ELSEVIER

Image and Vision Computing 20 (2002) 691–700

image AND vision COMPUTING

www.elsevier.com/locate/imavis

Tracking people in three dimensions using a hierarchical model of dynamics

I.A. Karaulova[a,*], P.M. Hall[b], A.D. Marshall[a]

[a]*Department of Computer Science, University of Cardiff, Cardiff CF24 3XF, UK*
[b]*Department of Computer Science, University of Bath, Bath, UK*

Received 10 June 2001; received in revised form 4 February 2002; accepted 14 March 2002

Abstract

We propose a novel hierarchical model of human dynamics for view independent tracking of a human figure in monocular video sequences. The model is trained using real data from a collection of people. The top of the hierarchy contains information about the whole body. The lower levels of the hierarchy contain more detailed information about possible poses of some subpart of the body. In this article we describe our model and present experiments that show we can recover 3D human figures from 2D images in a view independent manner, and also track people the system has not been trained on. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Tracking people; 30 pose recovery; Humany dynamics; Non-linear pca

1. Introduction

This paper introduces a novel model of human dynamics that allows view independent tracking of a 3D human figure in monocular video sequences. The model represents the body dynamics of a collection of people. It is trained on real life examples using a Gaussian Mixture Model (GMM) to encode geometry and kinematics, and a Hidden Markov Model (HMM) to encode dynamics. The model can be trained on either 2D or 3D data. It allows us to recover 3D human figures from 2D image sequences, to track unknown people, and improve tracking accuracy.

Tracking humans in video has applications in many areas including surveillance, computer games, films, and biodynamics. There is a large body of work related to tracking human motion in 3D. We are interested in general methods that allow one to track the whole body, rather than in specialised trackers for face, hands, etc. [12]. Encouraging results have been achieved in tracking whole body human motion in 3D using multiple cameras [1,5,13]. We are, however, interested in recovering 3D human motion from only one view.

To recover a 3D human pose on the basis of 2D data we need to know how the 3D human figure and 2D data are correlated. Goncalves et al. [4] utilised the correlation between the real human arm size and the size of the arm in the image in order to recover its 3D positions. This approach is, however, limited only to a person whose arm geometry was used in the system. In our proposed system, this limitation was overcome by embedding into the system the dynamics and geometry of several people and thus making it more general. Bowden et al. [2] encapsulated the correlation between 2D image data and 3D human body pose in the hybrid 2D–3D model trained on real life examples. The model they used allows 3D inference from 2D data, but their method does not generalise easily to new camera positions, because the 2D part of their model is not invariant to viewpoint.

Another useful feature when tracking objects in video sequences is a model of the object dynamics. Goncalves et al. [4] used a Kalman filter for arm tracking, which is a very general mechanism and does not describe the way people move. In recent years HMMs have been applied to human behaviour prediction and recognition [16,18], and are becoming recognised a valuable mechanism for modelling human motion from real life data. We combine the use of HMM with the condensation algorithm [11] to track model states, as do Ong and Gong [14].

Hogg [6] created a view invariant model of a human body. When tracking, it was projected onto the frames in the video sequence to choose the best fitting pose of the model. We act likewise. For Hogg, the space of valid poses was

* Corresponding author.
E-mail addresses: j.a.karaulova@cs.cf.ac.uk (I.A. Karaulova), pmh@cs.bath.ac.uk (P.M. Hall), dave@cs.cf.ac.uk (A.D. Marshall).

PII: S0262-8856(02)00059-8

'hard-wired' into the model, rather than learnt from examples and there was no model of the dynamics of a human body. We build on Hogg's work by learning the valid poses from examples, and using HMMs for a description of dynamics.

We describe the structure of our hierarchical model of dynamics in Section 2, look in detail at the tracking process in Section 3, present our 2D and 3D experiments in Sections 4 and 5 and conclude in Section 6.

2. Hierarchical model of human dynamics

A natural and common way to represent the human body is with connected parts. For example, a lower limb is connected to an upper limb, which in turn connects to the torso. Such models are often used in computer graphics [19]. However, our model is based on 'part-of' relationships. For example, a lower limb and an upper limb are parts of a whole limb, which in turn is a part of a whole body. We use a part-of decomposition because, as we explain below, our model of a collection of people comprises a hierarchy of eigenspaces in which a 'high-level' eigenspace contains the major components of a 'lower-level' eigenspace. As we explain below, these eigenspaces are used to specify valid *poses* for a collection of people performing a particular action (such as walking or jumping). We regard the transition from one pose to the next as the *dynamics* of the action, and encode this using HMMs. We train our model, both poses and dynamics, from real data. Next we describe the model of valid poses, and then move on to describe the HMMs for dynamics.

2.1. A model of valid poses

It is convenient to begin the description of our model by considering a model of an individual person in a particular pose (as in Fig. 1), and use this to develop the content in the root node of our hierarchy.

We mark three-dimensional (3D) vertices, $\mathbf{x} \in \mathscr{R}^3$ at well-defined locations, such as the knee and elbow. Over the whole body there are N such vertices, which we collect into a vector $\mathbf{p} \in \mathscr{R}^{3N}$. This vector encodes the geometry of the body. As the individual performs an action the vector varies in time and hence is a continuous function $\mathbf{p}(t) \in \mathscr{R}^{3N}$. We sample it at M points in time (typically in each frame of a sequence) to obtain a discrete set of poses $\{\mathbf{p}_t\}$. This encodes kinematics.

We wish to model the poses (and, later, dynamics) for a collection of K individuals, and so must represent the collection $\{\mathbf{p}_{i,t}\}$, where the subscript i refers to a particular individual. This set samples the distribution of valid skeleton poses and can be represented by a $(3N \times MK)$ matrix, $\mathbf{P}$. It is captured from real data using a variety of vision systems (typically, in our experiments we capture between 200 and 1000 vectors). This distribution is highly non-linear, due to geometrical and physical constraints on the valid positions of vertices, therefore, we model this distribution with a GMM in reduced dimensionality space.

Similar approaches to model non-linear distributions were used by Heap et al. [9], and Bowden et al. [2] and later utilised by Ong and Gong [14] for learning the state space of their model. Our approach consists of the following steps:

1. Remove dimensions representing small variations in pose by standard PCA, so that the distribution of $\{\mathbf{p}_{i,t}\}$ is represented by the eigenspace model (eigenmodel)

   $$(\bar{\mathbf{p}}, \mathbf{U}, \mathbf{\Lambda}, MK)$$

   in which $\bar{\mathbf{p}}$ is the mean of the set, $\mathbf{U}$ is a $(3N \times s)$ matrix of eigenvectors, $\mathbf{\Lambda}$ are the eigenvalues, and MK is the set cardinality; note $s \leq \min(3N, MK)$
2. Projecting the original data set into this eigenspace to acquire dimensionality-reduced samples,

   $$\mathbf{r}_{i,t} = \mathbf{U}^{\mathrm{T}}(\mathbf{p}_{i,t} - \bar{\mathbf{p}})$$

3. Cluster the projected data into a number of Gaussian distributions using Expectation Maximisation, each cluster represented by its mean and covariance matrix, thus creating a GMM. Each cluster, q_k can also be represented by an eigenmodel

   $$q_k = (\bar{\mathbf{r}}_k, \mathbf{V}_k, \sigma_k, N_k)$$

PCA is often used to constrain variations, and representing reduced-dimensionality set with a number of clusters improves the specificity [2,9] of this and better models any non-linearities in the system.

Thus far we have considered only the root node of our model. As mentioned, nodes below the root correspond to major body parts, such as the arm, as in Fig. 1. The pose of such a part can be represented as a vector, the elements of which come from the whole body vector. Thus the domain of the part pose is the landmark points that compose this part. Consequently, a collection of poses for a part (the collection ranging over time and individuals) can be treated in exactly the same way as the set of poses for the whole body that is modelled as a GMM.

Our model comprises a hierarchy of nodes, with a hierarchy of eigenspaces in each node (Fig. 1). When we refer to "the eigenspace of a node" we mean the root eigenspace in the hierarchy of eigenspaces at that node. The eigenspaces of nodes at lower levels are partially contained within those eigenspaces of nodes at higher levels (since they are estimated from the part of the data used to estimate the eigenspaces at the higher levels), thus forming a dependency. Eigenspaces in nodes at the same level are independent (orthogonal). This representation is advantageous: because of dimensionality reduction the eigenspaces in nodes at the higher-levels encode only the major

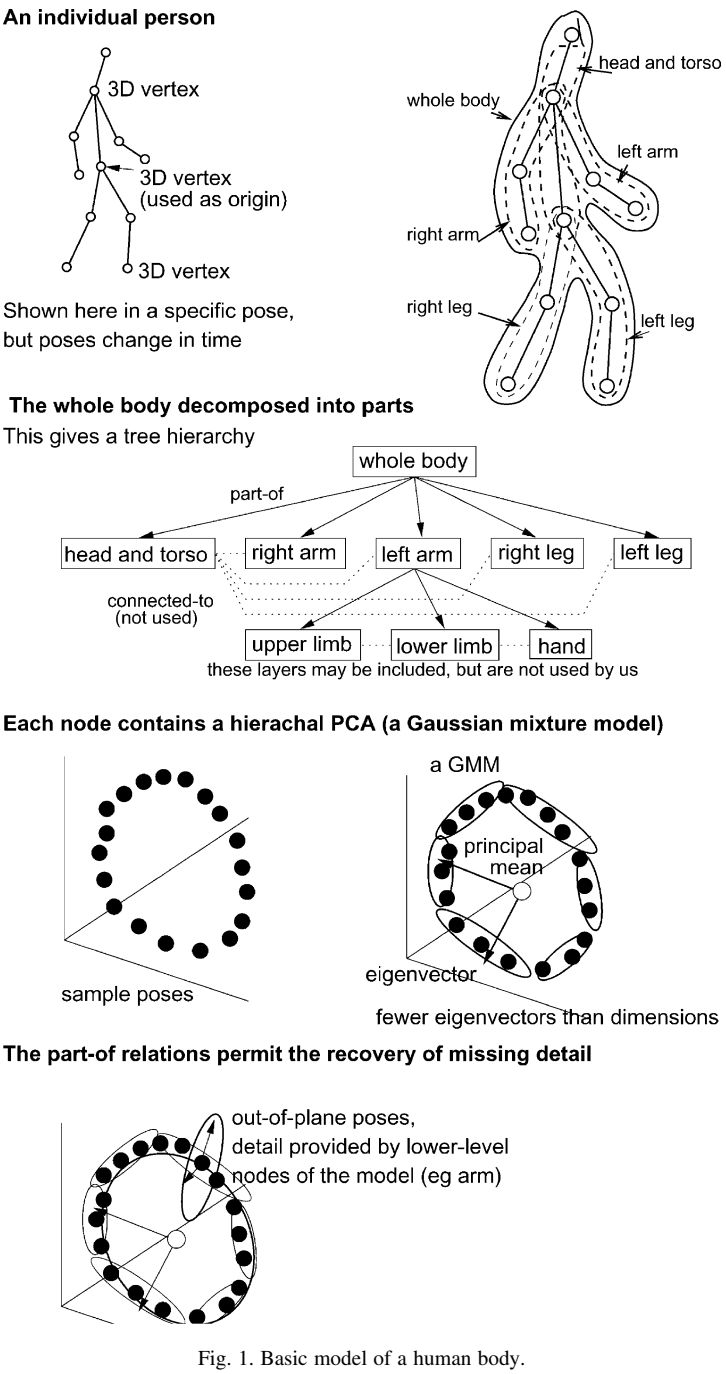


Fig. 1. Basic model of a human body.

variants of valid poses, the lower-level nodes encode minor variations, and hence capture detail that would otherwise be lost, in a compact way. We make use of this when tracking humans, as explained in Section 3. Overall, our model greatly improves specificity and yet retains the advantages of PCA.

2.2. Modelling dynamics

In our model GMM captures the variety of poses the figure can have, but we also would like to have a mechanism, which given a human figure pose at time t would be able to predict what pose the figure is likely to acquire at time $t+1$. For this purpose we adopt HMMs.

HMMs have been used for some time in the speech processing [3,7] representing possible transitions from one sound into another. Recently they have found use in computer vision for interpreting and predicting human behaviour [16,18]. Currently, for reasons of simplicity, we use an HMM only in the root node of our hierarchical model. We will comment further on this in Section 6.

A continuous observation HMM consists of the following elements:

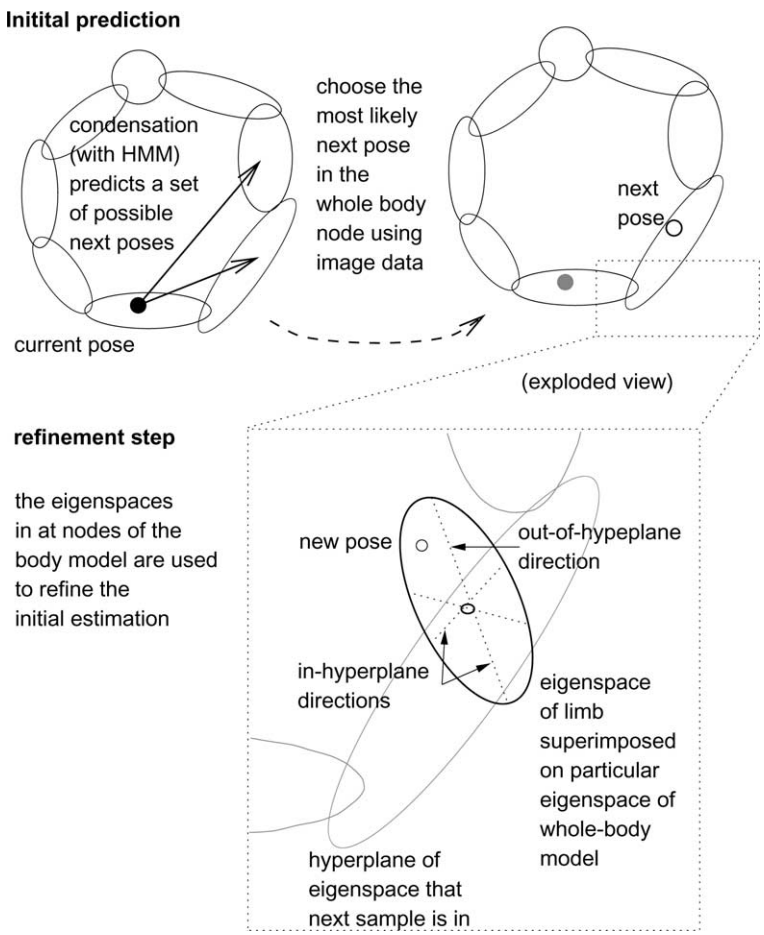


Fig. 2. Tracking process with refinement step.

- $t_1, t_2, t_3, t_4, \ldots$, which are discrete clock times.
- $q_1, q_2, \ldots, q_N$, which are a number of discrete states. In our case each state is represented by an eigenmodel within a GMM. At each clock time t a new state is possibly entered.
- $\mathbf{A} = \{a_{ij}\}$, $a_{ij} = p(q_j \text{ at time } t+1 | q_i \text{ at time } t)$, which are the probabilities of transitions between states.
- $\mathbf{B} = \{b_j(o)\}$, where $b_j(o) = p(o_t | q_j$ at time $t)$ is an observation density distribution at state j, which is just the probability that a sample o_t belongs to state j.
- $\mathbf{\Pi} = \{\pi_i\}$—initial probabilities of being in state i at time $t = 1$.

We initialise the matrix of possible transitions with all elements equal, and use the Gaussian components of the GMM at the root node of our hierarchy as the observation densities in order to estimate the transition probabilities using the Baum-Welch [15] iterative method. Equivalently, estimating Gaussian components can be done in a single procedure together with estimating the HMM [7].

So far we have described our hierarchical model which represents the geometry and dynamics of a collection of people. In Section 3 we explain how we use this model for tracking.

3. Tracking human figure in a video sequence

We aim to track a figure in image sequences, from frame to frame. In principal the images can be 3D (perhaps acquired from a body scanner) or from two dimensional image sequences as obtained from a video camera; our tracking method is largely independent of image modality. This is because we track figure poses in the model just described, and use data only to choose between a set of poses that have been generated using the condensation algorithm [11].

Our tracking process can be thought of as a multi-level refinement procedure. It starts by estimating the pose of the whole human body using the HMM and the eigenspace at the root node of our hierarchical model. Then it refines the poses of the body parts using the eigenspaces of the corresponding model nodes (Fig. 2).

It is convenient to start describing the tracking procedure with describing the estimation of the pose of the whole figure. This step involves using the condensation algorithm in combination with HMM to generate a set of poses in the top-level node eigenspace. We initialise the condensation algorithm by generating n/k sample poses in each Gaussian of the top-level GMM, where n is the total number of the

samples and k is the number of Gaussians. We then weight each sample $\mathbf{s}_i$ against the image data. In case of manually pre-segmented data (Section 4) we weight each sample with the sum of the Euclidian distances between the estimated image landmark positions and pre-segmented image landmark positions. In case of automatically segmented image data (see details in Section 5) we weight each sample with the sum of the absolute values of the differences of the corresponding estimated and automatically segmented binary images. Finally, we estimate the fitness of each sample $\mathbf{s}_i$ as $f_i = \exp(-w_i/C)$, where w_i is the weight of the sample and C is a constant, and normalise these values so that $\sum_{i=1}^{n} f_i = 1$. We use the fitness values to generate the distribution of samples in the next step. We estimate the figure pose in the current frame as the weighted mean of the current sample set, where the weights are the fitness values.

In the following iterations of the condensation algorithm we select $n - k$ samples from the old samples on the basis of their fitness values as in the condensation algorithm [11]. To assist to the recovery from failures we additionally generate a sample from each Gaussian, k samples altogether. We find the probability of each sample belonging to each of the Gaussians and then assess the likelihood of the next pose belonging to each of the Gaussians on the basis of the above probability values and HMM transition probability values. We use this likelihood to generate a corresponding proportion of the total number of samples in each Gaussian. Then we estimate the new samples fitness with the image data in the same way as above. This method allows us to track the pose of the figure as the pose changes from one cluster to another with a limited number of samples. In our experiments we use approximately 250 samples on each iteration of the condensation algorithm.

Refinement of a particular body part pose is performed in the following way. The estimated pose of this part is passed from the previous refinement stage. This pose is projected into the eigenspace of the corresponding model node and the probability of it belonging to each cluster of this eigenspace are estimated. A set of samples from each cluster in the eigenspace is then generated, the number of samples belonging to each cluster proportionate to the obtained probabilities. The samples of the part poses are then reconstructed to their original space and each sample is assigned a weight according to how well it fits the data in the current input frame. The refined part pose is estimated as a weighted mean of the reconstructed set.

4. Experiments with manually pre-segmented images

In this article we restrict ourselves to considering walking motion of a small sample of people. In this section we track pre-segmented landmark points. In Section 5 we track automatically segmented features. We performed a number of experiments in both 2D and 3D. The experiments in 3D show that we are able to recover 3D configurations of the skeleton on the basis of previously unseen 2D image data, invariant of the camera view. The 2D experiments show that the system is able to track both people it has been trained on and people it has not been trained on. We also showed that using our hierarchy noticeably improves the precision of tracking in 2D. To monitor the precision of tracking we computed the error for each skeleton vertex as the Euclidian distance between the tracked vertex position and the ground truth vertex position. We experimented with different coordinate systems for representing 3D and 2D skeleton vertex positions, including Cartesian, Polar, and Twist representations, but so far we have found Cartesian coordinates to give the best results in the experiments. We also experimented with including vertex velocities in our data set, but this did not provide significant improvement in tracking.

4.1. 3D experiments

The data consists of 320 frames of a walking 3D human skeleton which was captured using an optical marker-based system consisting of eight cameras. The human skeleton is represented by 32 vertices and connecting bones (Fig. 3). The configuration of the skeleton in each frame is represented by a state vector consisting of 3D Cartesian coordinates of each vertex. The data we used for training is the 3D data from 200 frames, the rest of the frames were used for testing. The two sets of 2D testing data were obtained by parallel projection of the rest of the 3D data frames into side and front camera views.

Our hierarchy comprises two levels. The root contains the HMM for a whole human skeleton. The second level consists of five nodes; one for the right leg, one for the left leg, one for the right arm, one for the left arm, and one for the torso and the head. Each node contains the GMM for the body part.

We trained our model on 200 frames, keeping 90% of the eigenenergy in the root, leaving just two eigenvectors. The first of these vectors describes forward–backward motion of rigid arms and legs. The second of the vectors describes the degree of bending of the knees and elbows (Fig. 4). We kept 95% of the eigenenergy in the remaining nodes, and used 50 Gaussians in each of the GMMs. In our experiments the transition probabilities in HMM seem to have settled to reasonable values just after 4–5 iterations.

We tracked the 3D skeleton in both of the projected sequences (side and front views) using only one sequence at a time (Fig. 3). When tracking in the side view the average error in the image plane over all vertices and frames was 3.8 pixels with the standard deviation of around 1 pixel, with the vertical size of the whole figure being 160 pixels (Fig. 5). The precision is better for the upper part of the body including arms but worse for the legs. When the recovered 3D model was projected into the front view, the average error in the (new) image plane was only 1.15 pixel. We attribute this to the fact that there is more variation in the

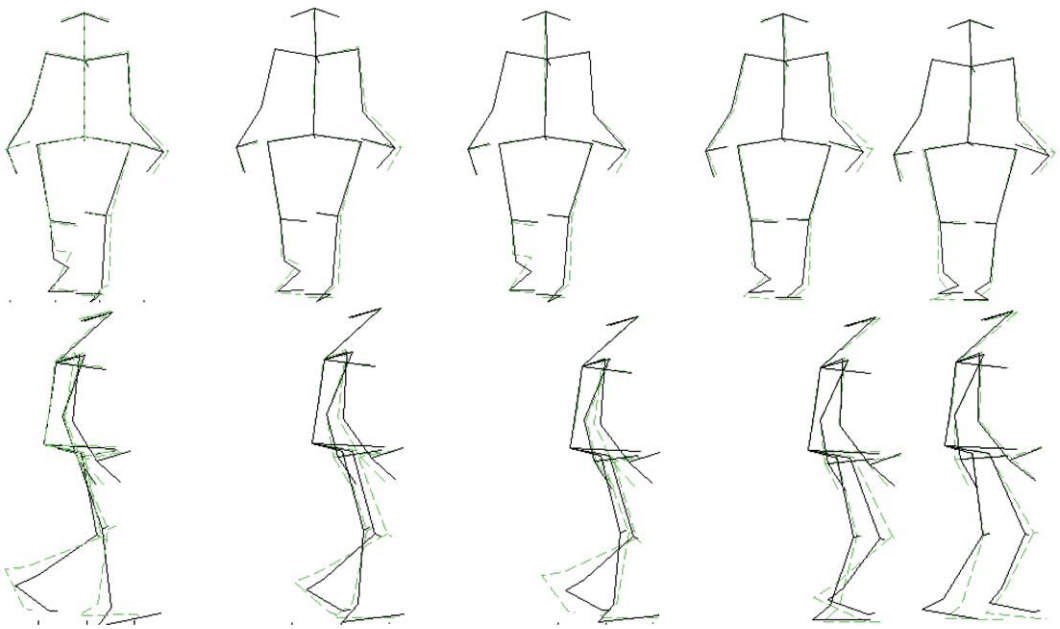


Fig. 3. Tracking skeleton in 3D: front and recovered side views. The tracked figure is drawn with solid black lines and the ground truth figure is dashed grey.

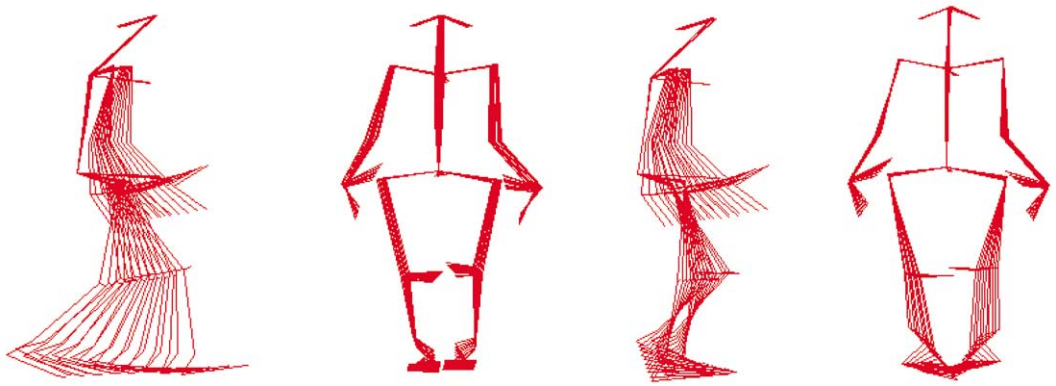


Fig. 4. Two main modes of 3D variation in the global eigenspace, side and front views.

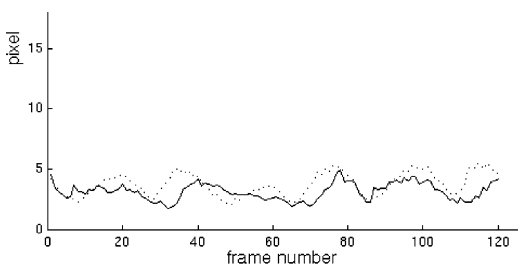


Fig. 5. Average error of tracking a skeleton in 3D using a side view: the error of tracking using only the top level of the hierarchy is shown in dotted black line, the error of tracking using all levels of the hierarchy is shown in solid black.

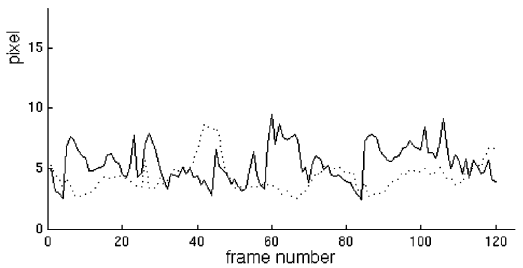


Fig. 6. Average error of tracking a skeleton in 3D using a front view: the error of tracking using only the top level of the hierarchy is shown in dotted black line, the error of tracking using all levels of the hierarchy is shown in solid black.

Fig. 7. Two frames from a video sequence in which we track a 2D skeleton: the ground truth figure is shown in white solid line, the first estimate is shown in dotted line, and the refined figure is dashed white.

side view comparing to the front view (Fig. 4) and that the 3D hierarchical model has been trained on insufficient data, but this observation is worthy of further investigation.

When we used the second level of the hierarchy for fine-tuning we found that the results were not significantly different, and on occasion worse. The average error was 3.2 pixels with the standard deviation of around 0.7 pixel (Fig. 5). We conjecture that results would improve were we to use HMMs at each node in the skeletal model. This is a subject of future research.

When tracking using the front view, the average error was 4.4 pixels with the standard deviation of 1.3 pixels. The results were slightly worse (mean error of 5.5 with the standard deviation of 1.5 pixel) when we used the second level of the hierarchy (Fig. 6). We attribute this mainly due to a large degree of ambiguity of the front view. This ambiguity may be resolved in part by perspective projection, but also by appealing to additional information in the image such lighting information. The use of HMMs on all the levels of the hierarchy could also help.

4.2. 2D experiments

The training and testing data was obtained by hand-marking 29 video sequences of five people walking parallel to the camera field of view, each about 40 frames long, thus giving around 1000 frames altogether. The skeleton figure consists of nine connected vertices representing the right side of the human body, right leg, right arm, right half of the torso and head positions (Fig. 7).

The model was trained on 21 video sequences chosen from 29 that were available. It was tested on the remaining six video sequences of people it had been trained on, and also two video sequences of two people it had not been trained on.

The skeleton model consists of two levels, the first level being for the right-hand side of the whole body and the second level consisting of three nodes, one for the right leg, one for the right arm and one for the right part of the torso and head.

The average error for a person the model had been trained on is 6.8 pixels with the standard deviation of 1.3 pixel when using second level of the hierarchy and 9.4 pixels with the standard deviation of 1.2 pixel when using only the top level of the hierarchy, with the vertical size of the whole figure being about 400 pixels (Fig. 8). This demonstrates an improvement. The average error for a person the model has not been trained on is 12.6 pixels with the standard deviation of 1.4 pixels when using the full hierarchy. For a whole body model alone the average error is 17.5 pixels and the standard deviation is 2.2 pixel. Again, there are benefits to be had from the hierarchical model.

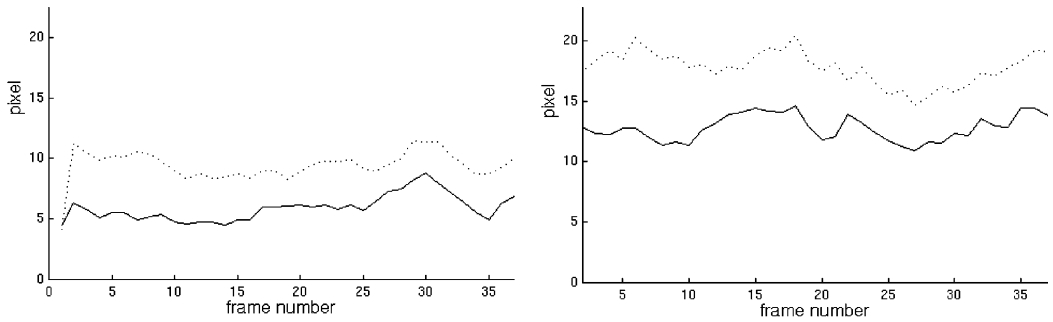
5. Experiments with automatically segmented images

In these experiments we complement our model with volumetric parts to model the human body. We label the arcs between body vertices with truncated cones and the head vertex with a sphere. Thus we have a volumetric artifice to our model, which we use in tracking.

For these experiments we use images obtained from a single calibrated video camera. We assume that we have trained a hierarchical model on the 3D training set describing walking human motion.

We need to find the 3D position of the human figure in the image, i.e. the 3D position of the coordinate origin associated with that figure in relation to the world coordinate system. We also need to find the 3D pose of the human figure in the image. To do so we follow these steps:

1. Calibrate a single video camera.
2. Sample background with no person present.
3. Identify the person walking by removing the background, leaving a binary image with a foreground figure.



(a) (b)

Fig. 8. Average error of tracking a skeleton in 2D: (a) the error of tracking a person it has been trained on before; (b) the error of tracking a person it has not been trained on before. The error of tracking using only the top level of the hierarchy is shown in black dotted line. The error of tracking using all levels of the hierarchy is shown in black solid line.

4. Estimate the three-dimensional location of the person's 'centre of gravity', this is the origin of our figure.
5. Estimate the whole body pose using condensation algorithm [11] in conjunction with an HMM using the eigenspace at the root node of our hierarchy.
6. Refine the poses of the body subparts using the corresponding nodes in our hierarchical model.
7. Move to the next frame of the image sequence, repeat from step 3.

We now explain these steps in greater detail. The camera is calibrated using Tsai's standard method [17], and background removed to leave a foreground figure using a method due to Horprasert et al. [10].

We calculate the 3D origin of our person in the following way: the x coordinate is estimated using the x coordinate of the centre of gravity of the figure in the image. Due to perspective the corresponding y estimate is inaccurate—for example, the person's feet contribute to the projected 'height' of the person in the (2D) image. We, therefore, use a scaled ratio of the height of a bounding box that encloses the figure. In our experiments we have found that a ratio of 5/12 gives reasonable result. However, this is a parameter that could be learnt for an individual and calibrated for a known camera position. The third component, depth, is then estimated using standard geometric projection: knowing the height, in pixels, of the bounding box above, the know physical height, in millimetres, of the individual and the calibrated camera focal length, the depth is readily calculated.

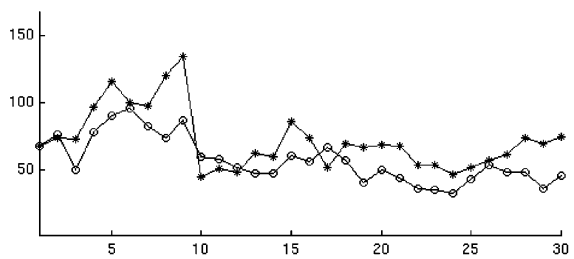


Fig. 9. Error of tracking (in mm) on a synthetic video sequence. The $*$'s show the error without the use of the hierarchy, the o's show the error when using the hierarchy.

For the initial estimate of the body pose in each video frame we make use of the hierarchical structure of the kinematic-dynamic model (Fig. 1). First we produce an initial estimate by using the condensation algorithm in conjunction with an HMM (as described in Section 3) to generate a set of body poses in the root node eigenspace of our hierarchical model (Fig. 2). These poses are then reprojected to their original 3D space, and weighted according how well they fit the current image. The whole body pose is estimated as the weighted mean of the reconstructed set.

We test each generated human body pose against the image data in the following way. Each pose is used to fix in space a volumetric figure modelling the human body, and this is perspectively projected into the viewing plane. For each body part we project a truncated cone or a sphere whose position is defined by the pose. For testing we measure the correlation between the projected figure and the image figure. The correlation measure we use is the sum of the absolute values of the differences of the two images. Both images are binary as illustrated in the image of the reconstructed body (Fig. 10, second and third columns).

Below we show experimentally that our method is able to reconstruct from a monocular video sequence without the use of markers and in the presence of partial occlusion. We also obtained an estimate of three-dimensional accuracy using synthetic image sequences.

We obtained training 3D motion data by filming a person's motion with three cameras placed in front of that person with different viewing angles. Nineteen markers were placed on a person at joints and extremities. The 2D positions of the markers were hand-marked in each of the video-frames and their 3D positions reconstructed knowing each camera's calibration parameters. We built our hierarchical model to include the motion of the whole body and five submodels describing the motion of each leg, arm, and a head and a torso in more detail.

Fig. 10 shows selected images from a video sequence, with the corresponding reconstructed 3D figures

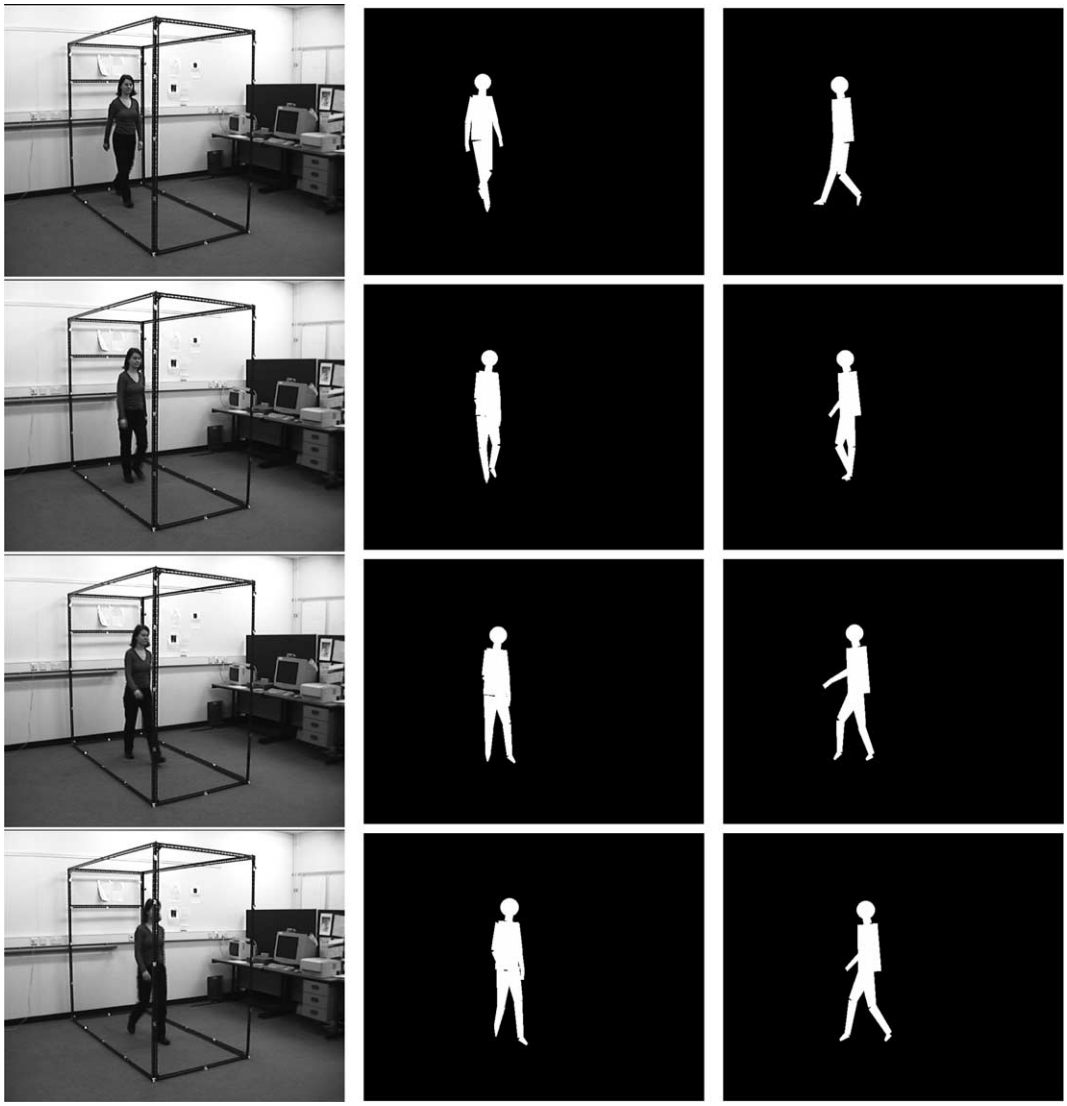


Fig. 10. Original images (1st column) and corresponding reconstructed 3D figures projected into original (2nd column) and side (3rd column) views.

projected into the same view shown side by side. In the right column is the figure, in the same pose, shown from an alternative view point. It should be noted there is a partial occlusion, which has been handled when reconstructing 3D pose of the human body, this is shown in the last two images and is present in many more images in the sequence. Self-occlusions are also present in all of the frames (a leg is occluded in the first shown frame, and an arm in the rest of the frames).

To assess the accuracy of our tracking, we generated a synthetic 2D video sequence using the hand-marked 3D motion training data. The figure was composed of truncated cones (connecting the joints located by the 3D markers) and a sphere (head), which were projected onto a view plane. Our video sequence comprised of 30 frames. We then tracked the generated human figure in the resulting video sequence and compared the extracted 3D positions of the vertices on the body to the ones that were used to generate the synthetic video sequence. We performed the comparisons (a) using only the top level of the hierarchy, which is a standard HPCA of the motion of the whole body, (b) all levels of our hierarchical model. We calculated the error as the average Euclidian distance over all the vertices in each frame. Fig. 9 shows the error of tracking (in mm) on the synthetic video sequence with and without the use of the hierarchy. We can see that for most of the video frames the hierarchy improves the precision. In general, the graph of tracking error with the use of the hierarchy is smoother than without the hierarchy. In two frames, the 3D error with the use of the hierarchy is actually worse than without the use of the hierarchy. This happened because when fitting the 3D model into the 2D image data we were measuring the correlation between the 2D data sets, and occasionally the depth fit can become worse while the rest of the fit becomes better.

6. Conclusions and future research

We described a novel hierarchical model for view independent tracking of the human figure in monocular video sequences. The main contribution of our hierarchical model is the representation of minor variations of a 3D data set in a useful and compact manner, which allows greater specificity while tracking. We trained and tested the model on 3D data and showed that the system is capable of deriving 3D data from just one, not specified, 2D view.

We also trained the system on 2D data collected from the video sequences of three different people. The precision improved when we used the second level of hierarchy. The system was also able to track the 2D skeleton of a person it had not been trained on, thus showing that it is general enough to track different people, including previously not seen.

Our model is not homogeneous—HMM appears only at the root node. This may explain the deterioration of the performance in particular 3D situations when using the whole hierarchy. However, the 3D data was in insufficient quantity for us to be sure of our conclusions in this regard. Clearly, further work is needed. Nonetheless we were able to demonstrate view-independence using 3D data.

In our future work we are also going to make our models extendible by building on our previous work in Ref. [8].

Acknowledgements

The authors of this article would like to thank Tim Child from Televirtual for kindly providing us some of the experimental data and the practical participants of experiments for making the data acquisition possible.

References

[1] C. Bregler, J. Malik, Tracking people with twists and exponential maps, IEEE CVPR Proceedings (1998) also available at http://wwwcsberkley/edu/~bregler/pubs.html.

[2] R. Bowden, T.A. Mitchell, M. Sarhadi, Reconstructing 3D pose and motion from a single camera view, BMVC Proceedings (1998) 904–913.

[3] J. Deller, J. Proakis, J. Hansen, Discrete-time Processing of Speech Signals, Macmillan, New York, 1993.

[4] L. Goncalves, E. Bernardo, E. Ursella, P. Perona, Monocular tracking of the human arm in 3D, ICCV Proceedings (1995) 764–770.

[5] D.M. Gavrila, L.S. Davis, 3-D model-based tracking of humans in action: a multi-view approach, CVPR Proceedings (1996) 73–79.

[6] D. Hogg, Model-based vision: a program to see a walking person, Image and Vision Computing February (1983) 5–20.

[7] X. Huang, Y. Ariki, M. Jack, Hidden Markov Models for Speech Recognition, Edinburgh University Press, Edinburgh, 1990.

[8] P. Hall, D. Marshall, R. Martin, Adding and subtracting eigenspaces, BMVC Proceedings September (1999) 453–462.

[9] T. Heap, D. Hogg, Improving specificity in pdms using a hierarchical approach, BMVC Proceedings September (1997) 80–89.

[10] T. Horprasert, D. Harwood, L.S. Davies, A statistical approach for real-time robust background subtraction and shadow detection, Proceedings of IEEE ICCV'99 FRAME-RATE Workshop, Kerkyra, Greece September (1999) available at http://www.cs.umd.edu/users/thanarat/Publication.html.

[11] M. Isard, A. Blake, Condensation-conditional density propagation for visual tracking, International Journal of Computer Vision 28 (1998) 5–28.

[12] P.H. Kelly, E.A. Hunter, K. Kreutz-Delgado, R. Jain, Lip posture estimation using kinematically constrained mixture models, BMVC Proceedings September (1998) 74–83.

[13] I.A. Kakadiaris, D. Metaxas, Model-based estimation of 3D human motion with occlusion based on active multi-viewpoint selection, CVPR Proceedings June (1996) 81–87.

[14] E.J. Ong, S. Gong, A dynamic human model using hybrid 2D–3D representations in hierarchical PCA space, BMVC Proceedings September (1999) 33–42.

[15] L. Rabiner, B. Juang, An introduction to hidden Markov models, IEEE ASSP Magazine January (1986) 4–16.

[16] T. Starner, A. Pentland, Real-time american sign language recognition from video using hidden Markov models. MIT Media Laboratory Perceptual Computing Section Technical report No. 375, available at http://vismod.www.media.mit.edu/cgi-bin/tr_pagemaker.

[17] R.Y. Tsai, An efficient and accurate camera calibration technique for 3D machine vision, IEEE CVPR Proceedings (1986) 364–374.

[18] M. Walter, A. Psarrou, S. Gong, Learning prior and observation augmented density models for behaviour recognition, BMVC Proceedings September (1999) 33–42.

[19] A. Watt, 3D Computer Graphics, third ed., Addison-Wesley, Reading, MA, 2000.