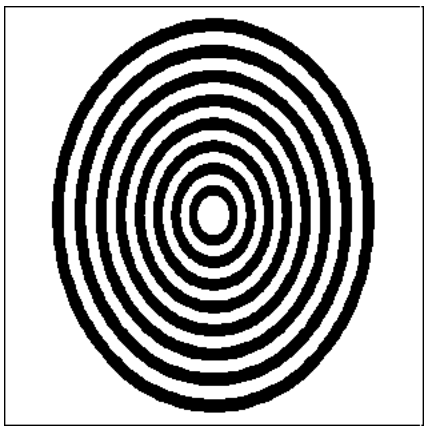


Figure 6: Synthetic ellipsoid image.

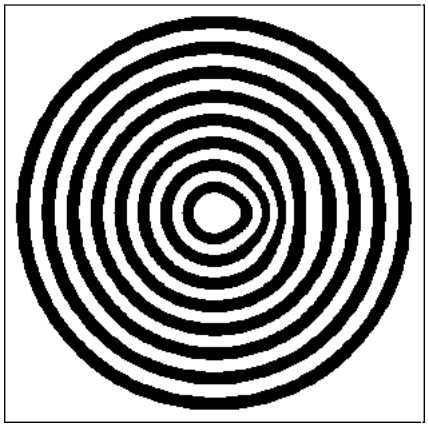


Figure 7: Synthetic "bump on sphere" image.

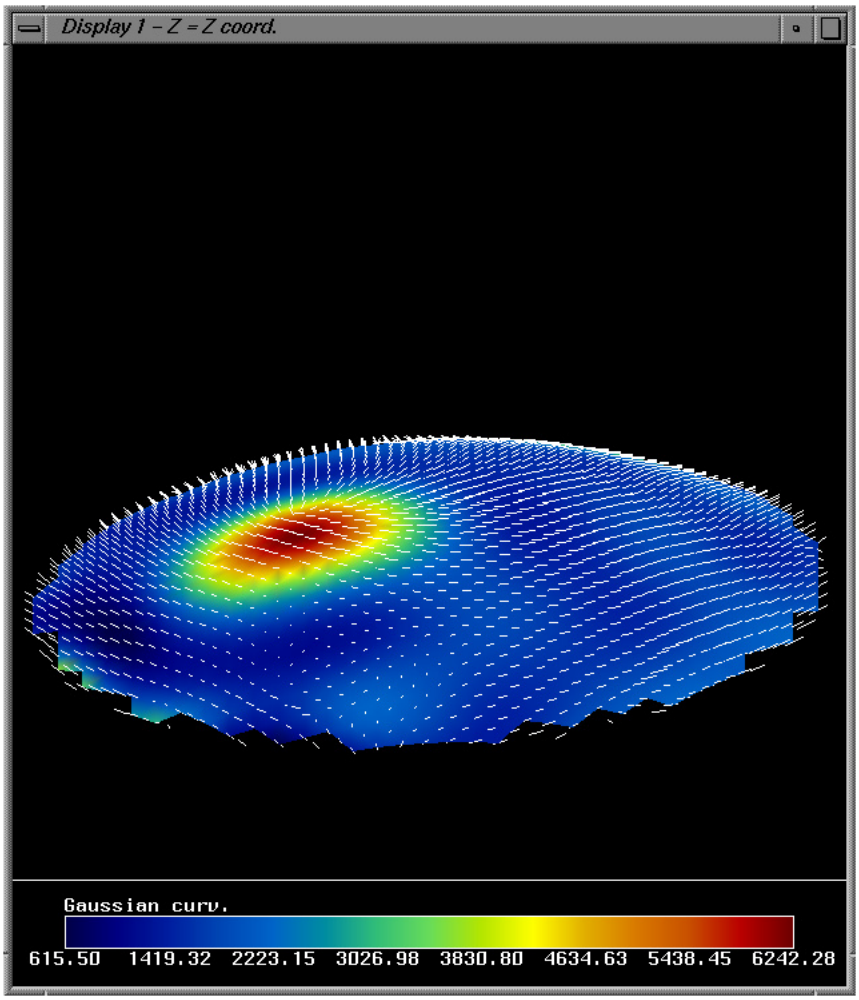
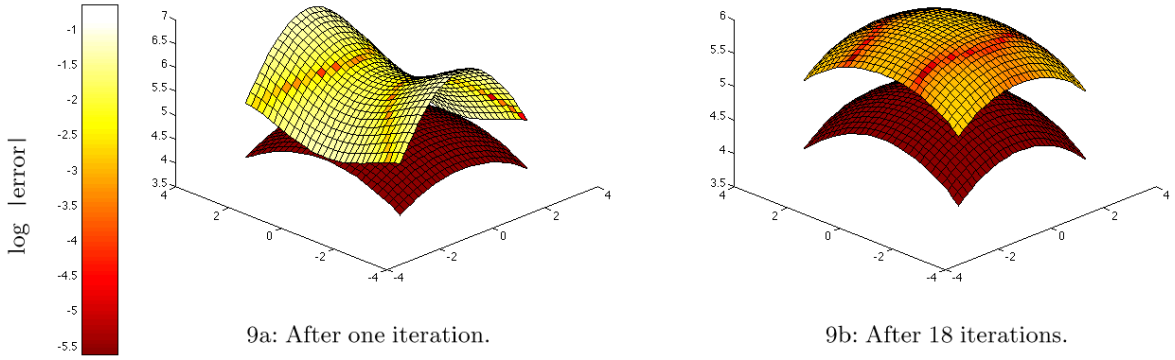


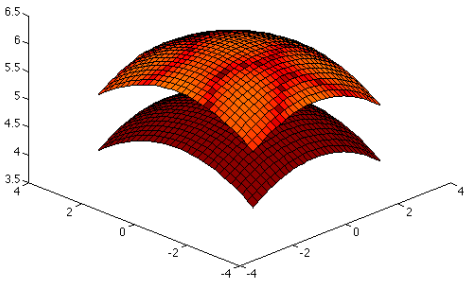
Figure 8: Visualization in 3D of surface reconstructed from real data.

Figure 9: Steps in recovering the shape of an ellipsoid.

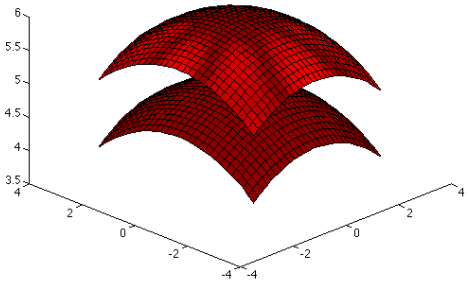


9a: After one iteration.

9b: After 18 iterations.

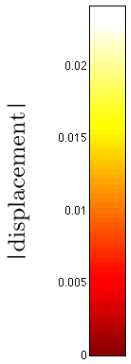


9c: After 36 iterations.

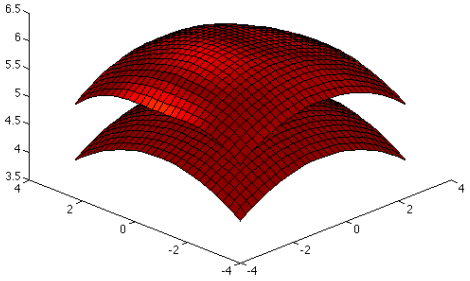


9d: After 64 iterations (final).

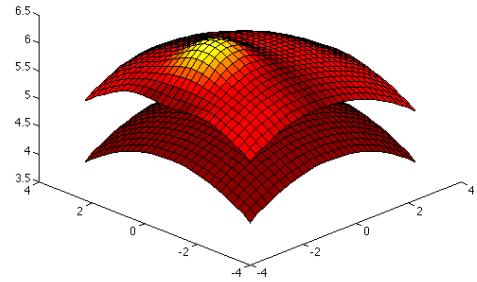
Figure 10: Steps in recovering the shape of a ‘‘bump on sphere’’.



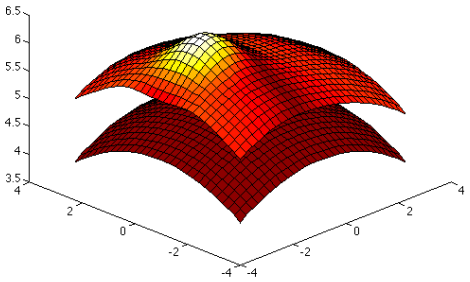
10a: After one iteration.



10b: After 10 iterations.



10c: After 25 iterations.



10d: After 53 iterations (final).