M.I.T Media Laboratory Perceptual Computing Section Technical Report No. 275
Appeared in *Proc. of the IEEE Workshop on Biomedical Image Analysis*, **Seattle, June 1994 (pp. 70–79).**

On Modal Modeling for Medical Images: Underconstrained Shape Description and Data Compression

Stan Sclaroff and Alex P. Pentland

Perceptual Computing Section, The MIT Media Laboratory
20 Ames St., Cambridge MA 02139

Abstract

We have previously described modal analysis, an efficient, physically-based solution for recovering, tracking, and recognizing solid models from 2-D and 3-D sensor data. The underlying representation consists of two levels: modal deformations, which describe the overall shape of a solid, and displacement maps, which employ a multiscale wavelet representation to provide local and fine surface detail. This paper addresses the problem of recovering modal models in the underconstrained case of fitting a 3-D model to contours found in medical slice and X-ray data. We will describe an extension which can be used to incorporate measurement uncertainty while estimating the modal deformation parameters. Finally, we give details about how to compress dense 3-D point data from surfaces, by use of displacement maps and wavelets.

1 Introduction

Contours present an interesting problem in fitting deformable solids to medical data: when a 3-D model is recovered from contours extracted from 2-D slice data, the model fitting algorithm must make an intelligent guess at what's going on in the third dimension — things are uncertain in the dimension perpendicular to a slice. This classic machine vision problem also arises when fitting partial or occluded data: how can we fill in information where we have no measurements, or noisy measurements?

To solve these problems, we should carry measurement uncertainty along in our solid model recovery framework, recovering not only model parameters, but also their distributions [7; 27]. Such a framework can be used to set limits on our model parameters based on their expected values and variances. In addition, recovering parameter distributions can provide a framework for dynamically adaptive acquisition of future data, and for sensor fusion.

Another commonly used method for underconstrained shape recovery is to incorporate *a priori* physical knowledge about the material properties of the object being modeled [18; 24]. The notion of employing physically-based solid shape modeling has recently become popular in the shape-modeling literature [3; 6; 11; 13; 14; 26]. However because they are mesh-based, these rubber-sheet models often have the drawback that it is not easy to compare two objects, or to compute geometric properties like surface area, volume, or minimum distance. For such representation, the only general method for determining if two surfaces are equivalent is to generate a number of sample points at corresponding positions on the two surfaces, and observe the distances between the two sets of sample points. Not only is this a clumsy and costly way to determine if two surfaces are equivalent, but when the two surfaces have very different parameterizations it can also be quite difficult to generate sample points at "corresponding locations" on the two surfaces.

1.1 Modal models

Pentland, Essa, and Sclaroff addressed this problem of obtaining a canonical description by adopting an approach based on *modal analysis* [18; 8]. This approach to geometric modeling is based on parameterized implicit functions, for instance, polynomial deformations of superquadric ellipsoid functions. A physical model is developed by discretizing the geometric model via the finite element method, and then making use of Galerkin polynomial approximations for shape interpolation. The geometric and physical models are then unified by using the finite element eigenmode shape functions as the underlying geometry's polynomial deformations. These deformations are called *free vibration modes* and their shapes are represented by a basis transform matrix $\mathbf{\Phi}$. Readers are directed to reference [18] for more complete explanation of this method.

By describing the finite element equations in terms of the object's free vibration modes, the system of equations is decoupled into a superposition of independent, frequency-ordered deformations. The decoupled modal dynamic governing equations can be written in matrix form (shown here with damping neglected):

$$\ddot{\tilde{\mathbf{U}}} + \tilde{\mathbf{K}}_6\tilde{\mathbf{U}} = \tilde{\mathbf{R}} \tag{1}$$

where $\tilde{\mathbf{K}}_6$ is the diagonal modal stiffness matrix as described in [18], and $\tilde{\mathbf{R}}(t) = \mathbf{\Phi}^T\mathbf{R}(t)$ is the load vector

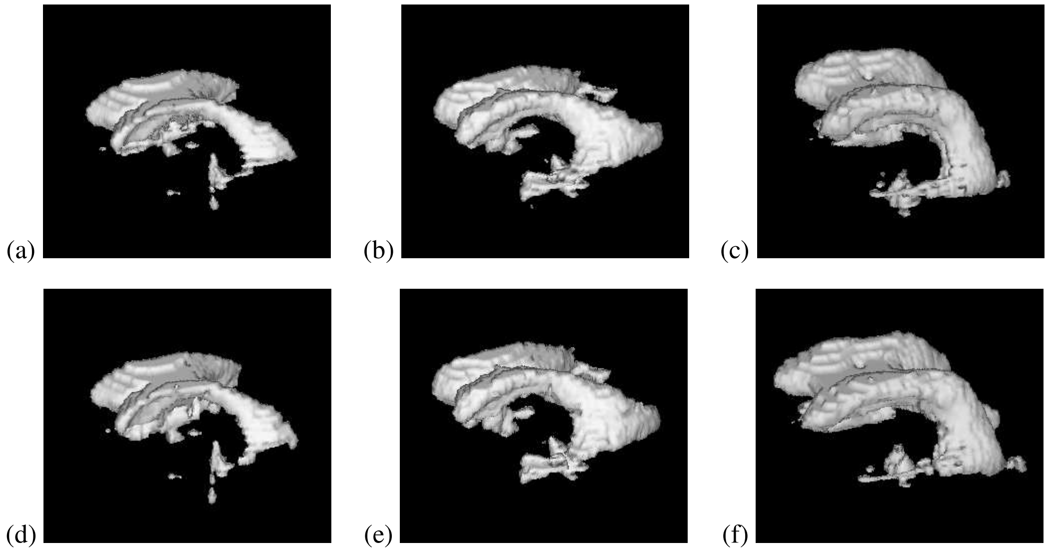

Figure 1: Using modal models to account for normal variation in head shape. The top row shows reconstructions of the ventricular system of the human brain created from MR images: (a) normal, healthy adult, (b) Alzheimer's disease causes an enlargement of ventricles, and (c) normal-pressure hydrocephalus causes an even greater enlargement of the ventricles, along with an overall "puffy" appearance. The bottom row (e,f,g) shows each of the ventricles normalized for cranium shape by warping each patient's cranium to a reference cranium using a modal model as described in [12].

which contains the virtual spring forces transformed into the modal basis.

Transformation into modal space produces a Fourier-like shape description that is maximally efficient at describing the shape and its physical properties. As with the Fourier description, we can smoothly trade accuracy against computational cost by discarding some of the high-frequency modes. Consequently, this transformation can greatly reduce data storage requirements, and the computational cost of physical simulations.

Finally, by combining a modal description with an implicit function representation for geometry (as detailed below), we can greatly reduce the computational cost associated with finding surface intersection, volume, area, *etc.* By discarding the higher-order modes, we get a parametric deformable model representation. The parameters are physically meaningful, and correspond to how much bending, tapering, pinching, etc., is needed to represent an object. As has been demonstrated in [12; 14; 18; 17] the modal representation can be quite effective for model recovery, recognition, and nonrigid motion tracking. An example of application of modal modeling to medical data is provided in the next section.

1.2 A medical example

Various neurological disorders affect the gross anatomical shape of different brain structures, as illustrated in Figure 1. Because of both genetic and environmental factors, however, biological structures have a large normal range of variation. The onset of a disease introduces further changes in morphology, with each particular disease causing its own type of shape changes and range of variation in these changes. Therefore to properly study the pathological deformations, we must first account for the large normal variation in biological shape.

For structures inside the cranium, a significant part of this normal shape variability is due to the different head shapes seen across individuals. These macroscopic shape effects interfere with the analysis of local shape deformations caused by disease.

We can use the modal modeling technique described above to separate out normal and disease-process deformations, by creating a physical model that describes the macroscopic effects caused by different head shapes. This allows us to accurately warp the cranial contents so that we obtain inter-patient shape differences that are largely independent of head shape. A related paper appearing in CVPR '94 describes using this approach to obtain more accurate classification of Alzheimer's, normal-pressure hydrocephalus, and normal patients [12].

1.3 Displacement maps

As mentioned in Section 1.1, discarding high frequency modes has many benefits; and as was demonstrated in the previous example, the modal representation can be profitably employed in volumetric data. There is, however, one drawback: by truncating out high-frequency shape information, we have effectively discarded local details. This will be a problem when fitting models to dense 3-D data like MR data, because we will not be able to capture local shape details in a meaningful way. To get around this problem, Sclaroff [19], proposed a model recovery framework which incorporates wavelet-based *displacement maps* to add fine detail to a solid model described by deformable analytic implicit functions. Such maps store surface displacements as a coded relief map, where brighter shades signify positive offsets, and darker shades signify negative offsets. A similar family of global-to-local, shape continuum models was also described by Terzopoulos and Metaxas [23] and has since been described by Vemuri *et al.* [25].

Displacement maps are similar to the offsets used for B-splines in that they allow for local and fine-detail control over the shape of the surface. In the context of solid models, displacement maps function by offsetting the analytic surface of the solid along its surface normals. The displacement maps themselves are computed, interpolated and efficiently compressed for storage by using a *wavelet representation* (quadrature mirror filters) [21; 19]. Such a multiscale representation affords not only a method for computing a continuum of shape models, it also provides a framework for regularizing surface scattered point data when actually building the displacement map [16].

This detail map methodology is especially appealing because the map can be stored as an intensity image. The most useful side-effects of this are that displacement maps can be compressed via standard image compression techniques [21], and that the representation is especially well-suited for physical simulation [20]. Since the resulting part representation is still an implicit function (displacement maps merely offset the underlying deformed function) intersection and inside/outside measurements are easy to compute.

1.4 Overview

This paper will describe two parts of the modal model recovery framework which would be particularly useful in the area of fitting volumetric models to medical data. We will concern ourselves first with fitting modal models to three dimensional slice data, and second with computing, storing, and interpolating displacement maps using a wavelet representation.

In Section 2, we describe two different methods for dealing with the underconstrained problems associated with fitting modal models to planar slice and contour data. The first method employs a strain minimization term to enforce certain *a priori* geometric and physical constraints. The second method utilizes a recursive least squares framework to allow for addition of new slices as they are collected, and also enforces a minimum strain solution. The methods are demonstrated on contours taken from artificial data and from real X-ray data of a human heart.

In Section 3, we first describe the mathematics behind our representation; we describe how modal deformations and displacement maps yield a multiresolution implicit function. We then describe how to recover displacement maps from scattered point data. The method is demonstrated on range data for a human head. Finally, we describe a wavelet-based displacement map representation which allows for interpolation between scales, and for very efficient storage of local details at different scales.

1.5 Software implementation

Implementations of these algorithms in 'C' code are available from whitechapel.media.mit.edu, in the files /u/ftp/misc/modal.tar.Z and /u/ftp/misc/epic.

2 Estimating 3-D shape from contours

To recover models from images, we need a load vector $\tilde{\mathbf{R}}$ to plug into Equation 1 which will pull our deformable models towards low-level image features. The task will then be to calculate the best-fitting (minimum energy) part description from a relatively small number of noisy edge, corner, contours, and other feature positions [18; 24]. This is typically done by attaching virtual springs between sensor measurements and our deformable object, implicitly specifying correspondences between FEM nodes and image features. In most situations this correspondence is not given, and so must be determined in parallel with the equilibrium solution.

Before correspondences can be made, however, we need to calculate an initial guess of the model's size and orientation (assuming the image has already been segmented into "blobs"). This is done via the *moments method* [18; 24]. In the moments method, the first step is to define an ellipsoidal coordinate system by examination of the data's moments of inertia.

2.1 Interpolating displacements

Once an initial model estimate has been obtained, we then turn to the problem of attaching virtual springs between the measurements and the model's surface — often these attachment points are not at the finite element nodes. When a spring is attached to the model's surface at a point $\overline{\mathbf{x}}_{(i)}$ which is not a finite element node, then we need to distribute its loads $\overline{\mathbf{r}}_{(i)}$ and resulting displacements $\overline{\mathbf{u}}_{(i)}$ in a manner which is consistent with the finite element interpolation functions $\mathbf{H}_{(i)}$. In general, we build an interpolation matrix which relates the displacements at the m attachment points to displacements at the n nodes:

$$\overline{\mathbf{U}} = \overline{\mathbf{H}}\mathbf{U}, \tag{2}$$

where $\mathbf{U}$ is the vector of nodal displacements, $\overline{\mathbf{U}}$ is the vector of spring attachment point displacements, and $\overline{\mathbf{H}}$ is the interpolation matrix:

$$\overline{\mathbf{H}} = \begin{bmatrix} \mathbf{H}_{(1)} \\ \vdots \\ \mathbf{H}_{(m)} \end{bmatrix}. \tag{3}$$

Each $\mathbf{H}_{(i)}$ is a $3 \times 3n$ interpolation submatrix which relates the displacement at a particular attachment point $\overline{\mathbf{x}}_{(i)}$ to the displacements at the node points as described in [18].

Recall that nodal and modal displacements are related by the modal transform:

$$\mathbf{U} = \mathbf{\Phi}\tilde{\mathbf{U}}. \tag{4}$$

By using Equation 2, we can then interpolate modal displacements at the spring attachment points:

$$\overline{\mathbf{U}} = \overline{\mathbf{\Phi}}\tilde{\mathbf{U}}. \tag{5}$$

where the interpolated modal transform matrix $\overline{\mathbf{\Phi}} = \overline{\mathbf{H}}\mathbf{\Phi}$ describes how the modal displacements at the nodes relate to displacements at the spring attachment points.

2.2 The weighting matrix

To recover a shape, we will need to calculate a modal parameter vector which describes how to deform our model to fit the measurement data; intuitively, the measurements represent anchor points for springs of various strengths which are pulling our deformable model towards the data. In this framework every object starts out as an ellipsoid, and then is smoothly deformed into its recovered shape. An object's shape description is simply the vector of *modal amplitudes* $\tilde{\mathbf{U}}$ which specifies the deformations needed to obtain the final shape.

Assume that we have a vector of three-dimensional sensor measurements $\overline{\mathbf{X}}^m$ which originate from the surface of a single object in the image. Then, given a vector of corresponding attachment points on the model's surface $\overline{\mathbf{X}}$, we can calculate the displacements needed to move the surface points to fit the data points. In the general case, we can solve for the final shape by computing the dynamic solution to Equation 1 via direct integration techniques [2], or if our initial guess is reasonably close to the shape to be recovered, we can solve for the static solution ($\ddot{\tilde{\mathbf{U}}} = 0$) in closed-form:

$$\tilde{\mathbf{U}} = \tilde{\mathbf{K}}_6^{-1}\tilde{\mathbf{R}}. \tag{6}$$

The modal load vector $\tilde{\mathbf{R}}$ incorporates the spring forces which push and pull the model towards the measurements:

$$\tilde{\mathbf{R}} = k\overline{\mathbf{\Phi}}^T\mathbf{W}^2(\overline{\mathbf{X}}^m - \overline{\mathbf{X}}) \tag{7}$$

where k is an overall spring constant, and $\mathbf{W}$ is a spring weighting matrix whose diagonal elements are inversely proportional to the measurement standard deviations:

$$\mathbf{W} = diag\left[\frac{1}{\sigma_{x_1}}, \frac{1}{\sigma_{y_1}}, \frac{1}{\sigma_{z_1}}, \ldots \frac{1}{\sigma_{x_m}}, \frac{1}{\sigma_{y_m}}, \frac{1}{\sigma_{z_m}}\right]. \tag{8}$$

This assumes that the measurements are decoupled; to make the weighting scheme more general, we can compute this matrix directly from the measurement covariance matrix $\mathcal{N}$:

$$\mathbf{W} = \mathcal{N}^{-\frac{1}{2}}. \tag{9}$$

This type of weighting matrix is particularly useful for fitting to slice data or for merging measurements from different sources, since we can explicitly incorporate measurement uncertainty.

2.3 Least-squares fitting

If we use weighted springs, it can sometimes be difficult to choose the appropriate spring constant k to get the surface to displace enough to tightly match the measurements. To get around this, some authors have suggested the use of "ratchet springs" which are dynamically tightened until the surface matches the data [10], or slowly inflate finite element "balloons" [5; 6; 13]. In these methods the material or spring properties have to be iteratively tweaked until a tight fit is achieved.

Within the modal framework, it is possible to solve for a similar tight fit in closed form. When the number of independent sensor measurements equals the number of FEM nodes, then the modal deformation parameters $\tilde{\mathbf{U}}$ are simply solved for by inverting $\overline{\mathbf{\Phi}}$ in Equation 5.

In general, however, the number of measurements will be greater than the number of deformation parameters to be estimated, and so we solve via weighted least squares:

$$\tilde{\mathbf{U}} = \left(\mathbf{W}\overline{\mathbf{\Phi}}\right)^+\overline{\mathbf{U}}, \tag{10}$$

where $\left(\mathbf{W}\overline{\mathbf{\Phi}}\right)^+$ is the pseudo-inverse, and $\overline{\mathbf{U}} = \overline{\mathbf{X}}^m - \overline{\mathbf{X}}$ describes the desired surface point displacements.

When there are more measurements than nodes, the weighting matrix will cause the model to more closely fit measurements which are more certain. Intuitively, this is similar to spring-based fitting described in the previous section except that we are explicitly trying to fit the surface to known displacements, given certain weighting criteria.

2.4 Strain minimizing least squares

Unfortunately, a simple weighted least squares scheme will not always suffice: it is often the case that the image

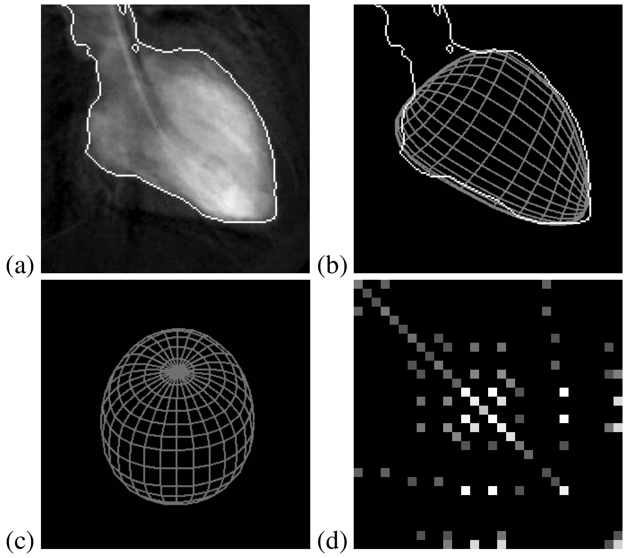

Figure 2: An example showing the use of the strain minimizing least squares to recover a 3-D deformable solid model from a 2-D image contour. The original image and contour are shown in (a). The model is recovered from the contour as shown in (b). An orthogonal side view is shown in (c). The fitting employed 30 deformation modes. The magnitude image of the estimated 30×30 estimation covariance matrix for the recovered mode amplitudes is shown in (d), where bright entries signify large covariances, and black values small. Bright off-diagonal elements in the matrix point to modes which have a strong z component.

measurements will cause the least squares solution to be underconstrained. This is particularly true when recovering geometry from image contours or from data collected from only one side of an object. In such cases we can resort to solving for the dynamic solution, performing modal analysis with fewer modes [18], or we can solve for the modes via least squares, if we update our least squares to include a strain energy minimization term. In the modal framework, the strain energy is computed in terms of the modal displacements and the stiffness associated with each of the modes:

$$E_{strain} = \tilde{\mathbf{U}}^T \tilde{\mathbf{K}}_6 \tilde{\mathbf{U}}. \tag{11}$$

For our least squares solution, we want to minimize error which includes this strain energy term:

$$E = \underbrace{\left[\overline{\mathbf{U}} - \overline{\mathbf{\Phi}}\tilde{\mathbf{U}}\right]^T \left[\overline{\mathbf{U}} - \overline{\mathbf{\Phi}}\tilde{\mathbf{U}}\right]}_{\text{squared fitting error}} + \underbrace{\lambda \tilde{\mathbf{U}}^T \tilde{\mathbf{K}}_6 \tilde{\mathbf{U}}}_{\text{strain energy}}. \tag{12}$$

Thus we will exploit our underlying physical model to enforce certain geometric constraints. The strain energy measure will allow us to incorporate some prior knowledge about how stretchy the shape is, how much it resists compression, *etc.* Using this extra knowledge, we can infer what "reasonable" displacements would be at points on the surface which are occluded, along the axis perpendicular to a data slice. This strain operator has parallels in the smoothness functional employed in regularization [22].

Differentiating with respect to the modal parameter vector, we get:

$$\frac{\partial E}{\partial \tilde{\mathbf{U}}} = 2\left[\overline{\mathbf{\Phi}}^T \overline{\mathbf{\Phi}}\tilde{\mathbf{U}} - \overline{\mathbf{\Phi}}^T \overline{\mathbf{U}} + \lambda \tilde{\mathbf{K}}_6 \tilde{\mathbf{U}}\right]. \tag{13}$$

Moving terms, we obtain our least squares solution:

$$\tilde{\mathbf{U}} = \left[\overline{\mathbf{\Phi}}^T \overline{\mathbf{\Phi}} + \lambda \tilde{\mathbf{K}}_6\right]^{-1} \overline{\mathbf{\Phi}}^T \overline{\mathbf{U}}. \tag{14}$$

As before, we would like to incorporate measurement uncertainties in the form of a spring weighting matrix. We can perform weighted least squares by including the inverse of the measurement covariance matrix $\mathcal{N}^{-1}$:

$$\tilde{\mathbf{U}} = \left[\overline{\mathbf{\Phi}}^T \mathcal{N}^{-1} \overline{\mathbf{\Phi}} + \lambda \tilde{\mathbf{K}}_6\right]^{-1} \overline{\mathbf{\Phi}}^T \mathcal{N}^{-1} \overline{\mathbf{U}}. \tag{15}$$

Finally, the estimation covariance matrix $\mathbf{P}$ for the recovered modal amplitudes $\tilde{\mathbf{U}}$, is:

$$\mathbf{P} = \left[\overline{\mathbf{\Phi}}^T \mathcal{N}^{-1} \overline{\mathbf{\Phi}} + \lambda \tilde{\mathbf{K}}_6\right]^{-1}. \tag{16}$$

2.4.1 Example: fitting to X-ray contour data

Figure 2 demonstrates how in the underconstrained case a model was recovered from image contour data. Figure 2(a) shows a frame from a sequence of X-ray images, with the edge contours overlayed. The contours were extracted by first thresholding the X-ray imagery, and then finding the Laplacian zero-crossings. From these contours 3-D shape was estimated by use of Equation 15. The fitting employed 30 deformation modes. The spring weighting matrix was diagonal, and was adjusted to reflect the fact that the spring stiffness in the direction perpendicular to the line of sight (z springs) is weaker. The resulting estimate is shown in Figure 2(b) as a 3-D wireframe overlayed on the original X-ray data. Figure 2(c) shows the recovered model from the side.

It can be seen that a reasonable estimate of 3-D shape was obtained. Because only bounding contour information was available, the shape estimated along the z axis (shown in Figure 2(d)) is determined by finding the minimum stress state that still fits the bounding contour. Note that use of a minimum stress criterion for the solution means that symmetric and mirror symmetric shapes are preferred. The z-axis shape cannot, therefore, be regarded as accurate but only as plausible and consistent. This uncertainty about parameters which have some z component can be effectively estimated by using Equation 16. Figure 2(d) depicts a grey-scale image of the 30×30 estimation covariance matrix for the recovered mode amplitudes. Bright off-diagonal elements in the matrix point to modes which have a strong z component.

2.5 Recursive formulation

Suppose that we are given measurement sets for a static object at different times, or that we are given a succession of slices through an object. It would be useful to be able to iteratively fit a model, recursively "fusing" new measurements into our estimated model parameters — without restarting our fitting calculations from square one. In such cases, we can employ a recursive weighted least squares model. The resulting algorithm can also recover the covariances for the estimated modal parameters. This allows for dynamically adaptive acquisition of future data in such a way as to optimally increase certainty in the recovered parameters. The feasability of an "active" approach to MRI acquisition is discussed in [15].

In our recursive formulation the least squares estimates of the modal parameters $\tilde{\mathbf{U}}$ are found by solving:

$$\tilde{\mathbf{U}}_i = \tilde{\mathbf{U}}_{i-1} + \mathbf{K}_i \left[\overline{\mathbf{U}}_i - \overline{\mathbf{\Phi}}_i \tilde{\mathbf{U}}_{i-1} \right] - \lambda \mathbf{P}_i \tilde{\mathbf{K}}_6 \tilde{\mathbf{U}}_{i-1} \quad (17)$$

where $\mathbf{P}_i$ is the estimated parameter covariance matrix at the i^{th} iteration

$$\mathbf{P}_i^{-1} = \mathbf{P}_{i-1}^{-1} + \overline{\mathbf{\Phi}}_i^T \mathcal{N}_i^{-1} \overline{\mathbf{\Phi}}_i + \lambda \tilde{\mathbf{K}}_6, \quad (18)$$

and $\mathbf{K}_i$ is the gain matrix

$$\mathbf{K}_i = \mathbf{P}_i \overline{\mathbf{\Phi}}_i^T \mathcal{N}_i^{-1}. \quad (19)$$

At each iteration, the inverse $\mathbf{P}_i^{-1}$ can be computed inexpensively via a recursive QR algorithm, or by recursive updates to an LDL^T decomposition [9].

2.5.1 Example: recursively fitting slice data

Figures 3 and 4 show an example of using the recursive fitting algorithm to recover a modal model from contours which were extracted from slices through range data. Sixteen planar slices were used. At each iteration, a new contour was "acquired" and recursively added to the strain minimizing, least squares fit. The diagram in Figure 3(a) shows the sixteen planar slices, and their ordering and positions. The slices are cut through the head from above, with the slice plane rotating around the head's center axis. Figure 3(b) shows the contour extracted from the first slice. The data for this example was acquired using a 360-degree Cyberware range scanner.

In our experience with contour data of this sort, the low-order deformation modes have a tendency to settle down first, and then the higher frequency modes begin to converge. This is better seen in the graph in Figure 3(c). The graph shows a plot of the modal amplitudes for the first ten, lowest-order, nonrigid modes as they are iteratively estimated for the human head data. Note that in this example, the recursive formulation begins to converge on these values around the tenth slice or so. Convergence on modal

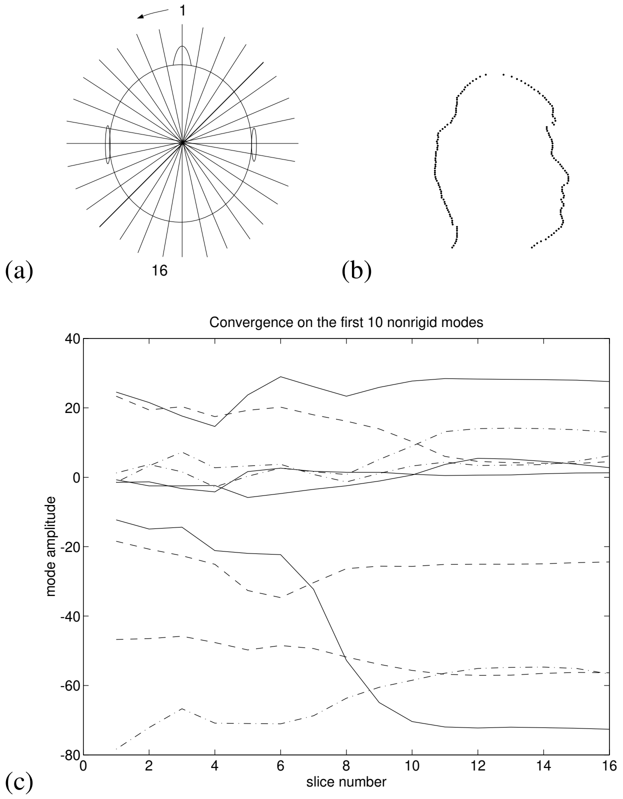

Figure 3: Recursively fitting a model to head silhouette data. The diagram (a) shows the sixteen planar slice ordering and positions used in the fitting iteration (top view of head). An example slice is shown in (b). The graph (c) shows a plot of the modal amplitudes for the first ten, lowest-order, nonrigid modes as they are iteratively estimated.

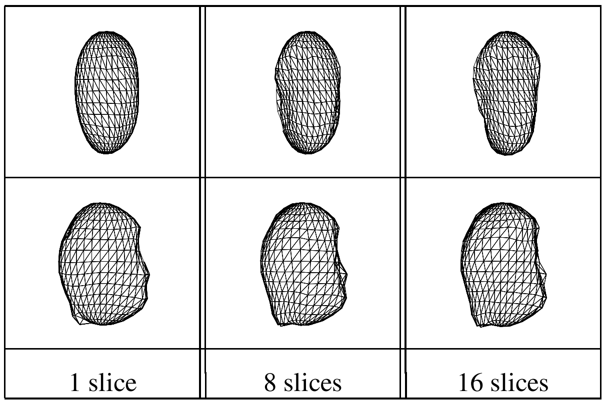

Figure 4: The recovered geometry from recursive fitting of a human head from slice data shown in Figure 3. An 81 parameter modal model was recovered, using the strain minimizing algorithm described in the text. The top row of the figure shows the front view of the model at each step, and the bottom row shows the orthogonal side view.

values can vary depending on the way slices were ordered and/or configured.

Figure 4 shows the recursively recovered deformable model (shown in wireframe, hidden surfaces removed). The fitting progressed for sixteen iterations, starting with the profile slice of the head. Eighty-one modal vibration modes were used to describe the model (27 node finite element). The top row of the figure shows the front view of the model after the addition of the first, eighth, and sixteenth data slices, and the bottom row shows the corresponding orthogonal side views. As can be seen, the model converges rather quickly on a plausible fit, and fills out more detail in the front view as the iterations progress.

3 From global to local deformations

We must keep in mind that there are two contradictory problems which need to be solved in recovering a shape representation. If a computer is to recognize and reason about objects, then it needs a high-level representation which somehow reduces the confounding volume of surface information down to a manageable abstraction. If, however, a computer is to interact with the world, predict behavior, simulate reality, or encode images, then it needs a representation which is general and expressive enough to accurately represent localized surface details and physical behavior. A multiscale method is needed, one which allows for choosing the level of detail needed for comparison, simulation, or compression. A standard image compression technique for this is pyramid image encoding. This section will describe a multiresolution frequency coding method for representing three-dimensional shape (originally described in [20]). The resulting hybrid representation will be a physically-based, parametric model and supplemented with a *displacement map* which will be used to encode local surface detail.

Our implicit function representation defines a surface as a level set of a function f, most commonly the set of points for which $f(\mathbf{x}) = 0$. For instance, the inside-outside function we use for superquadric ellipsoids, before rotation, translation or deformation, is:

$$f(\mathbf{x}) = \left[(x^{2/\epsilon_2} + y^{2/\epsilon_2})^{\epsilon_2/\epsilon_1} + z^{2/\epsilon_1}\right]^{\epsilon_1/2} - 1, \qquad (20)$$

where ϵ_1 and ϵ_2 are the superquadric squareness parameters. In practice we have found this function better behaved than the standard superquadric inside-outside function, as it is more similar to the normal $\mathbf{L}_2$ distance metric. A solid defined in this way can be easily positioned and oriented, by transforming the implicit function:

$$\overline{\mathbf{x}} = \mathbf{M}\mathbf{x} + \mathbf{b} \qquad (21)$$

where $\mathbf{M}$ is a rotation matrix, and $\mathbf{b}$ is a translation vector. Similarly, the implicit function's positioned and oriented inside-outside function becomes:

$$f(\mathbf{x}) = f(\mathbf{M}^{-1}(\overline{\mathbf{x}} - \mathbf{b})). \qquad (22)$$

To detect a collision between a point $\mathbf{x} = (x, y, z)$ and the volume bounded by this surface, one simply substitutes the coordinates of $\mathbf{x}$ into the function f.

As in Barr [1], this basic set of functions can be generalized further by defining an appropriate set of global deformations $\mathcal{D}$ with parameters $\mathbf{u}$. For particular values of $\mathbf{u}$ the new deformed surface is defined using a deformation matrix $\mathcal{D}_\mathbf{u}$:

$$\overline{\mathbf{x}} = \mathbf{M}\mathcal{D}_\mathbf{u}\mathbf{x} + \mathbf{b} \qquad (23)$$

where $\overline{\mathbf{x}}$ is the position vector after rotation, deformation, and translation. Similarly, the inside-outside function becomes

$$f(\mathbf{x}) = f(\mathcal{D}_\mathbf{u}^{-1}\mathbf{M}^{-1}(\overline{\mathbf{x}} - \mathbf{b})). \qquad (24)$$

This inside-outside function is valid as long as the inverse deformation $\mathcal{D}_\mathbf{u}^{-1}$ exists. Thus by selecting a set of deformations that can be easily inverted, we can greatly expand the class of shapes that can be described using an implicit function representation.

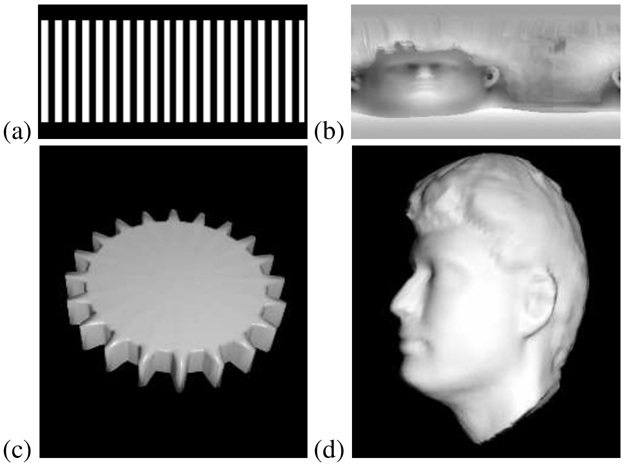

Figure 5: Superquadrics with displacement maps. Shown: (a) displacement map for a gear (b) displacement map for a head (c) gear and (b) head.

3.1 Displacement maps

The class of implicit functions can be generalized still further by defining the surface as the set of points for which $f(\mathbf{x}) = d$ for some displacement function $d(\eta, \omega)$, where η, ω are the point's coordinates in the surface's two-dimensional parametric space. We define $\tilde{\mathbf{x}}(\eta, \omega)$ to be the displaced surface point before rotation, deformation or translation:

$$\tilde{\mathbf{x}} = \mathbf{x} + d\mathbf{n} \qquad (25)$$

This will have the effect of displacing the surface along the surface normal in the original undeformed space before applying the deformation $\mathcal{D}_{\mathbf{u}}$.

Displacement maps are stored as two-dimensional floating point image arrays. Figure 5 shows some examples of superquadrics with displacement maps. In these examples, the displacement map function $d(\eta, \omega)$ is defined by bilinear interpolation between point samples. If a smoother interpolation is desired, standard spline or pyramid-based multiresolution methods [4] can be employed.

Given a point $\mathbf{x}(\eta, \omega)$ and normal $\mathbf{n}(\eta, \omega)$ on the undeformed implicit surface, deformation transform $\mathcal{D}_{\mathbf{u}}$, and scalar displacement map function $d(\eta, \omega)$, we define the position vector including deformations and displacement map:

$$\overline{\mathbf{x}} = \mathbf{M}\mathcal{D}_{\mathbf{u}}\tilde{\mathbf{x}} + \mathbf{b} = \mathbf{M}\mathcal{D}_{\mathbf{u}}(\mathbf{x} + d\mathbf{n}) + \mathbf{b} \qquad (26)$$

We also need to define $\bar{\mathbf{n}}$ to be the normal vector including deformations and displacement. To find $\bar{\mathbf{n}}$, it is first necessary to find $\tilde{\mathbf{n}}$, the surface normal for the displaced surface before deformation, and then apply deformations to get $\bar{\mathbf{n}}$. The normal $\tilde{\mathbf{n}}(\eta, \omega)$ is derived by taking the cross product of the partial derivatives for the undeformed displaced surface function, $\tilde{\mathbf{x}}(\eta, \omega)$. In our current implementation, the partials are calculated by finite differences.

The inside-outside function associated with Equation 26 is then:

$$f(\mathbf{x}) = f(\mathcal{D}_{\mathbf{u}}^{-1}\mathbf{M}^{-1}(\overline{\mathbf{x}} - \mathbf{b}) - d\mathbf{n}). \qquad (27)$$

It is difficult to evaluate a displacement mapped and deformed inside-outside function because we cannot know d or $\mathbf{n}$ in Equation 27 before we know $\mathbf{x}$, and *vice versa*. We therefore need a parametric projection function $P(\tilde{\mathbf{x}}) = (\eta, \omega)$, which can project an undeformed, displacement-mapped $\tilde{\mathbf{x}}$ point back onto the original, undisplaced implicit surface. These parameters can then be used to determine $\mathbf{n}$, $\mathbf{x}$, and d. The projection should be *normal projection* since the displacement map displaces surface points along the surface normal.

For example, the projection function $P(\tilde{\mathbf{x}}) = (\eta, \omega)$ for superquadric ellipsoids is computed as follows. We first find ω by observing:

$$\frac{\tilde{y}}{\tilde{x}} = \frac{\cos^{\epsilon_1}\eta \sin^{\epsilon_2}\omega}{\cos^{\epsilon_1}\eta \cos^{\epsilon_2}\omega} = \tan^{\epsilon_2}\omega \qquad (28)$$

where $(\tilde{x}, \tilde{y}, \tilde{z})^T = \tilde{\mathbf{x}}$ is the undeformed, displaced surface point. From Equation 28, we see that $\omega = \mathrm{atan}^{1/\epsilon_2}(\tilde{y}/\tilde{x})$. The remaining parameter, η, is determined by either $\eta = \mathrm{atan}^{1/\epsilon_1}((\tilde{z}\cos^{\epsilon_2}\omega)/\tilde{x})$ or $\eta = \mathrm{atan}^{1/\epsilon_1}((\tilde{z}\sin^{\epsilon_2}\omega)/\tilde{y})$ depending on whether $\tilde{x}$ or $\tilde{y}$ is larger.

3.2 Fitting a displacement map

If there are more degrees of freedom in the data points than in the deformation parameters, the deformed model will not generally pass through the data points exactly (*i.e.*, $f(\mathcal{D}_{\mathbf{u}}^{-1}\mathbf{M}^{-1}(\overline{\mathbf{x}} - \mathbf{b})) \neq 0, \overline{\mathbf{x}} \in \overline{\mathbf{X}}^m$). A more accurate approximation to the data points can be obtained by incorporating these residual differences into a displacement map. This is done by solving for the displacement map value $d(\eta, \omega)$ that yields $\overline{\mathbf{x}} = \mathbf{M}\mathcal{D}_{\mathbf{u}}(\mathbf{x} + d\mathbf{n}) + \mathbf{b}$ for some point $\mathbf{x}$ on the undeformed implicit surface. The final result is a generalized implicit function representation that normally provides an exact fit to the set of initial data points, and provides a smoothly interpolated surface between those points — except in certain degenerate cases, such as when Nyquist criteria are not satisfied.

To compute the displacement map each data point is subjected to the inverse deformation $\mathcal{D}_{\mathbf{u}}^{-1}$ to obtain $\tilde{\mathbf{x}}$, the point in the undeformed space. Next, we project $\tilde{\mathbf{x}}$ along the surface normal using $P(\tilde{\mathbf{x}})$ to obtain its two-dimensional parametric space coordinate (η, ω). Finally, we compute the undeformed point's normal distance to the undeformed implicit surface by substituting it's coordinates into the surface's inside-outside function:

$$d(\eta, \omega) = f(\tilde{\mathbf{x}}). \qquad (29)$$

In our system displacement maps are represented by a regularly spaced grid in the surface's parametric space. Thus as each point is projected and its displacement determined, the result can be spread to nearest grid points using Gaussian weighted averaging. This interpolation method works well only when data points are fairly dense. For the more general case of scattered data, we can employ a wavelet-based regularization method as described in [16]. This regularization method utilizes a quadrature mirror filter (QMF) basis.

3.3 Displacement map interpolation and storage

Since displacement maps are stored as large floating point image arrays (usually on the order of 100x100 or 256x256), there is some concern about the amount of space they occupy in memory or on disk. To get around this problem, we propose using a wavelet pyramid-based compression and interpolation scheme which significantly reduces the amount of space needed to store a displacement map [19]. Figure 6 shows an example of a head's displacement map which can be interpolated at different scales by using a QMF (Quadrature Mirror Filter) pyramid. The displacements were interpolated at three resolutions. At each level, the corresponding head and its rectangular displacement map are shown (whites signify peaks, blacks signify valleys). The smallest head, and its corresponding level in the continuum, represent the lowest frequency information

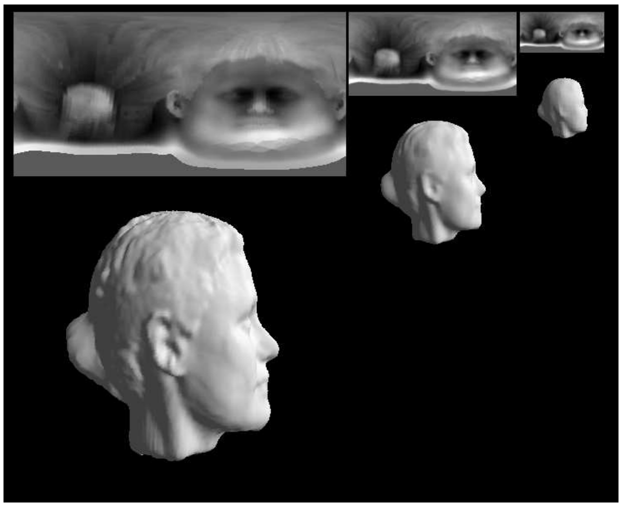

Figure 6: Displacement maps interpolated at various scales. Wavelet-based (QMF) interpolation allows for smooth transition of local shape at different scales.

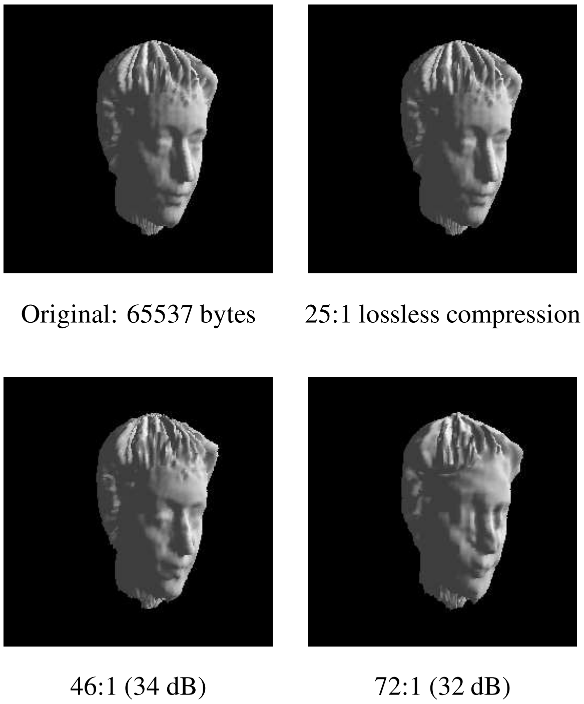

Figure 7: Using pyramid-based image coding to perform both lossless and lossy compression of a recovered head's displacement map. Signal to noise ratios shown were computed for the actual displacement map which when uncompressed is stored as a 2D eight-bit grayscale image.

about bumpiness; at each higher level, the map includes higher and higher frequency (resolution) information about local shape.

The QMF pyramid's hierarchy of scale is particularly useful for image interpolation between scales. Conveniently, QMF's have also been shown to be very efficient for image coding. Thus, to compress our displacement map images, we use a QMF-based compression package called EPIC, developed by Simoncelli and Adelson [21]. EPIC typically offers more effective lossless compression rates than a Laplacian pyramid or JPEG image coder.

Figure 7 shows an experiment in which we compressed a displacement map (256×256), therefore requiring 65537 bytes. The figure shows the original head, and then its compressed versions at various compression rates. The best we get is compressing the image to 2613 bytes; this represents a lossless compression rate of more than 25:1. To compress further, we employ a lossy coding scheme. The displacement map was compressed to 1422 bytes — a nearly 50:1 compression rate with an SNR around 34dB. Signal to noise ratios shown were computed for the actual displacement maps in 2D. Compressing even further, a 72:1 compression rate (32 dB) was achieved.

4 Conclusion

We have previously described modal analysis, an efficient, physically-based solution for recovering, tracking, and recognizing solid models from 2-D and 3-D sensor data. The underlying representation consists of two levels: modal deformations, which describe the overall shape of a solid, and displacement maps, which employ a multiscale wavelet representation to provide local and fine surface detail. This paper addresses the problem of recovering modal models in the underconstrained case of fitting a 3-D model to contours found in medical slice and X-ray data. We described an extension which can be used to incorporate measurement uncertainty while estimating the modal deformation parameters. The resulting formulation could be used for adaptive acquisition of data as suggested in [7; 27; 15].

Finally, we gave details about how to compress dense 3-D point data from surfaces, by use of displacement maps and wavelets. The inclusion of displacement maps provides a two-tiered method for describing an object: first the system will computes a low-order modal fit, and then it does a second pass in which more localized shape information is stored in a displacement map, using a wavelet representation to interpolate smoothly between local shape information at different scales. The resulting framework is useful for recovering volumetric models from both dense and sparse data, and provides a powerful framework for sensor fusion and for data compression.

References

[1] A. Barr. Global and Local Deformations of Solid Primitives. *Computer Graphics*, 18(3):21–30, 1984.

[2] K. Bathe. *Finite Element Procedures in Engineering Analysis*. Prentice-Hall, 1982.

[3] F. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE PAMI*, 11(6):567–585, June 1989.

[4] P. J. Burt and E. H. Adelson. A Multiresolution Spline With Application to Image Mosaics. *ACM Transactions on Graphics*, 2(4):217–236, 1983.

[5] Y. Chen and G. Medioni. Fitting a surface to 3-d points using an inflating balloon model. In *Proc. Second CAD-Based Vision Workshop*, pages 266–273, Feb 1994.

[6] L. Cohen. On active contour models and balloons. *CVGIP: Image Understanding*, To Appear 1992.

[7] H. Durrant-Whyte. Uncertain geometry in robotics. *IEEE Journal of Robotics and Automation*, 4(1), February 1988.

[8] I. Essa, S. Sclaroff, and A. Pentland. *Directions in Geometric Computing*, chapter Physically-based Modeling for Graphics and Vision. Information Geometers, U.K., 1992. R. Martin, Ed.

[9] G. Golub and C. van Loan. *Matrix Computations*. Johns Hopkins Press, 1989.

[10] A. Gupta and C.-C. Liang. 3-D model-data correspondence and nonrigid deformation. In *Proc. CVPR*, pages 756–757, 1993.

[11] W. Huang and D. Goldgof. Adaptive size physically-based models for nonrigid motion analysis. In *Proc. CVPR*, pages 833–835, 1992.

[12] J. Martin, A. Pentland, and R. Kikinis. Shape analysis of brain structures using physical and experimental modes. In *Proc. CVPR*, June 1994.

[13] T. McInerney and D. Terzopoulos. A finite element model for 3-D shape reconstruction and nonrigid motion tracking. In *Proc. ICCV*, May 1993.

[14] C. Nastar and N. Ayache. Fast segmentation, tracking, and analysis of deformable objects. In *Proc. ICCV*, pages 275–279, 1993.

[15] L. Panych and F. Jolesz. Dynamically adaptive MRI by wavelet transform encoding. Submitted to *Magnetice Resonance in Medicine*, March 1994.

[16] A. Pentland. Interpolation using wavelet bases. *IEEE PAMI*, 16(4), April 1994.

[17] A. Pentland and B. Horowitz. Recovery of non-rigid motion and structure. *IEEE PAMI*, 13(7), July 1991.

[18] A. Pentland and S. Sclaroff. Closed-form solutions for physically-based shape modeling and recognition. *IEEE PAMI*, 13(7), July 1991.

[19] S. Sclaroff. *Deformable Solids and Displacement Maps: a Multi-scale Technique for Model Recovery and Recognition*. Master's thesis, MIT Media Laboratory, June 1991.

[20] S. Sclaroff and A. Pentland. Generalized Implicit Functions for Computer Graphics. In *Computer Graphics*, July 1991.

[21] E. Simoncelli and E. Adelson. *Subband Image Coding*, chapter Subband Transforms. Kluwer Academic Publishers, 1990. J. W. Woods, Ed.

[22] D. Terzopoulos. The Computation of Visible Surface Representations. *IEEE PAMI*, 10(4):417–438, July 1988.

[23] D. Terzopoulos and D. Metaxas. Dynamic 3-D models with local and global deformations: Deformable superquadrics. *IEEE PAMI*, 13(7):703–714, July 1991.

[24] D. Terzopoulos, A. Witkin, and M. Kass. Constraints on deformable models: Recovering 3d shape and nonrigid motion. *Artificial Intelligence*, 36:91–123, 1988.

[25] B. Vemuri and A. Radisavljevic. From global to local, a continuum of shape models with fractal priors. In *Proc. CVPR*, pages 307–313, 1993.

[26] Y. F. Wang and J. F. Wang. Surface reconstruction models using deformable models with interior and boundary constraints. *IEEE PAMI*, 14(5):572–579, May 1992.

[27] P. Whaite and F. Ferrie. From Uncertainty to Visual Exploration. *IEEE PAMI*, 13(10):1038–1049, October 1991.