A Dynamic Human Model using Hybrid 2D-3D Representations in Hierarchical PCA Space

Eng-Jon Ong and Shaogang Gong

Department of Computer Science, Queen Mary and Westfield College,
London E1 4NS, UK
{ongej|sgg}@dcs.qmw.ac.uk

Abstract

We propose a novel framework for a hybrid 2D-3D dynamic human model with which robust matching and tracking of a 3D skeleton model of a human body among multiple views can be performed. We describe a method that measures the image ambiguity at each view. The 3D skeleton model and the correspondence between the model and its 2D images are learnt using hierarchical principal component analysis. Tracking in individual views is performed based on CONDENSATION.

1 Introduction

The ability to track a person's body parts is important in human gesture recognition and behaviour analysis. To this end, we adopt a model which utilises the underlying 3D structure of a person's body skeleton. The parameters of this model include the position of its 3D vertices. Additional parameters such as relative angles between bones can also be introduced into the model. Methods utilising 3D models for similar reasons include the use of inverse kinematics for estimating the rotation parameters [9]. Alternative methods were proposed to extract the bones using primitive solids such as superquadrics [2, 8] or cylinders [5]. A search procedure together with a fitness measurement can be used to compute the parameters of a 3D skeleton [2, 5]. Another approach is to explicitly learn and take advantage of the correlation between the 2D measurements and its corresponding 3D models. This has been attempted by Bowden *et al.* [1] where different 2D cues are fused with the 3D model and learnt using hierarchical PCA. This is the approach we have also adopted for single view mode match and tracking. We further extend this approach in Section 2 to the use of CONDENSATION algorithm to allow for a more robust and general solution.

Self occlusions and ambiguities in depth makes tracking 3D models difficult. To tackle some of these problems, we adopt a multi-view approach. To address the problem of integrating information from different views, one solution is to determine different visible body parts for view selection during tracking [7]. Alternatively we propose a method for integrating information from multiple views. This enables us to estimate the degree of ambiguity in the 2D cues to be used for tracking the 3D model. It also allows us to quantify the ambiguity of a given view among multiple available views. Furthermore, we extend this method to measure the ambiguity of individual parts of an extracted 3D skeleton in each individual view. This allows us to better fuse information from different

BMVC 1999 doi:10.5244/C.13.4

views. Details on model tracking through multiple views are described in Section 3. We then present our experiments in Section 4 before conclude in Section 5.

2 Learning 3D Skeleton Models for Tracking

A 3D skeleton of a human body can be defined as a collection of 3D vertices together with a hierarchical arrangement of bones. The bones are used to link two 3D vertices and constrain their possible movements. We start with an example-based Point Distribution Model (PDM) commonly used for modelling and tracking 2D shapes. This approach was extended to track 3D skeletons whereby a hybrid 2D-3D representation of a human model using PDM was introduced [1]. Here we define a state vector representing an instance of a similar hybrid model consisting of observable 2D data and its corresponding 3D skeleton. The observable data are 2D features, including the 2D shape of a person's body and the positions of some body parts, which can be directly measured from the image. The shape is represented by the contour of a person's silhouette consisting of N_C number of 2D vertices, $\mathbf{v}_S = (x_1, y_1, ..., x_{N_C}, y_{N_C})$. The body parts consist of positions of the left hand (x_L, y_L), the right hand (x_R, y_R) and the head (x_H, y_H). The head is used as an alignment point for both the body parts and the contour. The positions of the remaining body parts (left and right hands) are concatenated into a vector, $\mathbf{v}_C$. Finally, the corresponding 3D data, which consists of N_T number of 3D vertices of a skeleton, is similarly concatenated into a vector ($\mathbf{v}_T$). A state vector, $\mathbf{v} = (\mathbf{v}_S, \mathbf{v}_C, \mathbf{v}_T)$, therefore consists of a hybrid concatenation of three 2D-3D components: 2D shape contour, 2D body parts positions and their 3D skeleton vertices (see Figure 1).

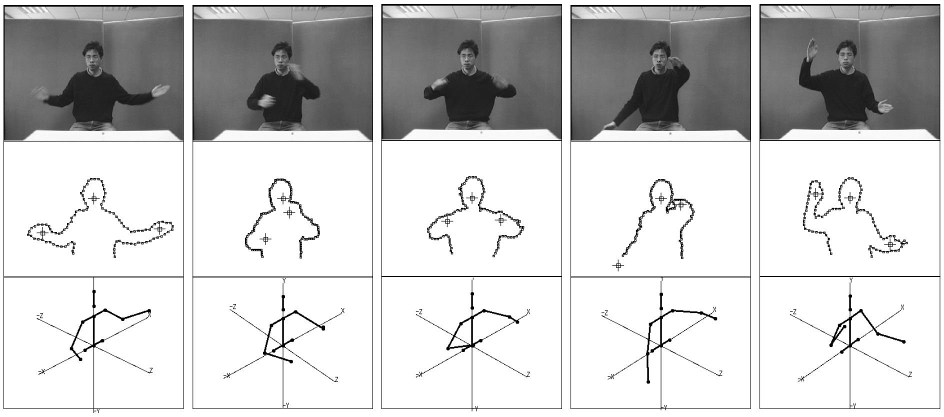

Figure 1: Different instances of the state vector is illustrated here. The top row corresponds to the input images. The middle row corresponds to the contours ($\mathbf{v}_S$) and body parts positions ($\mathbf{v}_C$). The bottom row shows the corresponding skeleton ($\mathbf{v}_T$).

Learning

The state space of this hybrid PDM model is inherently nonlinear. This is due to the articulated nature of the skeleton model together with the restriction that body segments are rigid. The space reachable by some parts of the skeleton model is nonlinear. In addition, physical constraints further limit the possible positions of the vertices. For example, we cannot move our hands through our bodies.

Learning the skeleton models would require some means of capturing this nonlinear space. One method which can be used to effectively represent high-dimensional nonlinear spaces is the Hierarchical Principal Component Analysis (PCA) [4]. This method consists of two main steps: (1) Dimensionality reduction using PCA is first performed on all the training examples to remove redundant dimensions. A dimensionality-reduced input space is then given by a global eigenspace into which all training examples are projected. Since the projection to the global eigenspace is linear, the resulting space occupied by the projected training examples remains nonlinear. (2) This nonlinear space occupied by the dimensionally reduced training examples is further captured using a number (N_{clust}) of clusters, $\{\mathbf{c_1}, ..., \mathbf{c}_{N_{clust}}\}$, in the global eigenspace. Each cluster $(\mathbf{c_i})$ is defined as $\{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, \mathbf{P}_i, \boldsymbol{\Lambda}_i\}$, where the mean vector $(\boldsymbol{\mu}_i)$ denotes the cluster's location, and the covariance matrix $(\boldsymbol{\Sigma}_i)$ represents the cluster's orientation and size. The eigenvectors of $\boldsymbol{\Sigma}_i$ is kept in $\mathbf{P}_i$ and the eigenvalues kept in the diagonal elements of a diagonal matrix $(\boldsymbol{\Lambda}_i)$. Since each cluster is represented by its principal components, we refer a set of such clusters as the set of localised principal components (see Figure 2).

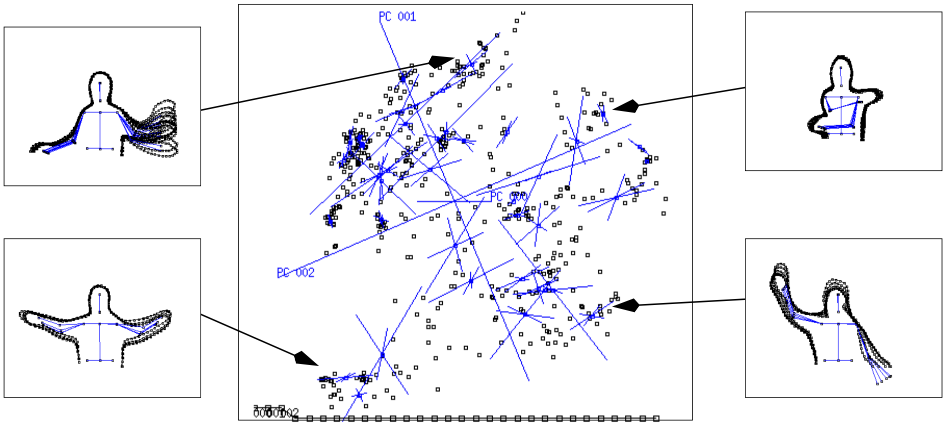

Figure 2: This diagram illustrates the projections of state vectors to the 3 largest global eigenvectors. Training example vectors are shown with the local principal components of different clusters. The side figures show the reconstruction of moving points along the largest local principal components from the mean of a cluster.

Modelling Discontinuous Dynamics

Having learnt the distribution of the training examples, we now proceed to track the state vectors. To make use of the learnt distribution, tracking is performed in the global eigenspace. The original state vector can be reconstructed by taking the linear combination of the global eigenvectors. The coefficients for the linear combination is given by the components of the tracked vectors in the global eigenspace.

The possibility of discontinuous 2D shape contours occurring between views needs to be addressed. In certain views, some feature points of a 2D shape contour have no correspondences to previous views. This can be caused by small 3D movements resulting large changes in the state vector (see Figure 3). Such phenomena in turn can cause a

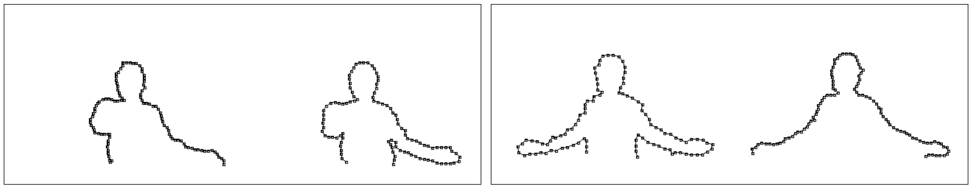

Figure 3: Small 3D body movements can sometimes cause huge discontinuities in the feature points of its corresponding 2D shape contour.

huge jump in the projections of state vectors into the global eigenspace. To address this problem we adopt the approach of modelling a matrix of transition probabilities between different subspaces in a global eigenspace [3] . The subspaces are represented by clusters in the global eigenspace.

CONDENSATION for Tracking

To estimate the 3D skeleton model, prediction on state transition must cope with its temporal, non-deterministic and discontinuous nature. The CONDENSATION [6] algorithm is adopted for this task. The stochastic nature of this algorithm allows discontinuous changes to be handled by modelling such effects into the dynamics of propagating multiple sample states therefore enabling the tracker to recover from failure. The ability to have multiple hypotheses also provides the potential for coping with ambiguities in the estimated skeleton in certain views. Although this does not resolve ambiguities directly, tracking is made more robust when the problem arises. The CONDENSATION algorithm is used to track a population of projected state vectors as a probability density function in the global eigenspace, as illustrated in Figure 4. The sample prediction step is modified to make use of the transitional probability matrix. This allows the tracker to propagate samples across different subspaces, therefore coping with the discontinuous nature of the data.

The estimated state vector ($\mathbf{v}$) is reconstructed from the components of a given sample in the same manner, by taking a linear combination of N_{gev} number of global eigenvectors ($\mathbf{g}_1, ..., \mathbf{g}_{N_{gev}}$):

$$\mathbf{v} = \sum_{i=1}^{N_{gev}} s_i \mathbf{g}_i \tag{1}$$

In the reconstructed vector $\mathbf{v} = (\mathbf{v}_S, \mathbf{v}_C, \mathbf{v}_T)$, the accuracy of both the shape contour ($\mathbf{v}_S$) and the body parts ($\mathbf{v}_C$) are measured individually before combined to yield a final fitness value. More precisely, a prediction accuracy value (f_S) for the contour can be computed as follows: (1) Assign the prediction accuracy value, f_S = 0. (2) For N_C number of vertices of a contour: (a) Find the distance (s) from each 2D shape vertex position to the pixel of greatest intensity gradient by searching along its normal, (b) compute $f_S = f_S + s$.

We now consider how to measure the accuracy of the state vector's predictions for the body parts' positions, (x_{p1}, y_{p1}) and (x_{p2}, y_{p2}). In each frame, three skin-coloured regions of interests are tracked corresponding to positions of the hands and the face.

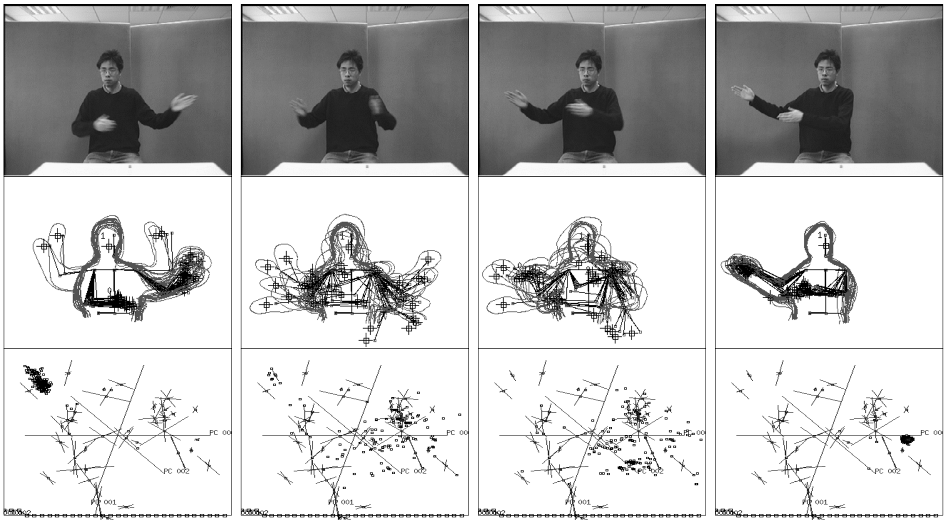

Figure 4: The tracking of a population of samples by CONDENSATION is illustrated here. The bottom row shows the global eigenspace with clusters of local principal components. The samples are highlighted by small dots. The middle row shows reconstructed state vectors using 10 out of 100 samples propagated over time. The top row are example input images.

Two of these three positions are taken and ordered into a four-dimensional vector $\mathbf{m}_b = (x_{m1}, y_{m1}, x_{m2}, y_{m2})$ where (x_{m1}, y_{m1}) and (x_{m2}, y_{m2}) are the coordinates of the first and second position respectively. We define the accuracy of the predictions for the body parts' positions as:

$$f_C = \sqrt{(x_{p1} - x_{m1})^2 + (y_{p1} - y_{m1})^2} + \sqrt{(x_{p2} - x_{m2})^2 + (y_{p2} - y_{m2})^2} \tag{2}$$

A final fitness value (f_n) for $\mathbf{v}$ of the n^{th} sample, $(\mathbf{s}_n^{(t+1)})$, is then given by the individual fitness measurements as follows:

$$f_n = O \exp(\frac{-f_C}{2P}) + R \exp(\frac{-f_S}{2Q})$$

where O and R are scale constants used to even out differences in scale between the two weighted fitness measurements, f_C and f_S. Constants P and Q represents the amount of variance or tolerance allowed between the predicted and observed values for the body parts and the shape contour respectively.

3 Tracking from Multiple Views

Multiple views can disambiguate situations under which self-occlusions occur. Here we describe a method for estimating the degree of ambiguities in 2D image measurements at a given view. When applied to different views, this measurement allows us to select the least ambiguous information from one of the available views.

Modelling the Ambiguity of a View

We describe a method to measure the degree of ambiguities in the observations of a state vector. Observations are ambiguous when more than one projections of a 3D skeleton can be matched. Figure 5 illustrates an example of this phenomenon. We consider that a state vector's observation information is ambiguous if there are many other state vectors which have *similar* observations but *dis-similar* underlying 3D skeleton models.

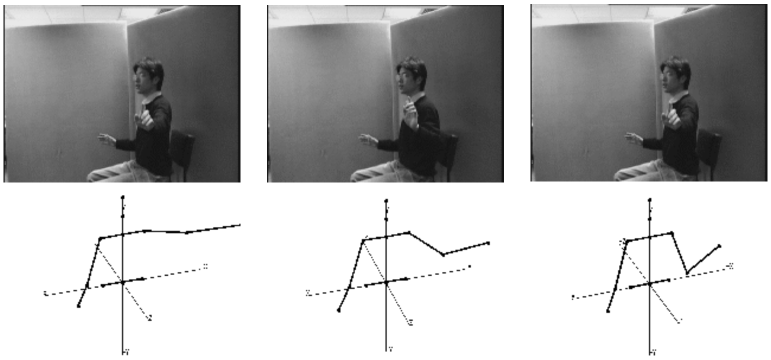

Figure 5: An example illustrates the ambiguous nature of 2D observations in certain views. Despite that the 3D skeleton is changing, there is little difference in the 2D shape contour and the positions of the hands in the image.

Let us define an *observation subspace* of a state vector $(\mathbf{v}_C, \mathbf{v}_S, \mathbf{v}_T)$ to be $\mathbf{v}_S$ and $\mathbf{v}_C$. There are N_C number of vertices for a shape contour $(\mathbf{v}_S)$ while the number of tracked body parts positions is two. Thus, the observation subspace spans from dimension 0 through dimension $2N_C + 4$ in the state vector. We also define a *skeleton subspace* to be the dimensions of a state vector for all the 3D skeleton vertices. There are N_T number of 3D vertices for a 3D skeleton. To measure how close the 2D observations in two state vectors are, we define an *observation-distance* (d_{ob}) between two state vectors **x** and **y** as:

$$d_{ob}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{2N_C+4} \sqrt{dx_{ob}^2 + dy_{ob}^2} \tag{3}$$

where $dx_{ob} = x_{2i} - y_{2i}$ and $dy_{ob} = x_{2i+1} - y_{2i+1}$. Similarly, to measure how similar the 3D skeleton component in two state vectors are, we define a *skeleton-distance* (d_s) between two state vectors **x** and **y** as:

$$d_s(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{3N_T} \sqrt{dx_s^2 + dy_s^2 + dz_s^2} \tag{4}$$

where $dx_s = x_{3i} - y_{3i}$, $dy_s = x_{3i+1} - y_{3i+1}$ and $dz_s = x_{3i+2} - y_{3i+2}$. Based on a set of K number of example state vectors of the 2D-3D hybrid representation, $\{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_K\}$, we now define the ambiguity of an example state vector as:

$$a \quad = \quad \sum_{j=1}^{K} G(s_{2D}, \sigma) f(s_{3D}, \beta, t) \tag{5}$$

where $s_{2D} = d_o(\mathbf{w}, \mathbf{y}_j)$ and $s_{3D} = d_s(\mathbf{w}, \mathbf{y}_j)$. Equation (5) formulates our notion of *ambiguity between a 3D skeleton and its 2D observations in a single view*. If two example state vectors have very similar observations, we would like to consider the difference in their 3D skeletons. This is the role of the one-dimensional Gaussian kernel $G(x, \sigma)$. It returns the similarity between observations of two example state vectors. The smaller the observation-distance, the greater the similarity. The rate at which this value reduces as the observation-distance increases is determined by the standard deviation (σ) of the Gaussian kernel.

If they both have similar observations and 3D skeletons, there is little or no difference in them, the ambiguity ratio should not be increased. A translated Sigmoid function is then used

$$f(x, \beta, t) = \frac{1}{1 + \exp(-\beta(x - t))} \tag{6}$$

Its role is to weight down contributions of example state vectors whose ambiguity is currently measured. Precisely how dis-similar two skeletons must be before they are considered to cause ambiguities is determined by t and β, the sigmoid's mid-point and scaling parameter respectively.

Fusion of Model States between Views

In order to make use of the ambiguity measurement, a set of K number of example state vectors , $\{\mathbf{y}_1, ..., \mathbf{y}_K\}$ of the 2D-3D hybrid representation is provided. The ambiguity, a_i for each example, $\mathbf{y}_i$ is computed using Equation (5) resulting in the set $\{a_1, .., a_K\}$. Given a novel state vector, $\mathbf{n}$, its ambiguity can be calculated by assigning it the ambiguity value of its nearest neighbour:

$$a_{\mathbf{n}} = a_j \mid \quad d_e(\mathbf{n}, \mathbf{y}_i) < d_e(\mathbf{n}, \mathbf{y}_j) \quad \forall i, i = 1, ..., j-1, j+1, ..., K \tag{7}$$

This is applied to tracking using multiple views. For each individual view, the 3D skeleton is tracked using the CONDENSATION algorithm as described in Section 2. We select M samples with the highest fitness values given by Equation (2). The model state vector ($\mathbf{v}$) is reconstructed from each sample using Equation (1). Its ambiguity is calculated using Equation (7). We select the sample which has the highest product of its fitness value given by CONDENSATION together with its ambiguity value.

4 Experiments

First, our single view based skeleton model was used to track upper torso of people performing a series of different gestures. The 2D shape contour was represented using 100 image feature points. The 3D skeleton consisted of 12 vertices for the upper torso. The hands and head were tracked. However, only the positions of the hands were used in the state vector. This resulted in a state vector of 240 dimensions. The training data was obtained from sixteen continuous sequences. In each sequence, a subject was requested to perform random gestures. It was found that 30 eigenvectors accounted for roughly 80 percent of the distribution in the eigenspace (Figure 6). This number was found to be sufficient for tracking. A total of 40 clusters were used to approximate the space occupied by the projected training examples as seen in Figure 2.

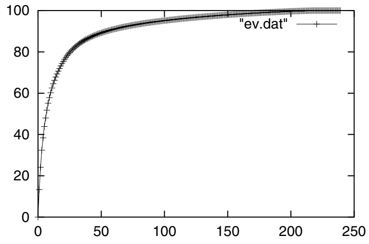

Figure 6: The weight distribution of eigenvectors in the global eigenspace used to represent the hybrid 2D-3D state vectors.

Preliminary results from the single view tracker is shown in Figure 8. Due to the lack of sufficient training examples, the transitional probability matrix built was not an accurate representation of its real values. This has slowed down the transition of the samples across different clusters. However, it was found that iterating the CONDENSATION process for a number of times over the same frame (5 times was sufficient for our experiments) allowed the samples to converge on the correct subspace. It was also found that 200 samples was sufficient to allow for a fairly accurate tracking of the 3D skeleton.

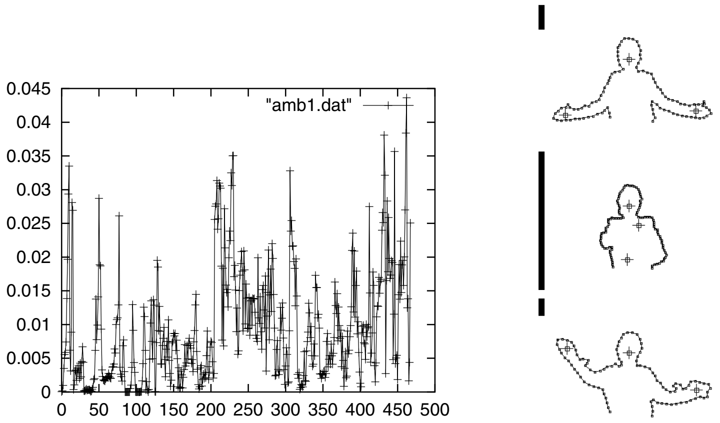

Figure 7: The ambiguity graph is shown on the left. Whilst the x-axis of the graph shows the frame index of an example sequence, the y-axis shows the ambiguity value for each frame. A few example frames of the shape contours are also shown on the right. Note that the length of the bars at the top-left corner in each shape image indicates the degree of ambiguity of that shape contour in the given frame.

The ambiguity measurements were tested on a single view for evaluation. The results can be seen in Figure 7. It was found that examples which have the arms crossed tended to have greater ambiguities than those in which the arms were stretched out. This is intuitively correct since the body shape contours are more similar in views where the arms are close to the body than when the arms are stretched out.

5 Conclusions

In this paper, we have proposed a framework in which dynamic human bodies are modelled by hybrid 2D-3D representations in both a global and localised hierarchical PCA spaces. The models are tracked using CONDENSATION where the distribution prior are learnt from examples.

Training examples were obtained from real image sequences, thus removing the need for synthesising unrealistic images of humans at various poses. These examples were learnt using the hierarchical PCA approach. The method implicitly captures complex dynamics of human body movement, foregoing the need to tackle the kinematics of a human body explicitly. Additionally, it serves as a mechanism whereby novel views can be learnt incrementally. As a result, it improves the robustness of tracking.

The use of the CONDENSATION algorithm has allowed us to track nonlinear and discontinuous distributions of temporal structures within the hierarchical PCA space. More importantly, it provides us with an ability to recover from momentarily loosing track of a body when occurs in a given view. Finally, we have proposed a new method for which ambiguities in a single view can be evaluated and constrained through multiple views.

References

[1] R. Bowden, T. Mitchell, and M. Sarhadi. Reconstructing 3d pose and motion from a single camera view. In *BMVC*, pages 904–913, Southhampton, 1998.

[2] D. Gavrila and L. Davis. Towards 3-d model based tracking and recognition of human movement:a multi-view approach. In *FG'95*, Zurich, 1995.

[3] T. Heap. *Learning Deformable Shape Models for Object Tracking*. PhD thesis, School of Computer Studies, University of Leeds, UK, September 1997.

[4] T. Heap and D. Hogg. Improving specificity in pdms using a hierarchical approach. In *BMVC*, pages 80–89, Essex, UK, September 1997.

[5] D. Hogg. Model based vision: A program to see a walking person. *Image Vision Computing*, 1(1):5–20, 1983.

[6] M. Isard and A. Blake. Condensation - conditional density propogation for visual tracking. *Int. J. Computer Vision*, 1998.

[7] I. Kakadiaris and D. Metaxas. Model-based estimation of 3d human motion with occlusion based on active multi-viewpoint selection. In *CVPR*, San Francisco, June 1996.

[8] A. Pentland. Automatic extraction of deformable models. *Int. J. Computer Vision*, 4:107–126, 1990.

[9] J.M. Regh. *Visual Analysis of High DOF Articulated Objects with Application to Hand Tracking*. PhD thesis, Carnegie Mellon University, April 1995.

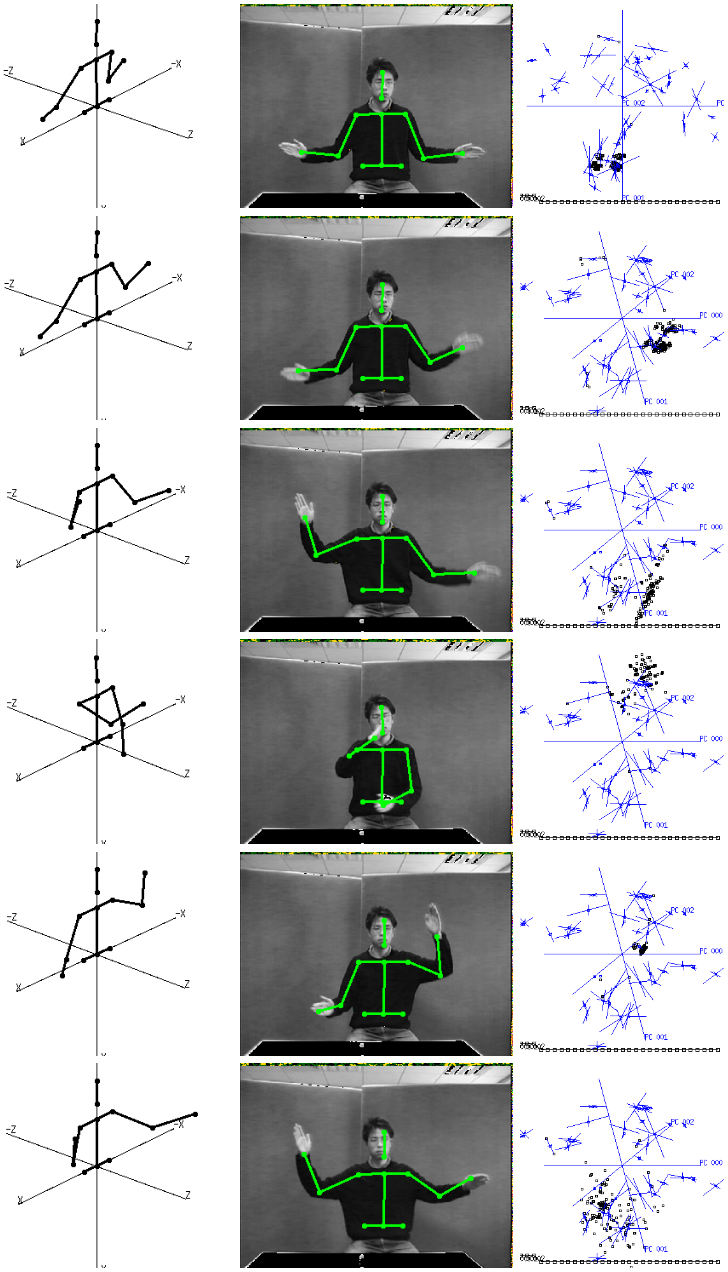

Figure 8: Single view tracking using the CONDENSATION algorithm. Every 10th frame is shown. For each frame, the left window shows the 3D skeleton, the middle window shows the 3D skeleton projected onto the image and the right window shows the global eigenspace together with the localised principal components and the samples tracked using CONDENSATION.