

Towards 3D Hand Tracking using a Deformable Model

(submitted to Second International Conference on Automatic Face and Gesture Recognition)

Tony Heap and David Hogg
School of Computer Studies, University of Leeds, Leeds LS2 9JT, UK.
(a.jh, dch)@scs.leeds.ac.uk

Abstract

In this paper we first describe how we have constructed a 3D deformable Point Distribution Model of the human hand, capturing training data semi-automatically from volume images via a physically-based model. We then show how we have attempted to use this model in tracking an unmarked hand moving with 6 degrees of freedom (plus deformation) in real time using a single video camera. In the course of this we show how to improve on a weighted least-squares pose parameter approximation at little computational cost. We note the successes and shortcomings of our system and discuss how it might be improved.

1 Motivations

There has long been a need for a vision-based hand tracking system which is capable of tracking movement with 6 degrees of freedom (DOF), along with articulation information, whilst being as unintrusive as possible. The use of hand markings or coloured gloves and the need for highly constrained environments are undesirable. Such a system should also be as widely available as possible; this implies low-cost technology, so ideally a single camera input should be used and real-time performance should be possible on a standard workstation.

From the plethora of work on vision-based hand tracking, relatively few have tackled the task of extracting full 6 DOF hand position *and* articulation. Notable successes have been due to Dorner [1], whose goal was American Sign Language interpretation, and Rehg and Kanade [2] who developed a system called *DigitEyes*. Both made use of a manually-constructed articulated hand model with *a priori* knowledge of inter-joint distances and valid pivot angles. Dorner relied on multi-coloured gloves to aid tracking; Rehg however tracked only from edge information, but had to revert to stereo input in order to track full hand articulation. Neither could achieve real-time performance without the use of specialised hardware, and a homogeneous background was always used.

With a view to overcoming these limitations, our tracker is based on a 3D version of the Point Distribution Model (PDM) [3]; this is a statistically-derived deformable model which affords several advantages:

- The model is constructed from real-life examples of hands in various positions, giving an accurate model which implicitly captures the ways in which a hand's shape can and can't vary. The specificity of the model proves to be invaluable when tracking from a single 2D image, from which data is both noisy and relatively sparse.
- The hand is modelled as a surface mesh, from which the positions of expected contours are easily derived. By sampling at every mesh vertex large amounts of good position information can be obtained, even in the case of partial occlusion or noise from background clutter.
- The technique uses linear mathematics in most calculations, which allows fast tracking rates.
- The hand posture is characterised by only a few scalars, easing gesture analysis.

The required training information is extracted semi-automatically from 3D Magnetic Resonance Images using a deformable surface mesh.

The model is used to track a hand in real-time (currently 10 frames/second on a standard 134MHz Silicon Graphics Indy workstation) using a single video camera for input. The model is projected (orthographically) onto input images and 3D edge detection is used to move and deform the model to fit image evidence.

The remainder of this paper is split into three sections. In the first the construction of the 3D PDM is described, in the second it is shown how this model is applied to hand tracking, and in the third the performance of the tracker is discussed, its shortcomings are highlighted and suggestions for improvement are made.

2 The Hand Model

2.1 Overview of Point Distribution Models

A PDM is a deformable model built from the statistical analysis of examples of the object being modelled. Given a collection of 3D training images of an object, the Cartesian coordinates of N strategically-chosen landmark points are recorded for each image. Training

example e is represented by a vector $\mathbf{x}_e = (x_{e1}, y_{e1}, z_{e1}, \dots, x_{eN}, y_{eN}, z_{eN})$.

The examples undergo least-squares alignment, and scaling to unit size; the pointwise mean shape $\bar{\mathbf{x}}$ is then calculated. Modes of variation are found using Principal Component Analysis (PCA) on the deviations of examples from the mean. These modes are represented by $3N$ orthonormal eigenvectors \mathbf{v}_j . A unit-sized object shape \mathbf{x}^U is generated by adding linear combinations of the t most significant variation vectors to the mean shape:

$$\mathbf{x}^U = \bar{\mathbf{x}} + \sum_{j=1}^t b_j \mathbf{v}_j \quad (1)$$

where b_j is the weighting for the j^{th} variation vector.

By ensuring $t \ll N$, only the important deformations are extracted, discarding training data noise, and thus object shape and variation can be captured compactly.

2.2 Acquiring Training Data

A key requirement for building such a model is the collection of landmark coordinate data from training images. Doing this manually for a 3D model is impractical due to the considerable effort required for image-model registration. Automatic mesh growing/deforming is hampered by the need for point correspondences between examples.

These setbacks can be overcome by capturing training data semi-automatically using a physically-based model. A mesh (we used a Simplex Mesh [4]) is constructed on the surface of the hand in the first training image. This mesh is deformed to fit subsequent training examples under the action of various forces. A few manually-positioned *guiding* forces pull key features (such as the fingertips) roughly into position, and 3D edge detection is used to construct forces at *every* vertex to drive the model to an exact fit. Internal forces also act to constrain the model shape (for smoothness and evenness). Full details can be found in [5].

The mesh vertices can be used directly as landmark points for the PDM.

2.3 Model Construction

Surface meshes were fitted to 8 different hand images, all from the same person. From these, a PDM with 7 modes of variation was constructed. Most of the significant deformation (over 93%) is captured by the first five modes. Figure 1 shows the two main variation modes. It should be noted that 8 training examples are too few to use as a basis for a PDM. The modes of variation produced mainly constitute linear interpolations between the different hand shapes in the training

set. For this reason, the method is somewhat similar to *key frames*, as used by Blake [6].

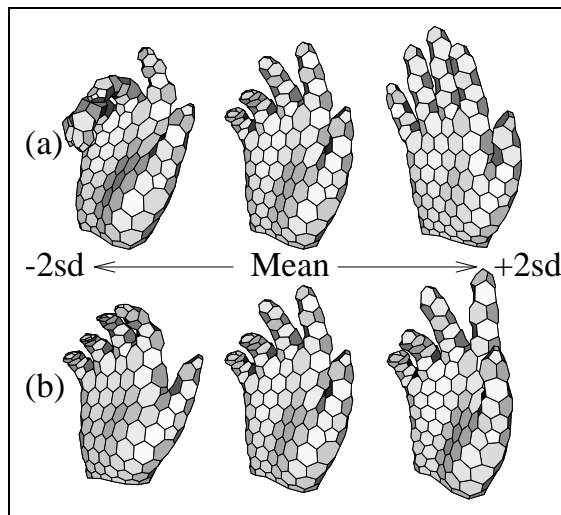


Figure 1: The first (a) and second (b) modes of variation of the 3D hand PDM.

3 Tracking

There has been much work on using PDMs for object location and tracking in both two and three dimensions. In most of this previous work, the dimensionality of the model has matched that of the input image (ie. 2D model for 2D images [7, 8, 9], 3D model for 3D images [10]). Work on matching a 3D model to a 2D image has so far assumed a ground plane constraint and only one degree of rotational freedom [11, 12]. We are attempting to match a 3D PDM to a 2D image under full 6 DOF.

The key to model-based object location is finding the set of model parameter values which cause the model to best fit the image data. In our case the parameters are a translation vector $\mathbf{u} = (u, v, w)$, a 3×3 rotation matrix \mathbf{R} , a scale factor s and the deformation parameters b_j . Iterative pose refinement is used – given a fair initial guess at an object’s location, local image information (eg. edge data) is extracted and used to calculate a small change in the model parameters which will improve the fit.

To compare the model to the image, it is necessary to project the model onto the image. The model is first deformed from the mean shape $\bar{\mathbf{x}}$ using equation (1). The deformed model \mathbf{x}^U is then rotated, scaled and translated into the *posed* model \mathbf{x}^P , such that the position \mathbf{x}_i^P of the i^{th} landmark is given by:

$$\mathbf{x}_i^P = s\mathbf{R}\mathbf{x}_i^U + \mathbf{u} \quad (2)$$

\mathbf{x}^P is currently projected into the 2D image using an orthographic projection (simply by dis-

carding the z -coordinates). This allows projections and inverse projections to be calculated quickly and, with a sufficiently distant camera, produces negligible distortion. Of course, z -position information is lost but, assuming a fixed-size object and known intrinsic camera properties, z position can be inferred from scaling (this is effectively a *scaled* orthographic projection).

As mentioned above, the idea is to find values for \mathbf{u} , s , \mathbf{R} and the b_j which give the best match between model and image. These parameters are updated iteratively based on collection of image evidence, specifically by finding the best local movement for individual model landmarks. The result is a collection of suggested landmark movements (in the form of (dx, dy) pairs) which undergo statistical voting to change the overall model pose. When used in this way, PDMs are commonly referred to as Active Shape Models (ASMs) [3].

Because the process is iterative, it extends naturally to tracking an object over a time sequence of images; the model's final position in one image is used as the starting position for the next image.

3.1 Gathering Image Evidence

The task is to find suggested movements for individual landmarks by examining image data. The evidence that can be gathered from a 2D image with respect to a 3D model is limited.

Firstly, if a hand is to be tracked unmarked, the only reliable position evidence that can be easily extracted is from edge data. No suggested movement can be made for vertices which are not on or near the model boundary in the current view (Shen observed this in his work on vehicle model building [11]). The *aperture problem* (see later) is also experienced, whereby even if an edge is found, the desired position *along* the edge is uncertain.

Secondly, because a single 2D image is being used for input, no depth information is available i.e. the z -coordinate of any discovered edge is uncertain (this is in fact another instance of the aperture problem).

The data required is a suggested movement $d\mathbf{x}_i^P$ for each model landmark i , along with an associated weighting W_i indicating how strong the evidence is for this movement (this is essential to allow for zero weightings for landmarks which provide no evidence). The method of information gathering is this: the unit normal \mathbf{n}_i to the model surface at landmark i is first found, defined as the normal to the plane containing landmark i 's three neighbours. If \mathbf{n}_i subtends an angle of less than 30° to the x - y plane, it is assumed that it is on or very near the model edge (in the current pose). This is imprecise, but is much faster than an exact boundary-finding algorithm. A line of

pixels is extracted from the image either side of landmark i and in the direction of the projection of \mathbf{n}_i into the x - y plane. The greatest intensity change (ie. edge) along this line is found and $d\mathbf{x}_i^P$ is set accordingly (its z component is set to zero). W_i is set to the magnitude of the intensity change. If \mathbf{n}_i subtends an angle of less than 30° to the x - y plane, it is assumed that it is not near an edge, and so cannot be used to gather image evidence; $d\mathbf{x}_i^P = \mathbf{0}$ and $W_i = 0$ are accordingly set.

Figure 2 shows an enlargement of the feature extraction on part of the hand. The model is shown in white and the suggested movements, where discovered, are shown as black lines. To increase speed, only every n^{th} pixel is sampled along the normal, intervals; this explains why some of the black lines do not quite meet the image edges.

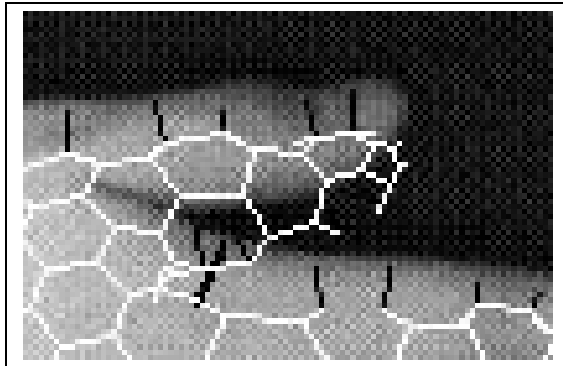


Figure 2: Suggested landmark movements.

3.2 Updating the Model Position

Given a suggested movement $d\mathbf{x}_i^P = (dx_i, dy_i, 0)$ for each landmark i , and an associated weighting W_i , the task is to update the model parameters u , v , s , \mathbf{R} and the shape parameters b_j . A weighted least-squares solution involves finding values for $\mathbf{u}' = \mathbf{u} + d\mathbf{u}$, $s' = s + ds$, $\mathbf{R}' = d\mathbf{R}\mathbf{R}$ and $b'_j = b_j + db_j$ which minimise ε in:

$$\varepsilon = \sum_{i=0}^N W_i \|\mathbf{x}_i^P + d\mathbf{x}_i - (s'\mathbf{R}'(\bar{\mathbf{x}} + \sum_{j=1}^t b'_j \mathbf{v}_j)_i + \mathbf{u}')\|^2 \quad (3)$$

The solutions for du , dv , ds , $d\mathbf{R}$ and the db_j are as follows:

$$du = \frac{\sum_{i=1}^N W_i dx_i}{\sum_{i=1}^N W_i}; \quad dv = \frac{\sum_{i=1}^N W_i dy_i}{\sum_{i=1}^N W_i} \quad (4)$$

du and dv are then used in the calculation of ds and $d\mathbf{R}$:

$$ds = \sqrt{\frac{\sum_{i=1}^N W_i \|\mathbf{x}_i + d\mathbf{x}_i^P - (\mathbf{u} + d\mathbf{u})\|^2}{\sum_{i=1}^N W_i \|\mathbf{x}_i - \mathbf{u}\|^2}} \quad (5)$$

To calculate $d\mathbf{R}$ a weighted version of Arun's singular value decomposition (SVD) method [13] is used. The 3×3 matrix H is first found.

$$H = \sum_{i=1}^N W_i (\mathbf{x}_i - \mathbf{u})(\mathbf{x}_i + d\mathbf{x}_i^P - (\mathbf{u} + d\mathbf{u}))^T \quad (6)$$

and then find the SVD of H :

$$H = U\Lambda V^T \quad (7)$$

$d\mathbf{R}$ is then given by:

$$d\mathbf{R} = VU^T \quad (8)$$

Before calculating the db_j , the effects of $d\mathbf{u}$, ds and $d\mathbf{R}$ are removed from each $d\mathbf{x}_i^P$:

$$d\mathbf{x}_i^{P'} = s' \mathbf{R}' \mathbf{x}_i^U + \mathbf{u}' - \mathbf{x}_i^P + d\mathbf{x}_i^P \quad (9)$$

The db_j are calculated independently, assuming all other quantities to be fixed. This does not give an exact solution but it avoids any matrix inversions and so is much faster. Iteration can be used to converge on the best solution, but this is not strictly necessary since the tracker is iterative over frames anyway.

$$db_j = \frac{\mathbf{v}_j^T \mathbf{W} d\mathbf{x}^{P'}}{\mathbf{v}_j^T \mathbf{W} \mathbf{v}_j} \quad (10)$$

where $\mathbf{W} = \text{diag}(W_1, W_1, W_1, \dots, W_N)$.

Although the weighted least-squares approach does find a suitable solution, it has been noted that convergence can be hampered by the *aperture problem* [14]: if an edge is found along a model normal, the landmark is encouraged towards that point. However the landmark's true resting position might be further along the edge. Also, in a 2D image, $dz = 0$ must be assumed, because there is no evidence to the contrary. The true resting position of the landmark may require $dz \neq 0$. Hill proposes a method to overcome these problems using *directional weights* [14], whereby landmarks are made free to 'slide' along target edges or across target planes.

Hill's solution involves the inversion of large weight matrices; it would be favourable to avoid this for reasons of speed. It is possible to improve on the 'simply' weighted least-squares approach without incurring too much computational cost. Directional information from the suggested landmark movements is used to determine how much the evidence from a particular landmark should contribute towards updating a particular parameter. For example, if the normal to landmark i is

parallel to the x axis, its image evidence should make no contribution in calculating dv . This tactic is put into practice as follows: the least-squares equations are as for the 'simply' weighted approach; however, in calculating the change dq to model parameter q , the weighting W_i is replaced with $W_{q,i}$, which is calculated from W_i and $d\mathbf{x}_i$ specifically with respect to parameter q . The following calculations are used:

$$W_{u,i} = W_i |du_i|; \quad W_{v,i} = W_i |dv_i| \quad (11)$$

$$W_{s,i} = \frac{W_i |d\mathbf{x}_i \cdot (\mathbf{x}_i - \mathbf{u})|}{|\mathbf{x}_i - \mathbf{u}|} \quad (12)$$

$$W_{\mathbf{R},i} = \sqrt{W_i^2 - W_{s,i}^2} \quad (13)$$

When finding the db_j , \mathbf{W} is replaced by $\mathbf{W}_D = \text{diag}(W_{x,1}, W_{y,1}, W_{z,1}, \dots, W_{z,N})$.

It is important to appreciate that the above weighting scheme does not fully encapsulate address the aperture problem; the weighting are calculated independently for each model parameter – no allowance is made for the interdependency of the parameters. However, it provides an improvement over the simply-weighted scheme at virtually no extra cost.

4 Performance Evaluation

An experimental mock-up of the tracker has been constructed using a single colour camera pointing down at a homogeneous dark surface and connected to a Silicon Graphics Indy workstation running at 134MHz. Images are captured from the camera at approximately 10Hz and the tracking algorithm is applied in real-time. Images are echoed to the workstation screen, with the hand model superimposed. To avoid the global search problem (a hand must be found before it can be tracked), the model is initialised centrally in the image and only begins tracking when a hand is moved into position 'under' it; this event is detected by the presence of strong edges at over 80% of the currently 'active' model landmarks. The user can see the model tracking their hand, this providing useful feedback. Figure 3 shows some snapshots from the experimental system.

A quantitative evaluation of the tracker has not yet been performed. A qualitative evaluation is as follows:

- Changes in x and y translation, scale and rotations in the x - y plane were tracked with no difficulty, irrespective of the hand pose.
- Rotations out of the x - y plane initially caused problems. In particular, the transition from (a) to (c) in Fig. 3 produced a decrease in scale instead of the expected rotation. This is because much of the evidence collected from the 2D image (the sides of

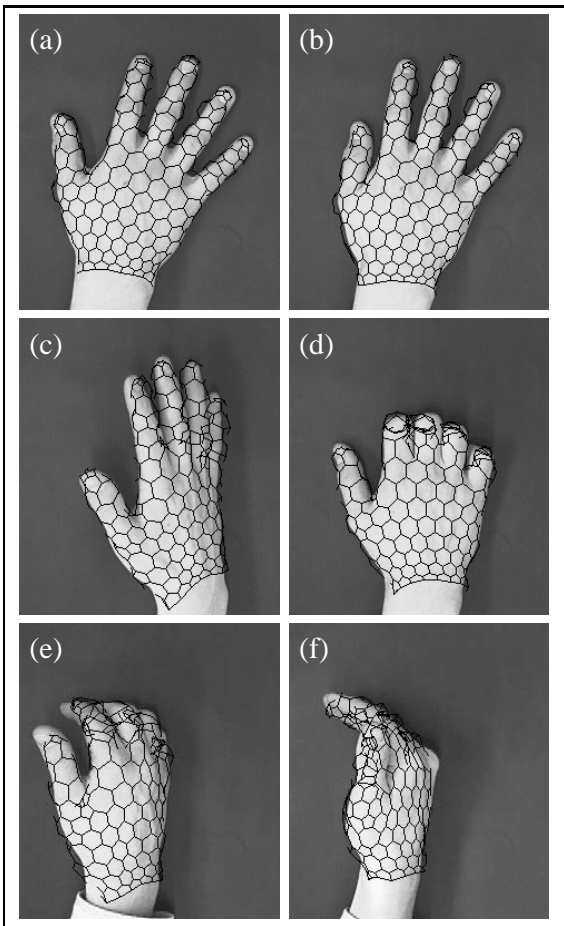


Figure 3: Hand tracking using the 3D PDM.

the hand moving inwards) is consistent with such a change, and the only evidence to the contrary comes from the static position of the fingertips (the wrist is unmarked and provides no clues). As a temporary measure the model size was fixed at a constant value. The rotations are then estimated correctly.

- Rotations out of the x - y plane were estimated better with the size constraint; however, success depended very much on the starting pose. Most problems were caused by ambiguity: because the hand is roughly planar, positive and negative rotations of the hand viewed from either a direct or sideways-on view appear very similar in an orthogonal projection (eg. the transition from (a) to (c)). Consequently the model sometimes rotated the wrong way. Also, as an object rotates in this way the visible edge is due to a different part of the hand. We rely on the change in pose being small enough, and the vertices in the model being close enough together, to minimise the effects of this.

- Clearly visible deformations were tracked well; for example, the transition from (a) to (b). Self-occluded deformations were tracked less well, since there is little image evidence to support them. An example is the transition from (a) to (d), which was always tracked accurately, but more slowly than visible deformations.

- Self-occlusions also caused other problems. Our tracking program cannot (yet) handle occlusion, and occluded vertices therefore tend to be ‘attracted’ to the nearest *visible* edge. This occurred in poses such as (e) and (f). The effect is counteracted to some extent by the model shape constraints, however, the use of a *linear* PDM to model essentially non-linear deformations means that implausible model shapes can occur.

These results are roughly as we expected for our first attempt at the problem. There are obviously a few issues to be addressed, namely:

- scale/rotation confusion
- planar rotation ambiguities
- occlusions
- implausible model shapes due to linear model

The improvements we plan to make to the system are as follows:

- Address the handling of occlusions. Previous work on this (due to Rehag [15]) has made use of *layered templates* to model occlusion. We hope to adopt a simpler method whereby we determine the visibility of each vertex individually by considering whether any model facets lie in front of it.
- Use a non-linear modelling technique to improve the accuracy and specificity of the hand model, thus improving tracking. We have already developed one extension to the PDM which allows for a better modelling of pivotal motion [16]. Initial experiments using this model for 3D tracking are inconclusive at present; there may be inherent instability problems.
- Improve the model’s mesh configuration in some way. At present, the distribution of vertices over the model surface is roughly uniform. However, it is apparent that some parts of the hand (eg. the fingertips) provide more important information than others. To reflect this, we have two possible schemes in mind: the first is simply to manually increase the concentration of vertices in such areas of the model; the second is to automatically detect which areas are ‘important’ and assign a higher weight to evidence collected from them. Cues for importance of

a vertex might be the local surface curvature or, more likely, *the amount of movement of the vertex over the training set* – it is these vertices whose position is most sought.

The system has also been tested against a cluttered background. Figure 4 shows an example. Performance was almost as good as for the controlled background, and it is hoped that attention to the above matters will improve it further still.

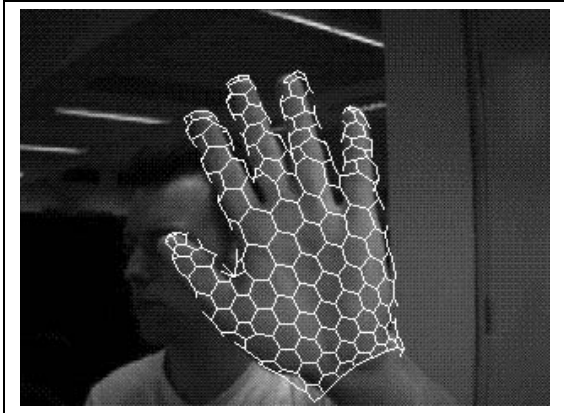


Figure 4: Tracking against a cluttered background.

5 Conclusions

A description has been given of a first attempt at tracking a non-marked human hand in real time from a single camera, using a deformable model (a Point Distribution Model, or PDM) of the hand built from training examples.

It has been shown how information can be extracted from a 2D image to move and deform a 3D model; the instances where this is most and least successful have been pointed out.

The main strength of this approach is the use of the PDM, which is, or has the potential to be, a very compact and accurate model for the range of legal hand shapes, providing good contour information. It also lends itself to simple, fast processing.

In its current form, the greatest failing is due to occlusion; this problem will be addressed in future work.

References

- [1] B. Dorner. Hand shape identification and tracking for sign language interpretation. Looking at People Workshop, Chambery, France, 1993.
- [2] J.M. Rehg and T. Kanade. Visual tracking of high DOF articulated structures: An application to human hand tracking. In *Proc. 3rd ECCV*, volume II, pages 35–45, Stockholm, Sweden, 1994. Springer-Verlag.
- [3] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham. Active Shape Models - their training and applications. *Computer Vision and Image Understanding*, 61(2), January 1995.
- [4] H. Delingette. Simplex Meshes: a general representation for 3D shape reconstruction. Technical Report 2214, INRIA, 1994.
- [5] A.J. Heap and D.C. Hogg. 3D deformable hand models. Gesture Workshop, York, UK., 1996.
- [6] A. Blake and M.A. Isard. 3d position, attitude and shape input using video tracking of hands and lips. In *Proc. ACM Siggraph*, pages 185–192, 1994.
- [7] A. Lanitis, C.J. Taylor, and T.F. Cootes. A generic system for classifying variable objects using flexible template matching. In *Proc. BMVC*, pages 329–338, Guildford, UK, 1993. BMVA Press.
- [8] A. Baumberg and D. Hogg. An efficient method for contour tracking using active shape models. In IEEE Computer Society Press, editor, *IEEE Workshop on Motion of Non-rigid and Articulated Objects*, pages 194–199, November 1994. Also available as ftp://agora.leeds.ac.uk/scs/doc/reports/1994/94_11.ps.Z.
- [9] A.J. Heap. Real-time hand tracking and gesture recognition using Smart Snakes. In *Proc. Interface to Human and Virtual Worlds*, Montpellier, France, June 1995. Also available as ftp://agora.leeds.ac.uk/scs/doc/reports/1995/95_5.ps.Z.
- [10] A. Hill, A. Thornham, and C.J. Taylor. Model-based interpretation of 3D medical images. In *Proc. BMVC*, pages 339–348, Guildford, UK, 1993. BMVA Press.
- [11] X. Shen and D.C. Hogg. Generic 3D shape model: Acquisitions and applications. In *Proc. CAIP*, Prague, Czech Republic, September 1995.
- [12] A.D. Worrall, J.D. Ferryman, G.D. Sullivan, and K.D. Baker. Pose and structure recovery using active models. In *Proc. BMVC*, Birmingham, UK, September 1995.
- [13] K.S. Arun, T.S. Huang, and S.D. Blostein. Least-squares fitting of two 3d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 89(5):698–700, 1987.
- [14] A. Hill and C.J. Taylor. Active shape models and the shape approximation problem. In *Proc. BMVC*, pages 157–166, Birmingham, UK, 1995. BMVA Press.
- [15] J.M. Rehg and T. Kanade. Visual tracking of self-occluding articulated objects. In *Proc. ICCV*, Boston, MA., 1995.
- [16] A.J. Heap and D.C. Hogg. Extending the Point Distribution Model using polar coordinates. In *Proc. CAIP*, Prague, Czech Republic, September 1995. Also available as ftp://agora.leeds.ac.uk/scs/doc/reports/1995/95_25.ps.Z.