MVA '92 IAPR Workshop on Machine Vision Applications Dec. 7-9,1992, Tokyo

MODAL DESCRIPTIONS FOR RECOGNITION AND TRACKING

Alex P. Pentland
The Media Laboratory,
Massachusetts Institute of Technology
20 Ames Street
Cambridge, MA 02139, USA

ABSTRACT

All shapes can be represented as deformations from a standard or prototypical shape; it is thought that this is how shape is represented in human perception. An object's *modes* are the eigenvectors of its elasticity matrix, and are a physically-motivated way to obtain a canonical description of shape in terms of deformation from a prototype. Modes provide an efficient and reliable method for recovering, recognizing, and tracking a 3-D solid models from 2-D and 3-D measurements. Several examples using this technology to recognize and track people will be presented.

1 INTRODUCTION

The representation of objects by their parts has a long tradition in computer-aided design, simulation, and in cognitive psychology. Indeed, in these areas it is the dominant strategy for representing complex 3-D objects. It is absolutely clear, therefore, that part representations are excellent for many computational and cognitive tasks. What is not so clear is how they might be useful in computer vision.

The first parts representation was suggested by Binford [5]; this is the idea of generalized cylinders. Unfortunately, the recovery of this type of representation seems to require elaborate line grouping and reasoning. Consequently, despite decades of effort, there are few reports of recovering such descriptions from real imagery [12]. Moreover, because such descriptions are often not unique it is unclear how they aid in object recognition.

The idea of generalized cylinders has subsequently been elaborated in two very different ways. One variation is due to Biederman [4], who suggested using the Cartesian product of qualitative properties such as tapering, cross-section, etc., in order to create a qualitative taxonomy of generalized cylinders. One advantage of this type of representation is that the properties can be chosen to be ones that are more easily recovered from imagery. Another is that it provides a way to define qualitative shape classes, an important problem in general-purpose vision. However only Dickinson, Pentland, and Rosenfeld [8] have reported being able to use this approach to recognize objects in real imagery, although Bergevin and Levine [3] have reported good success in interpreting vectorized line drawings.

Another alternative to generalized cylinders was suggested by Pentland in 1986 [13], who proposed a parametric version of generalized cylinders based on deformable superquadrics. Use of a parameterized implicit function, such as the superquadric, converts the problem of recovering a description into a relatively simple numerical optimization. Moreover, if the parameterization is orthogonal then the description is unique, making the recognition problem much easier. Dozens of authors have reported success at recovering this type of description from real data and then using it for recognition, e.g., [19, 15, 22].

Most recently, Pentland [16] has generalized this approach allow a large number of degrees of freedom, and to include the physical properties of objects. This has allowed difficult recognition problems, such as the recognition of people, to be addressed successfully. It has also allowed the construction of efficient Kalman filters for tracking both rigid and nonrigid objects. A summary of this recent work will be the topic of this paper.

2 THE REPRESENTATION

The modal representation may be thought of as describing objects using the force-and-process metaphor of modeling clay: shape is the result of pushing, pinching, and pulling on a lump of elastic material such as clay [13, 15]. Thus all shapes are represented as deformations from a standard or prototypical shape. It is thought that this is how shape is represented in human perception [13].

The mathematical formulation of the "a lump of clay" idea requires use of the finite element method (FEM), which is the standard engineering technique for describing physical behavior and deformation. Using the FEM we can characterize the elasticity of a prototypical "lump of clay", and calculate its *deformation modes*. These modes are the eigenvectors of the prototypes's stiffness matrix, and they provide a canonical description of all possible shapes in terms of deformation from the original prototypical shape.

In the FEM, energy functionals are formulated in terms of nodal displacements $\mathbf{U}$, and iterated to solve for the nodal displacements as a function of impinging loads $\mathbf{R}$:

$$\mathbf{M}\ddot{\mathbf{U}} + \mathbf{C}\dot{\mathbf{U}} + \mathbf{K}\mathbf{U} = \mathbf{R} \tag{1}$$

This equation is known as the FEM *governing equation*, where $\mathbf{U}$ is a $3n$ x 1 vector of the $(\Delta x, \Delta y, \Delta z)$ displacements of the n nodal points relative to the object's center of mass, $\mathbf{M}$, $\mathbf{C}$ and $\mathbf{K}$ are $3n$ by $3n$ matrices describing the mass, damping, and material stiffness between each point within the body, and $\mathbf{R}$ is a $3n$ x 1 vector describing the x, y, and z components of the forces acting on the nodes.

When a constant load is applied to a body it will, over time, come to an equilibrium condition described by

$$\mathbf{KU} = \mathbf{R} \qquad (2)$$

This equation is known as the *equilibrium governing equation*. The solution of the equilibrium equation for the nodal displacements $\mathbf{U}$ is the most common objective of finite element analyses.

In the type of shape modeling done in computer vision, sensor measurements are used to define virtual forces which deform the object to fit the data points. The equilibrium displacements $\mathbf{U}$ constitute the recovered shape. For additional detail, see reference [16].

2.1 Modal Analysis

To obtain an equilibrium solution $\mathbf{U}$, one integrates Equation 1 using an iterative numerical procedure at a cost proportional to the stiffness matrices' bandwidth. To reduce this cost we can transform the problem from the original nodal coordinate system to a new coordinate system whose basis vectors are the columns of an n x n matrix $\mathbf{P}$. In this new coordinate system the nodal displacements $\mathbf{U}$ become *generalized displacements* $\tilde{\mathbf{U}}$:

$$\mathbf{U} = \mathbf{P}\tilde{\mathbf{U}} \qquad (3)$$

Substituting Equation 3 into Equation 1 and premultiplying by $\mathbf{P}^T$ transforms the governing equation into the coordinate system defined by the basis $\mathbf{P}$:

$$\tilde{\mathbf{M}}\ddot{\tilde{\mathbf{U}}} + \tilde{\mathbf{C}}\dot{\tilde{\mathbf{U}}} + \tilde{\mathbf{K}}\tilde{\mathbf{U}} = \tilde{\mathbf{R}} \qquad (4)$$

where $\tilde{\mathbf{M}} = \mathbf{P}^T\mathbf{MP}$, $\tilde{\mathbf{C}} = \mathbf{P}^T\mathbf{CP}$, $\tilde{\mathbf{K}} = \mathbf{P}^T\mathbf{KP}$, and $\tilde{\mathbf{R}} = \mathbf{P}^T\mathbf{R}$. With this transformation of basis, a new system of stiffness, mass, and damping matrices can be obtained which has a smaller bandwidth then the original system.

The optimal basis $\mathbf{\Phi}$ has columns that are the eigenvectors of both $\mathbf{M}$ and $\mathbf{K}$ [2]. These eigenvectors are also known as the system's *free vibration modes*. Using this transformation matrix we have

$$\mathbf{\Phi}^T\mathbf{K}\mathbf{\Phi} = \mathbf{\Omega}^2, \qquad \mathbf{\Phi}^T\mathbf{M}\mathbf{\Phi} = \mathbf{I} \qquad (5)$$

where the diagonal elements of $\mathbf{\Omega}^2$ are the eigenvalues of $\mathbf{M}^{-1}\mathbf{K}$ and remaining elements are zero. When the damping matrix $\mathbf{C}$ is restricted to be *Rayleigh damping*, then it is also diagonalized by this transformation.

The lowest frequency modes are always the rigid-body modes of translation and rotation. The next-lowest frequency modes are smooth, whole-body deformations that leave the center of mass and rotation fixed. Compact bodies — solid objects like cylinders, boxes, or heads, whose dimensions are within the same order of magnitude — normally have low-order modes which are intuitive to humans: bending, pinching, tapering, scaling, twisting, and shearing. Some of the low-order mode shapes for a cube are shown in Figure 1. Bodies with very dissimilar dimensions, or which have holes, etc., can have very complex low-frequency modes.

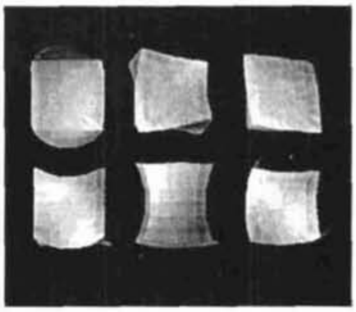


Figure 1: A few of the vibrations mode shapes of a 27 node isoparametric element.

2.2 Advantages

The modal representation provides a formulation whose degrees of freedom are orthogonal, and thus *decoupled*, and form a frequency-ordered orthonormal basis set analogous to the Fourier transform.

By decoupling the degrees of freedom we achieve substantial advantages:

- The fitting problem has a simple, efficient, closed-form solution.
- The model's intrinsic complexity can be adjusted to match the number of degrees of freedom in the data measurements, so that the solution can always be made overconstrained.
- When overconstrained, the solution is unique, except for rotational symmetries and degenerate conditions. Thus the solution is well-suited for recognition and database tasks.

Moreover, because the representation employed is based on the Finite Element method, the dynamics of the observed object can be accurately modeled. As a consequence, optimal estimates of object motion and shape can be made even in non-stationary enviroments, and physical predictions/simulation can be made directly from recovered models. sequences.

2.3 Modeling Using Implicit Functions

It is important to have a unified representation for both geometric and physical modeling. Our approach is to combine the modal shape deformations defined above with an implicit function surface such as a sphere or cube. This combination gives us the advantage of being able to accurately and simply describe physical deformations, and yet to be able to use the implicit function representation's *inside-outside* function for contact detection and model fitting [20, 14, 9].

In object-centered coordinates $\mathbf{r} = [r, s, t]^T$, the implicit equation of a spherical surface is

$$f(\mathbf{r}) = f(r, s, t) = r^2 + s^2 + t^2 - 1.0 = 0.0 \qquad (6)$$

This equation is also referred to as the surface's *inside-outside function*, because to detect contact between a point $\mathbf{X}_p = [X_p, Y_p, Z_p]^T$ and the volume bounded by this surface, one simply substitutes the coordinates of $\mathbf{X}$ into the function f. If the result is negative, then the point is inside the surface. Generalizations of this basic operation may be used to find line-surface intersections or surface-surface intersections.

A solid defined in this way can be easily positioned and oriented in global space, by transforming the implicit function to global coordinates, $\mathbf{X} = [X, Y, Z]^T$ we get [20]:

$$\mathbf{X} = \mathcal{R}\mathbf{r} + \mathbf{b} \qquad (7)$$

where $\mathcal{R}$ is a rotation matrix, and $\mathbf{b}$ is a translation vector. The implicit function's positioned and oriented (rigid) inside-outside function becomes (using Equation 7):

$$f(\mathbf{r}) = f(\mathcal{R}^{-1}(\mathbf{X} - \mathbf{b})). \qquad (8)$$

Any set of implicit shape functions can be generalized by combining them with a set of global deformations $\mathcal{D}$ with parameters $\mathbf{m}$. For particular values of $\mathbf{m}$ the new deformed surface is defined using a deformation matrix $\mathcal{D}_{\mathbf{m}}$:

$$\mathbf{X} = \mathcal{R}\mathcal{D}_{\mathbf{m}}\mathbf{r} + \mathbf{b} \qquad (9)$$

In our system the deformations used are the *modal shape polynomial functions*, defined by transforming the original finite element shape functions to the modal coordinate system (see [16] and Appendix A of this paper). These polynomials are a function of $\mathbf{r}$, Equation 9 becomes:

$$\mathbf{X} = \mathcal{R}\mathcal{D}_{\mathbf{m}}(\mathbf{r})\mathbf{r} + \mathbf{b} \qquad (10)$$

The inside-outside function, with nonrigid deformations becomes (using Equation 10):

$$f(\mathbf{r}) = f(\mathcal{D}_{\mathbf{m}}^{-1}(\mathbf{r})\mathcal{R}^{-1}(\mathbf{X} - \mathbf{b})) \qquad (11)$$

This inside-outside function is valid as long as the inverse polynomial mapping $\mathcal{D}_{\mathbf{m}}^{-1}(\mathbf{r})$ exists. In cases where a set of deformations has no closed-form inverse mapping, Newton-Raphson and other numerical iterative techniques have to be used.

This method of defining geometry, therefore, provides an inherently more efficient mathematical formulation for contact detection than geometric representations such as polygons or splines. See Pentland and Williams [14] and Sclaroff and Pentland [20] for a discussion of the computational complexity of contact detection algorithms.

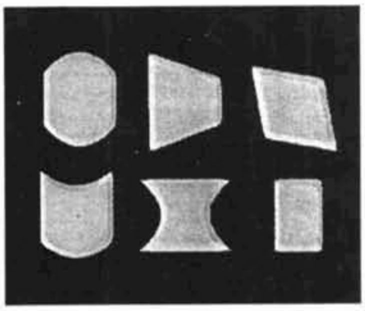


Figure 2: A few of the vibrations mode shapes of a cube, using idealized deformations

3 IDEALIZED MODES

For applications that do not require accurate physical modeling, such as object recognition, we have found that it is adequate to use a single set of particularly simple deformations derived using idealized elasticity properties. Moreover, because the elastic properties of the model are of no concern, it is sufficient to set $\tilde{\mathbf{K}}$ to be the identity matrix, except for rigid-body modes which have zero stiffness.

The entries of the idealized deformation matrix $\mathcal{D}_{\mathbf{m}}$ for these idealized modes are as follows,

$$\begin{aligned}
d_{00} &= m_6 + sm_{12} + tm_{15} - (m_{13} + m_{16})sgn(r) \\
&\quad -m_{14} - m_{17} \\
d_{01} &= m_{11} + 2s(m_{13} + sgn(r)m_{14}) \\
d_{02} &= m_{10} + 2t(m_{16} + sgn(r)m_{17}) \\
d_{10} &= m_{11} + 2r(m_{19} + sgn(s)m_{20}) \\
d_{11} &= m_7 + rm_{18} + tm_{21} - (m_{19} + m_{22})sgn(s) \\
&\quad -m_{20} - m_{23} \\
d_{12} &= m_9 + 2t(m_{22} + sgn(s)m_{23}) \\
d_{20} &= m_{10} + 2r(m_{25} + sgn(t)m_{26}) \\
d_{21} &= m_9 + 2s(m_{28} + sgn(t)m_{29}) \\
d_{22} &= m_8 + rm_{24} + sm_{27} - (m_{25} + m_{28})sgn(t) \\
&\quad -m_{26} - m_{29}
\end{aligned} \qquad (12)$$

where $\mathbf{m} = [m_0, m_1, \ldots, m_{p-1}]^T$ is a p x 1 vector of the *modal amplitudes*, and $\mathbf{r} = [r,\ s,\ t]^T =$ is the coordinate of a point in undeformed space.

The modal amplitudes m_i formulated in this way have an intuitive meaning. Modal amplitudes m_0 - m_5 are the rigid body modes of translation and rotation, m_6 - m_8 are the x, y, and z sizes, m_9 - m_{11} are shears about the x, y, and z axes and the rest are bends, tapers and pinches in various axes. Figure 2 illustrates a few of these idealized deformation modes for a cube; the reader should compare this figure to Figure 1.

4 RECOVERING 3-D MODELS

Let us assume that we are given m three-dimensional sensor measurements (in the global coordinate system) that originate from the surface of a single object

$$\mathbf{X}^w = [x_1^w, y_1^w, z_1^w, \cdots, x_m^w, y_m^w, z_m^w]^T \qquad (13)$$

We then attach virtual springs between these sensor measurement points and particular nodes on our deformable model. This defines an equilibrium equation whose solution $\mathbf{U}$ is the desired fit to the sensor data. Consequently, for m nodes with corresponding sensor measurements, we can calculate the virtual loads $\mathbf{R}$ exerted on the undeformed object while fitting it to the sensor measurements. For node k these loads are simply

$$[r_{3k}, r_{3k+1}, r_{3k+2}]^T = [x_k^w, y_k^w, z_k^w]^T - [x_k, y_k, z_k]^T \tag{14}$$

where

$$\mathbf{X} = [x_1, y_1, z_1, \cdots, x_n, y_n, z_n]^T \tag{15}$$

are the nodal coordinates of the undeformed object in the object's coordinate frame. When sensor measurements do not correspond exactly with existing nodes, the loads can be distributed to surrounding nodes using the interpolation functions used to define the finite element model, as described in [16].

Thus to fit a deformable solid to the measured data we solve the following equilibrium equation:

$$\mathbf{KU} = \mathbf{R} \tag{16}$$

where the loads $\mathbf{R}$ are as above, the material stiffness matrix $\mathbf{K}$ is as described above and in [16], and the equilibrium displacements $\mathbf{U}$ are to be solved for. The solution to the fitting problem is simply

$$\mathbf{U} = \mathbf{K}^{-1}\mathbf{R} \tag{17}$$

The difficulty in calculating this solution is the large dimensionality of $\mathbf{K}$, so that iterative solution techniques are normally employed.

However a closed-form solution is available simply by converting this equation to the modal coodinate system. This is accomplished by substituting $\mathbf{U} = \mathbf{\Phi}\tilde{\mathbf{U}}$ and premultiplying by $\mathbf{\Phi}^T$, so that the equilibrium equation becomes

$$\mathbf{\Phi}^T\mathbf{K}\mathbf{\Phi}\tilde{\mathbf{U}} = \mathbf{\Phi}^T\mathbf{R} \tag{18}$$

or equivalently

$$\tilde{\mathbf{K}}\tilde{\mathbf{U}} = \tilde{\mathbf{R}} \tag{19}$$

where $\tilde{\mathbf{R}} = \mathbf{\Phi}^T\mathbf{R}$ and $\tilde{\mathbf{K}} = \mathbf{\Phi}^T\mathbf{K}\mathbf{\Phi}$ is a *diagonal* matrix. Again, note that the calculation of $\mathbf{\Phi}$ needs to be performed only once as a precomputation, and then stored for all future applications. Further, it is normally not desirable to use all of the eigenvectors (as explained below), so that the $\mathbf{\Phi}$ matrix remains of managable size even when using large numbers of nodes. In our implementation $\mathbf{\Phi}$ is normally a 30 x $3n$ matrix, where n is the number of nodes.

The solution to the fitting problem, therefore, is obtained by inverting the diagonal matrix $\tilde{\mathbf{K}}$:

$$\tilde{\mathbf{U}} = \tilde{\mathbf{K}}^{-1}\tilde{\mathbf{R}} \tag{20}$$

Note, however, that as this formulation is posed in the object's coordinate system the rigid body modes have zero eigenvalues, and must therefore be solved for separately by setting $\tilde{u}_i = \tilde{r}_i$, $1 \leq i \leq 6$. The complete solution may be written in the original nodal coordinate system, as follows

$$\mathbf{U} = \mathbf{\Phi}(\tilde{\mathbf{K}} + \mathbf{I}_6)^{-1}\mathbf{\Phi}^T\mathbf{R} \tag{21}$$

where $\mathbf{I}_6$ is a matrix whose first six diagonal elements are ones, and remaining elements are zero.[1]

The major difficulty in calculating this solution occurs when there are fewer degrees of freedom in sensor measurements than in the nodal positions — as is normally the case in computer vision applications. Previous researchers have suggested adopting heuristics such as smoothness and symmetry to obtain a well-behaved solution; however in many cases the observed objects are neither smooth nor symmetric, and so an alternative method is desirable.

A better method is to discard some of the high-frequency modes, so that the number of degrees of freedom in $\tilde{\mathbf{U}}$ is equal to or less than the number of degrees of freedom in the sensor measurements. To accomplish this, one simply row and column reduces $\tilde{\mathbf{K}}$, and column reduces $\mathbf{\Phi}$ so that their rank is less than or equal to the number of available sensor measurement degrees of freedom. The motivation for this strategy is that:

- When there are fewer degrees of freedom in the sensor measurements than in the model, the high-frequency modes cannot in any sense be accurate, as there is insufficient data to constrain them. Their value primarily reflects the smoothness heuristic employed.
- While the high-frequency modes will not contain information, they are the *dominant* factor determining the cost of the solution, as they are both numerous and require the use of very small time steps [14].

Perhaps the most interesting consequence of discarding some of the high-frequency modes, however, is that it allows Equation 21 to provide a generically *overconstrained* estimate of object shape. Note that discarding high-frequency modes is **not** equivalent to a smoothness assumption, as sharp corners, creases, etc., can still be obtained. What we cannot do with a reduced-basis modal representation is place many creases or spikes close together.

4.1 Using 2-D Contours and Points

In the case where we are given only 2-D information we can still employ the same equations to estimate shape, however we must generalize Equation 21 to reflect the uncertainty we have about the z coordinate of each point. This can be accomplished by altering Equation 21 to reflect the fact that some sensor measurements are more certain than others. We accomplish this by introducing a $3n$ x $3n$ diagonal weighting matrix $\mathbf{W}$:

$$\tilde{\mathbf{U}} = \tilde{\mathbf{K}}_6^{-1}(\mathbf{W}\mathbf{\Phi})^T\mathbf{R} \tag{22}$$

[1] Inclusion of the matrix $\mathbf{I}_6$ into Equation 21 may also be interpreted as adding an external force that constrains the solution to have no residual translational or rotational stresses.

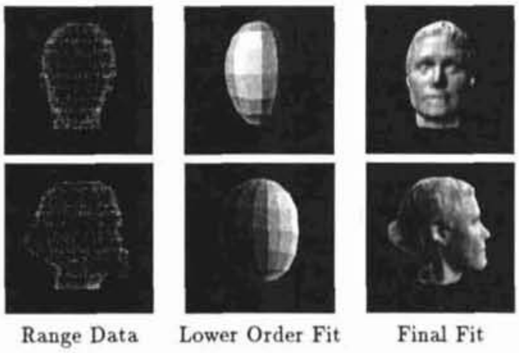


Figure 3: Fitting laser rangefinder data of a human face. Left column: original range data, Middle column: recovered 3-D model using only low-order modes, Left Column: full recovered model.

The diagonal entries of $\mathbf{W}$ are inversely proportional to the uncertainty (variance) of the data associated with each of the nodal coordinates. The effect of $\mathbf{W}$ is to make the strength of the virtual springs associated with each data point reflect the uncertainty of the measurement.

4.2 An Example Using 3-D Point Data

The left-hand image of Figure 3 shows an example using 360° laser rangefinder data of a human head. There are about 2500 data points. Equation 21 was then used to estimate the shape, using only the low-frequency 30 modes. The low-order recovered model is shown in the middle column; because of the large number of data points execution time on a Sun 4/330 was approximately 3 seconds. It can be seen that the low-order modes provide a sort of qualitative description of the overall head shape.

A full-dimensionality recovered model is shown in the right-hand image of 3. In the ThingWorld system [14, 15], rather than describing high-frequency surface details using a finite element model with as many degrees of freedom as there are data points, we normally augment a low-order finite element model with a spline description of the surface details. This provides us with a two-layered representation (low-order finite element model + surface detail spline description = final model) that we find to be both more efficient to recover and more useful in recognition, simulation, and visualization tasks than a fully-detailed finite element model.

5 OBJECT RECOGNITION

Perhaps the major drawback of previous shape-modeling techniques is that they have not been useful for recognition, comparision, or other database tasks. This is because they normally have more degrees of freedom than there are sensor measurements, so that the recovery process is underconstrained. Therefore, although heuristics such as smoothness or symmetry can be used to obtain a solution, they do not produce a stable, unique solution.

The major problem is that when the model has more degrees of freedom than the data, the model's nodes can slip about on the surface. The result is that there are an infinite number of valid combinations of nodal positions for any particular surface. This difficulty is common to all spline and piecewise polynomial representations, and is known as the *knot problem*.

For all such representations, 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.

The modal representation, assuming that all modes are employed, decouples the degrees of freedom, but it does not by itself reduce the total number of degrees of freedom. Consequently, a complete modal representation suffers from the same problems as all of the other representations.

The obvious solution to the problem of non-uniqueness is to discard enough of the high-frequency modes that we can obtain an overconstrained estimate of shape, as was done for the shape recovery problem above. Use of a reduced-basis modal representation results in a *unique* representation of shape because the modes (eigenvectors) form an orthonormal basis set. Therefore, there is only one way to represent an object, and that is in its canonical position.

Further, because the modal representation is frequency-ordered, it has stability properties that are similar to those of a Fourier decomposition. Just as with the Fourier decomposition, an exact subsampling of the data points points does not change the low-frequency modes. Similarly, irregularities in local sampling and measurement noise tend to primarily affect the high-frequency modes, leaving the low-frequency modes relatively unchanged.

The primary limitation of this uniqueness property stems from the linearization of rotation. Because the rotations are linearized, it is impossible to uniquely determine an object's rotation state. As a consequence object symmetries can lead to multiple descriptions, and errors in measuring object orientation will cause commensurate errors in shape desription.

Thus by employing a reduced-basis modal representation we can obtain overconstrained shape estimates that are also unique except for rotational symmetries. To compare objects l and k with known mode values $\tilde{\mathbf{U}}^l$ and $\tilde{\mathbf{U}}^k$, one simply compares the two vectors of mode values:

$$\varepsilon = \frac{\tilde{\mathbf{U}}^l \cdot \tilde{\mathbf{U}}^k}{\|\tilde{\mathbf{U}}^l\|\|\tilde{\mathbf{U}}^k\|} \qquad (23)$$

Vector norms other than the dot product can also be employed; in our experience all give roughly the same recognition accuracy.

To recognize a recovered model with estimated mode values $\tilde{\mathbf{U}}$, one compares the recovered mode values to the mode values of all of the p known models:

$$\varepsilon_k = \frac{\tilde{\mathbf{U}} \cdot \tilde{\mathbf{U}}^k}{\|\tilde{\mathbf{U}}\| \|\tilde{\mathbf{U}}^k\|} \qquad k = 1, 2, \ldots, p \tag{24}$$

The known model k with the maximum dot product ε_k is the model best matching the recovered model, and thus declared to be the model recognized. Note that for each known model k, only the vector of mode values $\tilde{\mathbf{U}}^k$ needs to be stored.

The first six entries of $\bar{\mathbf{\Phi}}$ are the rigid-body modes (translation and rotation), which are normally irrelevant for object recognition. Similarly, the seventh mode (overall volume) is sometimes irrelevent for object recognition, as many machine vision techniques recover shape only up to an overall scale factor. Thus rather than computing the dot product with all of the modes $\tilde{\mathbf{U}}$, we typically use only modes number eight and higher, e.g.,

$$\varepsilon_k = \frac{\sum_{i=8}^{i=m} \tilde{\mathbf{u}}_i \tilde{\mathbf{u}}_i^k}{\sqrt{\sum_{i=8}^{i=m} \tilde{\mathbf{u}}_i^2} \sqrt{\sum_{i=8}^{i=m} (\tilde{\mathbf{u}}_i^k)^2}} \qquad k = 1, 2, \ldots, p \tag{25}$$

where m is the total number of modes employed. By use of this formula we obtain translation, rotation, and scale-invariant matching.

The ability to compare the shapes of even complex objects by a simple dot product makes the modal representation well suited to recognition, comparison, and other database tasks. In the following section we will evaluate the reliablity of the combined shape recovery/recognition process.

5.1 Recognition: 3-D Data

To assess accuracy, we conducted an experiment to recover and recognize face models from range data generated by a laser range finder. In this experiment we obtained laser rangefinder data of eight people's heads from a five different viewing directions: the right side (−90°), halfway between right and front (−45°), front (0°), halfway between front and left (45°), and the left side (90°). We have found that people's heads are only approximately symmetric, so that the ±45° and ±90° degree views of each head have quite different detailed shape. In each case the range data was from the forward-facing, visible surface only.

Data from a 360° scan around each head was then used to form the stored model of each head that was later used for recognition. Full-detail versions of these eight reference models are shown in Figure 4; note that in some cases a significant amount of the data is missing. As previously, only the low order 30 deformation modes were used in the shape extraction and recognition procedure. Because the low order modes provide a coarse, qualitative summary of the object shape (see the middle column of Figure 3) they can be expected to be the most stable with respect to noise and viewpoint change. Total execution time on a standard Sun 4/330 averaged approximately 5 seconds per fitting and recognition trial.

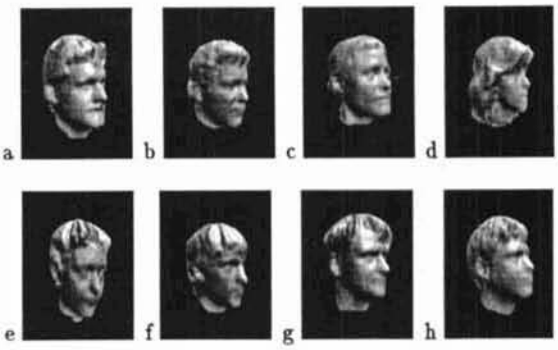


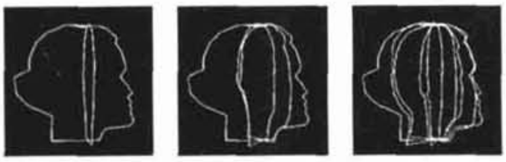
Figure 4: Eight heads used in our recognition experiment. Note that in some cases there is significant missing data.

	−90°	−45°	0°	45°	90°
a	0.16	-0.19	-0.24	-0.19	-0.12
b	0.26	0.28	0.35	0.37	0.30
c	**0.88**	**0.99**	**0.99**	**0.98**	**0.99**
d	0.03	0.15	0.21	0.21	0.11
e	0.06	-0.13	-0.10	-0.04	-0.06
f	0.58	0.44	0.46	0.46	0.50
g	0.53	0.53	0.52	0.50	0.58
h	0.42	0.47	0.53	0.49	0.50

Figure 5: Recognizing faces from five different points of view.

Recognition was accomplished by first recovering a 3-D model from the visible-surface range data, and then comparing the recovered mode values to the mode values stored for each of the three known head models using Equation 25. The known model producing the largest dot product was declared to be the recognized object. The first seven modes were not employed, so that the recognition process was translation, rotation, and scale invariant.

Figure 5 illustrates typical results from this experiment. The top row of Figure 5 illustrates the five models recovered from range data from the front, visible surface using viewpoints of −90°, −45°, 0°, 45°, and 90°. Each of these recovered head models look similar, and more importantly have approximately the same deformation mode values $\tilde{\mathbf{U}}$, despite the wide variations in input data. Modes 8 through 30 of these recovered models were then compared to each



2 Contours 3 Contours 5 Contours

Figure 6: Set of contour groups used in head recognition from contours example. These contours were taken from the same head depicted in Figure 3 (**Head c** in Figure 4).

of the stored head models. The dot products obtained are shown below each recovered head model.

In Figure 5 all of the input data was views of Kim (depicted as "head c" in the tables). As can be seen, the dot products between recovered 3-D model and known model are quite large for Kim's head model. In fact, in this example the smallest correct dot product is almost three times the magnitude of any of the incorrect dot products; the same was also true for range data of the other subjects.

In this experiment 92.5% accurate recognition was obtained. That is, we successfully recovered 3-D models and recognized each of the eight test subjects from each of the five different views with only three errors. Analysis of the recognition results showed that, while the average dot product between different reference models was 0.31 (72°), the average dot product between models recovered from different views of the same person was 0.95 (18°). Thus recognition was typically extremely certain. All three errors were from front-facing views, where relatively few discriminating features are visible; remember that only overall head shape, and not details of surface shape, were available to the recognition procedure as only 30 modes were employed.

5.2 Recognition: 2-D Data

In a similar head recovery and recognition experiment, we used a few 2-D head contours instead of full range data to see how well our techniques performed in the case of sparse silhouette data. In this experiment, we recovered heads from 2, 3, and finally 5 contours, in order to approximate the information available in an active vision scenario. In each trial, the contours were spaced evenly in rotation. An example of the contours used in this experiment is shown in Figure 6; these contours were taken from the same head depicted in Figure 3 ("head c" in Figure 4).

As in the previous experiment, the recovered heads were compared against the full-detail versions of the reference heads shown in Figure 4, and the model producing the largest dot product was declared to be the recognized object. Heads were compared using the scale, rotation, and translation invariant matching of Equation 25.

In our experiments with two contours, recognition accuracy averaged 93.75%. Accuracy improved as more contours were added until 96.875 percent of the heads were correctly identified when 5 contours were used. The results were not as good if the contours did not include the traditional side "silhouette," and performed best when the data's spring attachment was smoothed out more across the surface. Total execution times were slightly greater than those for the full data experiment, averaging 5 seconds per fitting and recognition trial on a Sun 4/330. The greater execution time is attributable to the more careful distribution and smoothing of spring attachment between contours and the underlying deformable model.

6 DYNAMIC TRACKING

In the previous sections we have addressed *static* shape estimation and recognition. For sequences, however, it is necessary to also consider the *dynamic* properties of the body and of the data measurements. The Kalman filter is the standard technique for obtaining estimates of the state vectors of dynamic models, and for predicting the state vectors at some later time. Outputs from the Kalman filter are the optimal (weighted) least-squares estimate for non-Gaussian noises.

Kalman filtering has been used in many motion estimation applications [6, 10], but normally it is both too expensive and requires too much storage to apply to the large number of variables typical of a whole-body finite element model. However, because the modal representation allows us to summarize the dynamic state of the body with only a small number of parameters, development of a Kalman filter is straightforward.

Because of space limitations, we will forego the detailed development of the Kalman filter equations. Readers interested in this detail are referred to references [17, 18, 1].

6.1 Tracking Examples

Figure 7(a) shows one frame from a sequence of X-ray images, with the zero-crossing edge contours overlayed. From these contours the 3-D shape was estimated using Equation 22. The resulting shape is shown in Figure 7(b) as a 3-D wireframe overlayed on the original X-ray data. Figure 7(c) shows the recovered model from the side. Because only bounding contour information was available, the shape estimated along the z axis (shown in Figure 7(c)) is determined by finding the minimum stress state that still fits the bounding contour. The z-axis shape cannot, therefore, be regarded as accurate but only as plausible and consistent. Note that use of a minimum stress criterion for solution means that symmetric and mirror symmetric shapes are preferred. Execution time was approximately one second on a standard Sun 4/330. For additional detail see Pentland and Sclaroff [16].

Figure 8 shows an example of recovering non-rigid motion from contour information. The 3-D shape and motion of the heart ventricle was tracked over time using the contour information shown at the top within each box of Figure 8.

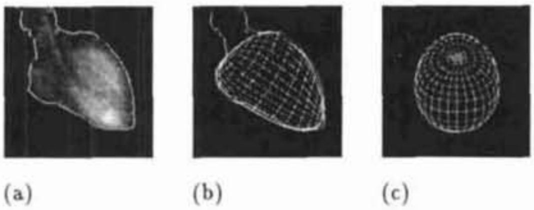


Figure 7: An example showing the use of a 2-D image contour to recover a 3-D deformable solid model. The original image and contour are shown in (a). The model is recovered from the contour as shown in (b). An orthogonal side view is shown in (c).

For each frame Equation 22 was used to obtain an estimate of 3-D shape from the contour information (see Figure 7). These shape estimates were then integrated in a Kalman filter formulation, as described in Pentland and Horowitz [17]. As in the single-image case, deformations along the z axis are determined by finding the minimum stress state that still fits the bounding contour. The z-axis deformations cannot, therefore, be regarded as accurate but only as plausible. Execution time was approximately one second per frame on a standard Sun 4/330.

Figure 9 illustrates a more complex example of tracking rigid and non-rigid motion. This figure shows three frames from a twelve image sequence of a well-known tin woodsman caught in the act of jumping. Despite the limited range of motion, this example is a difficult one because of the poor quality optical flow, due to pronounced highlights on thighs and other parts of the body.

In this example an articulated 3-D model was constructed by hand, with spring-like attachment constraints inserted between the various body parts. In this manner the combined behavior of the various parts were constrained to be consistent with the physics of the situation: parts must stay connected, movement by one part causes an equal but opposite reactions among the other parts, and inertia is conserved.

We then calculated optical flow by use of a block-wise Horn-Schunk algorithm, and then our Kalman filter formulation used to estimate the motions of the various parts. The between-part attachment constraints then introduce additional forces that enforce the conservation of force and inertia. All of these forces are then integrated to produce a final physically-consistent estimate of the overall motion at each instant in time.

The estimated motions for this sequence are illustrated by the bottom row of Figure 9. As can be seen by comparing the 3-D motion of the model with that in the original image, the resulting tracking is reasonably accurate. For additional information, see references [17, 18].

6.2 A Computer Graphics Application

Most interactive computer applications require harnessing the user with wires. This detracts both from the enjoyment

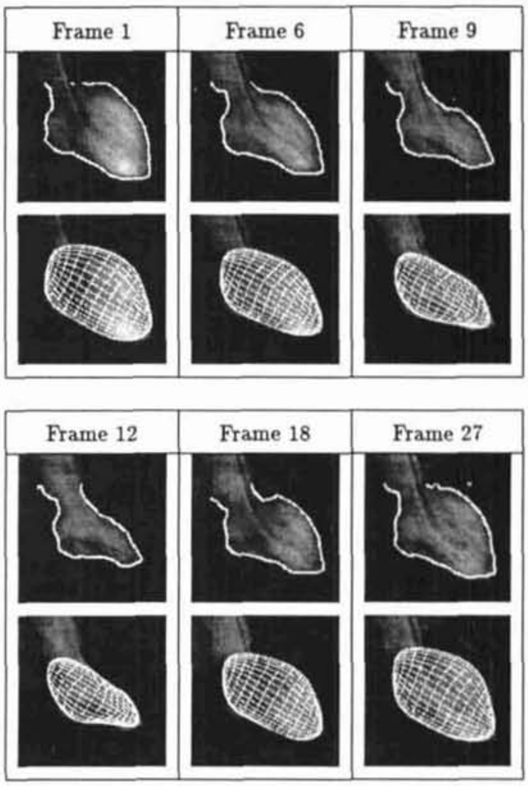


Figure 8: A heart's nonrigid motion as it was recovered from contours taken from a motion sequence. The contours, extracted via a simple threshold and zero crossing scheme, are shown along the top. The deformable model recovered from these contours is shown in wireframe.

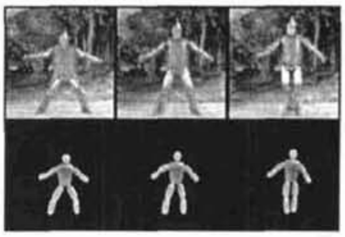


Figure 9: Three frames from an image sequence showing tracking of a jumping man using an articulated, physically-based model. Note that despite poor quality optical flow (due to pronounced highlights on thighs and other parts of the body) the overall tracking is reasonably accurate.

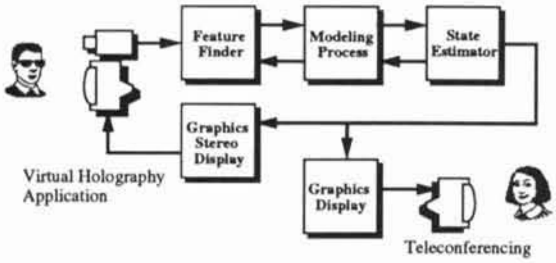


Figure 10: System organization

of the experience, and from the practicality of the system for day-to-day use. We have therefore used our tracking techniques to develop a completely passive system that provides "real-time" estimates of position and orientation, in a manner similar to the Polhemus sensor, but without the wires.

Our system uses a single CCD camera to estimate the low-order rigid-body modes. In initial testing, our system has shown itself to be competitive in accuracy with the Polhemus sensor. Our system has so far been applied only to the problem of tracking the user's head, however there is nothing in the formulation that is specific to the human head.

We have used this system in two separate applications: virtual holography, and teleconferencing. In the virtual holography application, a stereoscopic display is controlled by the user's head position so that the user sees an apparently solid object before him. He can look "around" the displayed objects, and see what they look like from various viewing positions. In the teleconferencing application, the user's head position is tracked, and used to control display of a 3-D model of the user's head to other teleconference participants.

In both applications there is a requirement for precise synchronization of action (display) with the exterior physical state (users head position). We have developed a combination of dataflow techniques and optimal linear prediction to obtain close synchronization between user movement and computer generated display, as is illustrated in Figure 10. This organization also allows the system to be distributed across several different computing engines.

Figure 11(a) illustrates the system operating in the virtual holography mode. In this mode the head position is tracked and used to control a stereographic display, thus simulating the experience of viewing a real object. Note the camera placed on top of the computer monitor; this is the sole input to the system.

Figure 11(b) compares the accuracy of the Polhemus sensor and our Kalman filter for tracking head position; the Kalman filter output has been calibrated to the Polhemus output. The horizontal axis is time, and the vertical axis is the x-coordinate of a user's head as he moves about in a general manner (including head rotations). The solid line is the Polhemus estimate of position, the dashed line the Kalman filter estimate of position.

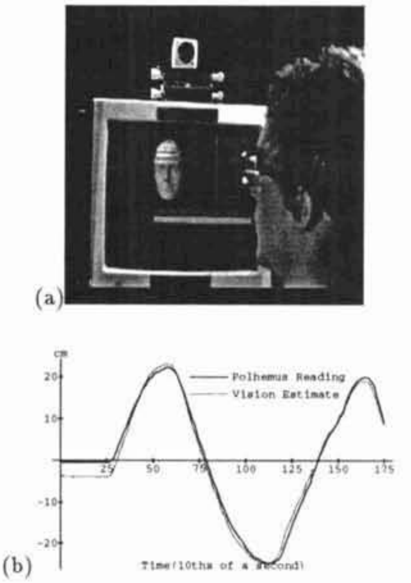


Figure 11: (a) The head tracking system in virtual holography mode, note camera at top of monitor. (b) Comparison of Polhemus and Kalman filter estimates of head position.

As can be seen, the Kalman and Polhemus estimates of position are quite similar. At the start the Kalman filter has no information about head position, just an initial guess. Once the head starts moving, the estimates quickly converge to the correct value. Although this is a very limited data sample, it is sufficient to form an initial estimate the Kalman filter's accuracy. Taking the Polhemus sensor's output as the "true" position signal, then the signal-to-noise ratio of the Kalman filter output is 22dB (0.6 % error). This is comparable to the advertised signal-to-noise ratio of the Polhemus sensor.

7 SUMMARY

We have described an efficient method for recovering, recognizing, and tracking 3-D solid models using either 2-D or 3-D sensor measurements. Because the recovered 3-D shape description is unique except for rotational symmetries, we may efficiently measure the similarity of different shapes by simply calculating normalized dot products between the mode values $\tilde{\mathbf{U}}$ of various objects. Such comparisons may be made position, orientation and/or size independent by simply excluding the first seven mode amplitudes. Thus the modal representation seems likely to be useful for applications such as object recognition and spatial database search.

The major weaknesses of our current method are

- The need to estimate initial object orientation to within ±15°, as in our formulation rotational variation has been linearized.

- The need to segment data into simple, approximately convex "blobs" in a stable, viewpoint-invariant manner.

In the current implementation of our system we use a standard method-of-moments to obtain initial estimates of object orientation, and a minimum description length technique to produce segmentations [7]. We are now working to integrate the segmentation procedure of Dickinson, Pentland, and Rosenfeld [8] to produce an integrated segmentation-fitting-recognition system.

References

[1] Azarbayejani, A., Starner, T., Horowitz, B., Pentland, A., (1992)"Interactive Graphics Without The Wires," *IEEE Tran. Pattern Analysis and Machine Intelligence*, special issue on computer graphics and vision, *in press.*

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

[3] Bergevin, R. and Levine, M. (1989) "Generic Object Recognition: Building Coarse 3D Descriptions from Line Drawings", Proceedings, IEEE Workshop on Interpretation of 3D Scenes, Austin, TX, November 1989, pp 68–74.

[4] I. Biederman. Recognition-by-Components: A Theory of Human Image Understanding. *Psychological Review*, 94(2):115–147, 1987.

[5] T. Binford. Visual Perception by Computer. *Presented at The IEEE Conference on Systems and Control*, December 1971.

[6] T. J. Broida and R. Chellappa. Estimation of Object Motion Parameters from Noisy Images, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(1):90–99. January 1986.

[7] T. Darrell, S. Sclaroff, and A. Pentland. Segmentation by Minimal Description. In *Proc. Third International Conference on Computer Vision*, December 1990.

[8] Dickinson, S., Pentland, A., and Rosenfeld, A., (1992) From Volumes to Views: An Approach to 3-D Object Recognition, *CVGIP: Image Understanding.* Vol. 55, No. 2, March, pp. 130-154.

[9] I. Essa. *Contact Detection, Collision Forces and Friction for Physically-Based Virtual World Modeling.* Master's thesis, Dept. of Civil Engineering, M.I.T., 1990.

[10] O. D. Faugeras, N. Ayache, and B. Faverjon Building Visual Maps by Combining Noisy Stereo Measurements, *Proc. IEEE Conf. on Robotics and Automation*, San Francisco, CA., April 1986.

[11] B. Friedland. *Control System Design.* McGraw-Hill, 1986.

[12] R. Mohan and R. Nevatia. Using Perceptual Organization to Extract 3D Structures. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11(11):1121–1139, November 1989.

[13] A. Pentland. Perceptual Organization and Representation of Natural Form. *Artificial Intelligence*, 28(3):293–331, 1986.

[14] A. Pentland and J. Williams. Good Vibrations : Modal Dynamics for Graphics and Animation. *Computer Graphics*, 23(4):215–222, 1989.

[15] A. Pentland. Automatic Extraction of Deformable Part Models. *International Journal of Computer Vision*, 107–126, 1990.

[16] Alex Pentland and Stan Sclaroff. Closed form solutions for physically based shape modeling and recovery. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(7):715–729, July 1991.

[17] Alex Pentland and Bradley Horowitz. Recovery of nonrigid motion and structure. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(7):730–742, July 1991.

[18] Alex Pentland, Bradley Horowitz, and Stan Sclaroff. Non-rigid motion and structure from contour. In *IEEE Workshop on Visual Motion*, pages 288–293. IEEE Computer Society, 1991.

[19] F. Solina and R. Bajcsy. Recovery of Parametric Models from Range Images: The Case for Superquadrics with Global Deformations. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12(2):131–147, 1990.

[20] Stan Sclaroff and Alex Pentland. Generalized implicit functions for computer graphics. *Computer Graphics*, 25(4):247–250, 1991.

[21] D. Terzopoulos, A. Witkin, and M. Kass. Symmetry-Seeking Models for 3-D Object Reconstruction. In *Proc. First Conference on Computer Vision*, pages 269–276, London, England, December 1987.

[22] Demetri Terzopoulos and Dimitri Metaxas. Dynamic 3d models with local and global deformations: Deformable superquadrics. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(7):703–714, July 1991.