

# Viewpoint invariant exemplar-based 3D human tracking

Eng-Jon Ong \*, Antonio S. Micilotta, Richard Bowden, Adrian Hilton

*Centre for Vision, Speech and Signal Processing, SEPS, University of Surrey, Guildford GU2 7XH, Surrey, UK*

Received 16 February 2006; accepted 8 August 2006

Available online 2 October 2006

## Abstract

This paper proposes a clustered exemplar-based model for performing viewpoint invariant tracking of the 3D motion of a human subject from a single camera. Each exemplar is associated with multiple view visual information of a person and the corresponding 3D skeletal pose. The visual information takes the form of contours obtained from different viewpoints around the subject. The inclusion of multi-view information is important for two reasons: viewpoint invariance; and generalisation to novel motions. Visual tracking of human motion is performed using a particle filter coupled to the dynamics of human movement represented by the exemplar-based model. Dynamics are modelled by clustering 3D skeletal motions with similar movement and encoding the flow both within and between clusters. Results of single view tracking demonstrate that the exemplar-based models incorporating dynamics generalise to viewpoint invariant tracking of novel movements.

© 2006 Elsevier Inc. All rights reserved.

*Keywords:* Exemplar-based; 3D human tracking; View-invariant; Learnt motion model

## 1. Introduction

The goal of this paper is to create a method to track a person regardless of the viewpoint. We also require that the 3D pose of the human body be recovered. The articulated structure of a human body makes it a complex object to model visually. This in turn introduces a number of challenging problems that need to be addressed before any form of tracking can be done.

The first problem has to do with the large variations present in the appearance of a human body. This is because of the high number of degrees of freedom in the human body, even for a small subset of actions (e.g., walking, running). This problem is further compounded by the fact that the 2D visual appearance of the human body can change dramatically for a single pose from different viewpoints. We address this problem by first grouping visual appearances based on the underlying skeleton. We then integrate both the visual information and body pose into an

*exemplar*. Specifically, an exemplar will incorporate the underlying 3D information of the human body as well as the appearance across different viewpoints. We will see that this is important for generalisation.

The physical variations of the human body will still require us to have many different exemplars. From a tracking perspective, there is the problem of increased complexity when trying to search for the exemplars that best match a given input image. Typical approaches estimate the underlying structure starting from the visual information. Our approach is the opposite of this, starting instead at the underlying structure. A suitable framework for this purpose is the particle filter. However, to use the particle filter, we first need a learnt model that can provide both the pose constraints and dynamics information for the exemplars.

This is achieved in two steps. We first segment the exemplar space into a set of clusters. These clusters represent the limits on human body poses, in effect, providing important physical constraints on the possible poses a body can assume. The next step is to model the dynamics of the exemplars. In our case, this involves modelling how

\* Corresponding author. Fax: +44 1483 686031.  
E-mail address: [e.ong@surrey.ac.uk](mailto:e.ong@surrey.ac.uk) (E.-J. Ong).

exemplars transit between different clusters and how they flow within an individual cluster. We achieve this by retaining the original sequence of exemplars or *flow vectors* that exists within a cluster. These flow vectors will be used to guide particles within a cluster. Additionally, they also provide important information on where particles should enter a cluster and where it has to exit a cluster.

The rest of the paper will be structured as follows. In the next section, we review previous related research work. In Section 3, we will describe the viewpoint invariant representation for the exemplars. Following this, we will show how these exemplars are clustered into groups of similar motion segments and its non-linear dynamics model in Section 4. Section 5 presents the application of the dynamics model for visually tracking the 3D pose of a human subject. In Section 6, we show results on the generalisation capability of our method, using the particle filter to track a subject following novel walk trajectories. Finally, in Section 7, we provide some conclusions and areas for further improvement.

## 2. Related work

One related approach was presented by Ren et al. [1]. Motion capture data were used in conjunction with the associated 2D silhouettes to track the 3D motions of a person for animating human characters. Here, discriminative silhouette features are learnt for estimating body pose and orientation. These are then used in a locality-sensitive hashing algorithm for locating the best body pose in a new image. Our method is different in that we have an exemplar flow-based model directly integrated into a particle filter framework for tracking the orientation and body pose of the subject. Other related approaches usually involve fusing 2D and 3D information into a high-dimensional vector [2,3], before learning a statistical generative model over them (e.g., Gaussian mixture models). The training data will then be discarded and replaced by the generative model. Another approach to learning generative models is to learn a low-dimensional manifold of the training data. Visual tracking of the 3D pose can then be restricted to the manifold, and the generative functions used to reproduce the original form. An example of this is presented by Sminchisescu and Jepson [4]. Here, a lower-dimensional manifold of the training data were modelled using a layered generative model. This was then applied to tracking the 3D human poser from a single camera. It was also shown by Rahimi et al. [5] that manifold learning can be improved when temporal data and user supervision are used. The recovery of 3D body poses can alternatively be done by learning separate mappings between the manifold space to the input (2D silhouette) and output space (3D body pose), as proposed by Elgammal and Lee [6]. To recover the 3D pose, the silhouette is first mapped into the manifold space. Next, the point in the manifold space is mapped to the 3D pose space. To handle silhouette variations arising from viewpoints changes, several view-dependent

models were learnt. In our work, we avoid the need for multiple models by using viewpoint-invariant exemplars.

A closely related approach is the exemplar-based approach where all the training examples are retained, and each 3D-pose is associated with a single corresponding 2D view. It is also very useful to have a method that can efficiently index into the best exemplar, as described by Shakhnarovich et al. [7]. Here, a parameter sensitive hashing method was developed for estimating the upper body pose, whilst providing sub-linear efficiency for finding the best matching exemplar. Stenger et al. [8] used the exemplar method for tracking the 3D pose of hands. Toyoma and Blake [9] developed an exemplar contour-based approach. Alternatively shape context-based exemplars can also be used to recover the 3D pose, in this case, the human body pose, as described by Mori and Malik [10].

When the motion of a subject is known *a priori*, it is possible to use motion models to provide additional constraints. One closely related exemplar-based motion model was recently described by Jenkins et al. [11] for use in controlling and interacting with a robot. Kinematic information in the form of joint angles is split into smaller modules. Each module encodes the flow of a small subset of kinematics motion. A spatio-temporal Isomap is used to perform clustering. Another possibility is to incorporate a limited amount of dynamics information directly into the exemplars themselves as recently proposed by Dimitrijevic et al. [12]. This was achieved by introducing spatio-temporal templates. Here, visual information (silhouette and edge orientation) from three consecutive frames is combined into a single template. This was then used to find key-poses of people walking. Temporal information can also be incorporated into a probabilistic model of the exemplars, as proposed by Sminchisescu et al. [13]. The probability of a particular exemplar (3D body pose) generating some visual input observation is modelled as a bayesian mixture of experts. It was then applied to finding the best exemplar from 2D inputs on various motion sequences of picking and dancing.

An alternative was proposed by Brand [14] where a HMM manifold in a configuration space was learnt. A similar approach was also used by Sidenbladh et al. [15] for tracking walking motions of a subject in using a single camera. An alternative was proposed by Howe et al. [16], where PCA and mixtures of Gaussians were used to model a high-dimensional vectors of concatenated 3D human motion segments. This was then used for tracking the 3D body pose of the human body. Along similar lines, Urtasun and Fua [17] proposed temporal motion models that were learnt using principal component analysis (PCA) for multiple motions. However, it is still unclear as to how issues due to viewpoint change caused by novel motion trajectories (e.g., turning walking motions) are handled by these approaches.

A recent alternative was proposed by Agarwal and Triggs [18] where regression methods are used to combine visual information (i.e., silhouette shape context information)

to yield the correct underlying 3D pose of the human body using a single camera. However, to cope with ambiguities that arise in tracking 3D objects in a single view, the dynamics information from previous state's was used. This approach was demonstrated too work on long sequences of a subject performing various actions. On the other hand, Ramanan et al. [19] has recently described work that does not use motion capture information. Instead, 2D textured rectangles of various segments of the body are learnt. A 2D puppet structure of these linked 2D texture rectangles is tracked using a loopy inference procedure on an underlying Bayesian net. This approach has presented good results on tracking people performing various activities. A similar approach was then used by Sigal et al. [20] for estimating the 3D pose of the human body. The 3D human body was modelled as a graphical model where relationships between different body parts are captured as conditional probability distributions. Thus, the 3D pose estimation problem becomes one of inference over a graphical model, where random variables correspond to individual limb parameters (e.g., position and orientation). To achieve this, the belief propagation in the graphical model was approximated using a particle filter.

### 3. Representation

This section presents the exemplar representation of human pose and view-dependant appearance. Each exemplar contains two parts; the underlying skeleton structure and its associated visual information.

#### 3.1. Underlying 3D skeleton

The underlying 3D human skeleton (Fig. 1a) will take the form of unit quaternion angles of joints on the body. Formally, we define the number of joints as  $N_j$ . The quaternion angle of the  $j$ th joint is represented by four numbers;  $(q_{j,1}, q_{j,2}, q_{j,3}, q_{j,4})$ . Finally, the 3D human skeleton of the  $i$ th exemplar can be represented by the vector:  $q = (q_{1,1}, q_{1,2}, q_{1,3}, q_{1,4}, \dots, q_{N_j,1}, q_{N_j,2}, q_{N_j,3}, q_{N_j,4})$ .

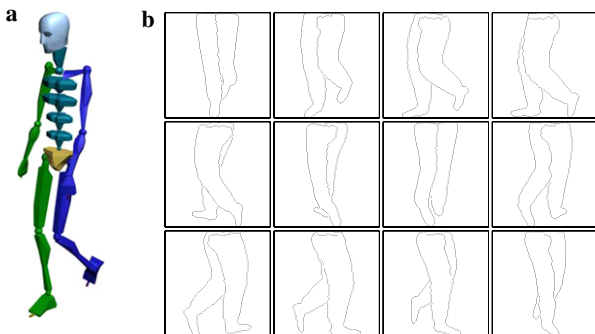


Fig. 1. An exemplar consists of an underlying skeleton (a) and a set of visual appearances around the skeleton (b).

#### 3.2. Visual information

Associated with each single 3D skeleton is its corresponding visual information. To this end, we have chosen to use an unordered set of edge pixels or “edgels” that lie on the contours of a person. The use of edgels provides us with useful information on internal edges of a person. This provides an advantage in differentiating ambiguous poses over the commonly used silhouettes that only provide information on the outer contour of the body shape. The number of edgels associated with a particular view of a person at a particular pose is defined as  $N_e$ , for the  $e$ th view. The edgel set can be represented using a vector of concatenated points as follows:  $c = (x_1, y_1, \dots, x_{N_e}, y_{N_e})$ .

For a fixed pose, we have a fixed number of views ( $N_v$ ) arranged in a circle around the skeleton (Fig. 1b). Formally, a 3D skeleton represented by its joint angles  $q$  can be associated with a set of 2D view edgels  $v = \{c_1, \dots, c_{N_v}\}$ .

#### 3.3. Training exemplars from motion capture data

In order to obtain the data described above, we have chosen to exploit the increasing availability of motion capture data. To do this, we have at our disposal various motion data of a person performing a variety of different actions. The motion capture data are in the Biovision format. This provides Euler angles for each of the joints and the translational offset for the entire body. The angles can be converted into quaternions for the 3D skeleton data.

In order to obtain the visual data, 3D-studio was used to provide a rendering platform. A generic human model was skinned onto the 3D skeletal structure present in the motion capture data. The body parts were coloured differently to aid the edgel extraction process. A set of 12 virtual cameras was created around the model. The 12 views for every frame in the motion sequences were rendered against a black background and saved. The edgel set of each view was then extracted using a standard edge filter.

### 4. Learnt exemplar motion model

In this section, we introduce the learnt dynamics model for the exemplars. This model will then be incorporated into a particle filter framework described in the next section to enable us to visually track a subject.

#### 4.1. Clustering motion sequences

The first step is to partition the exemplar space into a group of clusters. The clustering is performed using only the 3D skeleton joint angles of the exemplars. The  $K$ -means algorithm was chosen for this purpose, as it provides a simple but effective mechanism for partitioning a space into a number of distinct and separate regions (see Fig. 2a). We define the number of clusters to be  $N_C$ . Each cluster is then associated with a distinct group of exemplars that have similar 3D body poses.

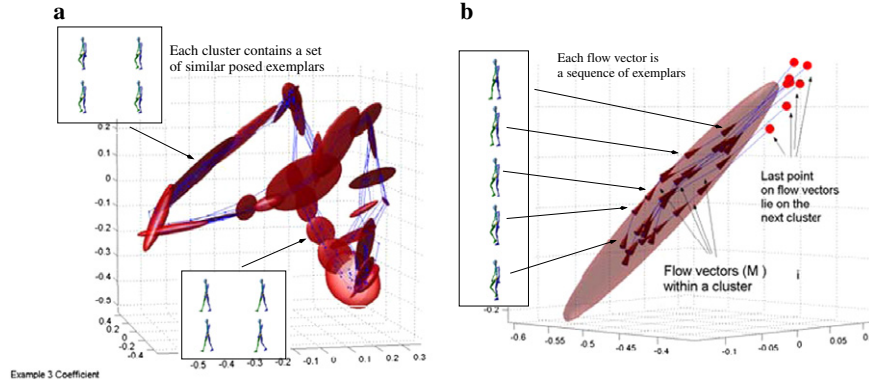


Fig. 2. PCA visualisation of the set of motion sequence clusters (a). Each cluster contains a set of flow vectors. Each flow vector is an exemplar motion segment (b).

4.2. Inter and intra cluster motion flow

To represent human motion dynamics, we model the flow within and between clusters. This requires information on points where one should enter and exit a cluster. Additionally, information on where to go after one has exited the cluster is also important.

The above issues can be addressed by ordering the exemplars within a cluster using the original motion capture data. The result is a set of motion segments or *flow vectors* within each cluster. As can be seen in Fig. 2b, each flow vector has the following properties:

- the start point where it enters a cluster,
- the end point where it leaves the cluster for another cluster,
- ordered points between the start and exit points.

In order to model the dynamics between clusters, we first append to each flow vector the next point on the original motion segment after the final point on the flow vector (shown as green circles on Fig. 2b). This allows us to locate the cluster an exemplar should transit to after following a flow vector within a particular cluster. This in effect links different clusters together.

Formally, a cluster ( $C_i$ ) contains two sets:  $C_i = \{E_i, M_i\}$ .  $E_i$  is the set of exemplars associated with the cluster, and takes the form of a set of indices:  $E_i = \{e_{1,i}, \dots, e_{|C_i|,i}\}$ , where

$|C_i|$  is the number of examples contained by the cluster. These exemplars are in turn organised into a number of flow vectors ( $N_i$ ) which we define as:  $M_i, i = 1, \dots, N_i$ . Each flow vector within a cluster is a vector of indices to the training exemplar database,  $O$ . We define the  $j$ th flow vector belonging to the  $i$ th cluster as:  $M_{i,j} = m_{1,i,j}, \dots, m_{|M_{i,j}|,i,j}$ , where  $|M_{i,j}|$  is the length of the flow vector, and  $m_{l,i,j}$  is a scalar index to a single exemplar in the training database. Note that the very last point on each flow vector is really a point in the *next* cluster after the flow vector leaves the current cluster (see Fig. 2b). This essentially provides the inter-cluster links.

5. Visual tracking

In this section, we will describe how the above cluster model will be integrated into a particle filter to give us the dynamics model for the exemplars. As a result, we mainly provide details on areas where our particle filter is different. For an in depth discussion on particle filters, we refer the reader to existing literature [21].

A particle is defined as the following tuple:  $p_i = \{c_i, f_i, k_i, v_i, w_i\}$ , each element is an index with the following definition (see Fig. 3):  $c_i$  is the cluster that contains the particle;  $f_i$  is the index of the flow vector the particle lies on within the cluster;  $k_i$  is the index of the point on the flow vector, which in turn is an index to an exemplar, or  $m_{k_i, c_i, f_i}$  as defined in Section 4.2;  $v_i$  is the view index into a particular edgel set on the exemplar; finally,  $w_i$  is the error

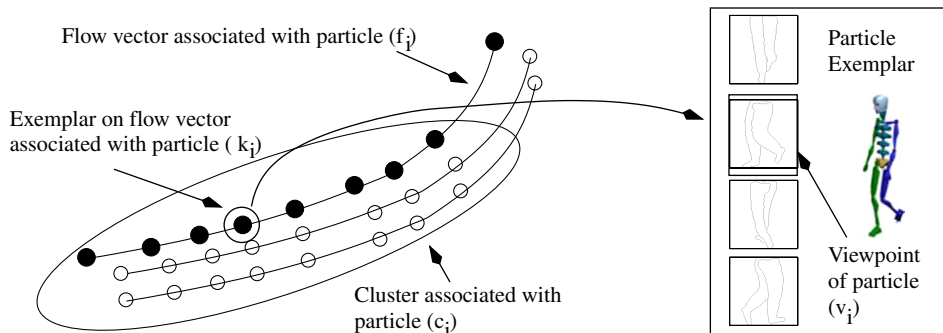


Fig. 3. Visual description of the different components of a particle ( $p_i = \{c_i, f_i, k_i, v_i, w_i\}$ ).

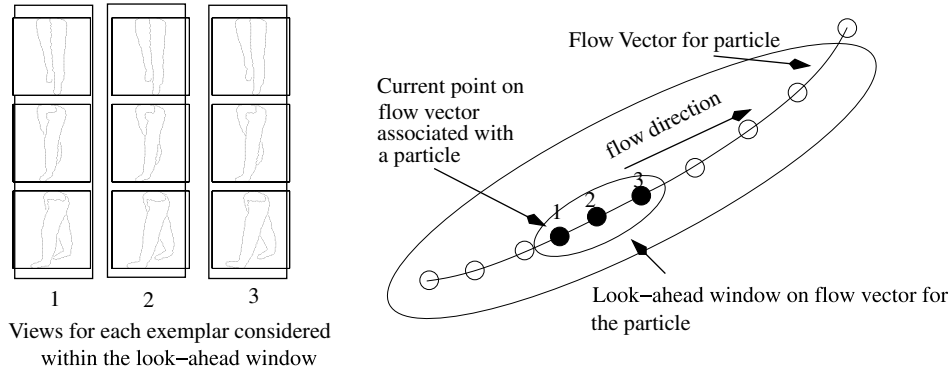


Fig. 4. Visual description of the visual cost look-ahead window.

value for the particle. In practice however, each particle has three additional values,  $x_i, y_i, scale_i$ . The offset values  $(x_i, y_i)$  account for the global translation of the tracked subject, whilst  $scale_i$  is a scaling factor for addressing the size variations of the subject in the image due to perspective effects.

Following initialisation, the propagation of a particle set between time frames proceeds in three stages. The first stage involves the selection of a new set of particles with respect to its error values. The second step is a diffusion step where a prediction of the future state of a particle is made. The third step involves the re-weighting of each particle. This is achieved by measuring how well a particle fits onto a given input image.

5.1. Particle diffusion

In order to predict the next state of a particle ( $p_i = \{c_i, f_i, k_i, v_i, w_i\}$ ), the particle is moved forward on the flow vector if a random probability value ( $r$ ) exceeds a heuristically set threshold ( $t_r$ ):

$$k_i = \begin{cases} k_i + 1 & r \geq t_r \\ k_i & r < t_r \end{cases} \quad (1)$$

The probability value ( $r$ ) is drawn from a uniform random distribution between 0 and 1. As a result, the threshold ( $t_r$ ) too lies between 0 and 1. It should be noted that when  $t_r$  is set low, a particle is encouraged to jump to the next point on the flow vector very frequently (i.e., frequently changing pose). However, too low a value for  $t_r$  may vary the pose faster than that of the subject, resulting in increasing errors in tracking. Conversely, the rate at which a particle moves along a flow vector, or change in body pose of the subject, can be slowed down by setting  $t_r$  higher. Again, if  $t_r$  is given too high a value, tracking errors will increase due to the pose varying too slowly with respect to that of the tracked subject. For the experiments presented in this paper, it was found that a value of 0.3 worked well.

When a particle reaches the end of a flow vector within a cluster ( $k_i \geq |M_{c_i, f_i}|$ ), it will effectively already be in at the starting point in the next cluster (see Section 4.2). We can then locate the new cluster ( $C_i^{new} = \{E_i^{new}, M_i^{new}\}$ ) that contains the exemplar associated with the particle

$$c_i^{new} | m_{k_i, c_i, f_i} \in E_i^{new} \quad (2)$$

In the new cluster, a flow vector is randomly selected by first drawing from a uniform random distribution of the set  $\{1 \dots N_{c_i}\}$ , where  $N_{c_i}$  is the number of flow vectors in the cluster  $c_i$ . The random number chosen is also the index for the new flow vector. The exemplar within the particle is then initialised to the first point on the selected flow vector.

5.2. Visual cost function

The visual cost function in our particle filter has two functions. The first is to provide an error value that can be used to weight a particle. To get the error of a particle, we need to be able to measure how well it fits an object in a given input image. To achieve this, we have chosen to use the distance transform method [22] to the input edge image (Fig. 5a) to ‘blur’ the edges (Fig. 5b). The distance trans-

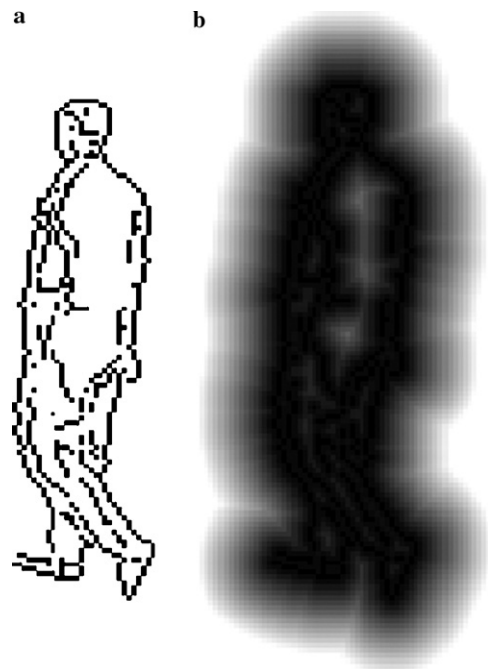


Fig. 5. An edge filtered image (a), resulting distance transform image (b).

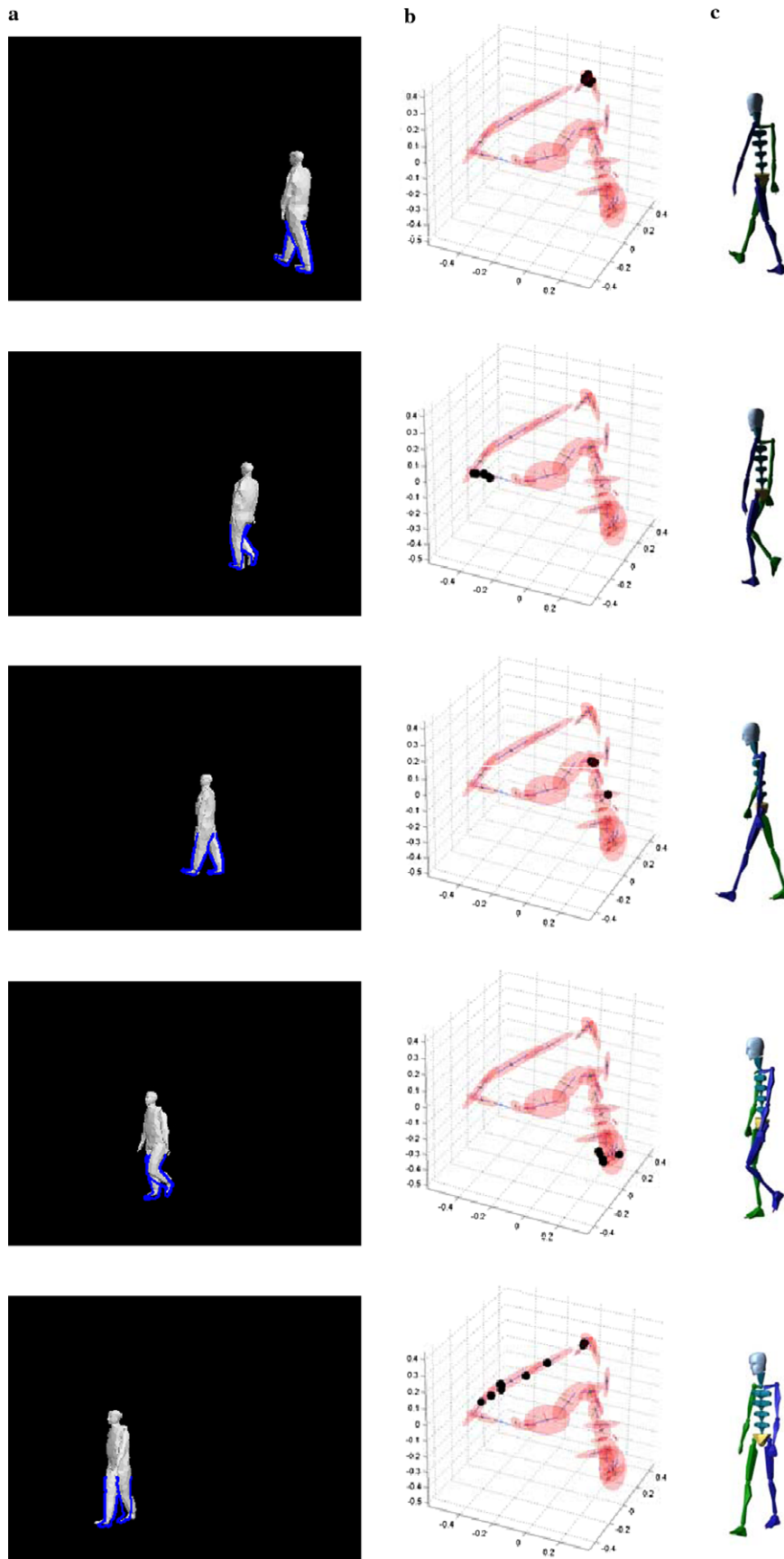


Fig. 6. Results of tracking performed on the 2D rendered sequence (159 frames) of a person turning left for frames 1,35, 69, 102, and 129, respectively. (a) Shows the input image, (b) shows the positions of the 10 best particles within the cluster model, (c) the skeleton of the best particle.

form specifies the distance of each pixel to the nearest non-zero edge, which are the darker pixels of the distance image correspond to the original edges. To calculate the error value of a viewpoint associated with a particle, Chamfer matching is used [23]. We first translate and scale all the corresponding edges to that of the particle. The error value is the mean of the distance transform image pixels that fall under the transformed edges.

The second step within the visual cost function involves a “look-ahead” allowing us to consider poses that are further along a flow vector associated to a particle. This allows for variation in speed of the observed motion relative to the corresponding exemplars within the learnt motion model. Additionally, multiple views are considered around the current viewpoint to handle possible turning motions. This is achieved by measuring the error of exemplars further down the flow vector that a particle lies on. Additionally, for each point on the flow vector considered, we look across a window of viewpoints (see Fig. 4). The particle viewpoint and on-flow-vector index are both updated to the viewpoint on the exemplar with the smallest error.

## 6. Experiments

This section will provide results for various experiments for testing the effectiveness of the proposed tracking model in generalising to novel and unseen walking motions. To

this end, it was found that visual information from the lower body was adequate for successful tracking. In the next section, a description of the training data used for generating the exemplars is given. A brief discussion on the initialisation methods used for the experiments is then given. Next, descriptions and results of the test experiments are given. Each experiment applies the learnt model to image sequences that contain the following aspects not present in the training data:

- Arbitrary walking directions (all experiments): the training sequences used only contained walking trajectories down the  $z$ -axis. However, all test sequences will contain walking trajectories in other directions.
- Changing walking trajectories (Experiment 1, Experiment 3): these experiments tests the ability for the model to generalise from a straight walking trajectories to various walking trajectories with turns.
- Novel subjects (Experiment 2, Experiment 3): these sets of experiments also contained tests to see how well the tracking method can cope with both changes in the visual appearance as well as movement styles of novel test subjects.

In order to account for global translations and image scale variations of the tracked subject, a simple background subtraction was first performed on each input image. The mean position and bounding box of the

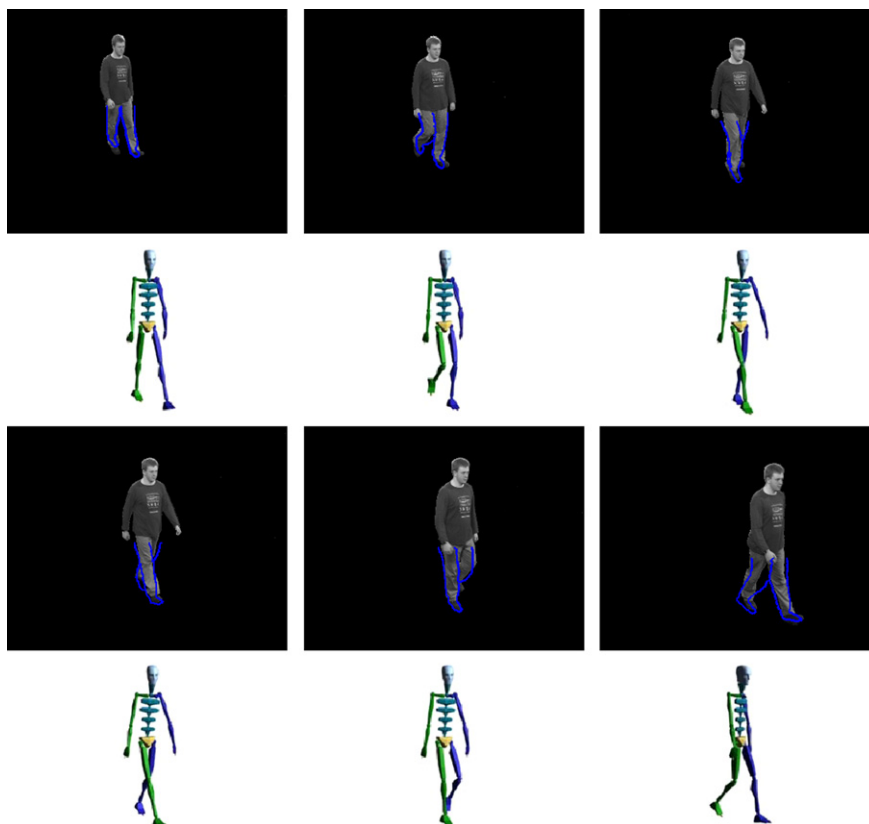


Fig. 7. Results of tracking a subject walking straight from an unseen view-point (46 frames total) for frames 3, 15, 21, 25, 35, and 44, respectively.

foreground pixels was then obtained. The mean position with added Gaussian white noise was then used to set each particle's offset values. Additionally, the scale factor of each particle was obtained by first estimating the variance of the  $y$  values of foreground pixels. A ratio between half the variance (with added Gaussian noise) and the bounding box height of the particles visual examples provides the estimated scaling factor.

### 6.1. Exemplars training data

To learn the motion model, a database that consisted of motion capture data (29 joint angles for the whole body) of a single subject only walking in a straight line ahead was used (down the  $z$ -axis). The total number of frames used was 734. Every frame in the training sequence was used, to generate exemplars as described in Section 3.3. It was found that the storage requirement for all 734 exemplars was small and totalled approximately 13 Mb of memory. Following this, an exemplar motion model was built as described in Section 4. The number of clusters in the motion model was set empirically to 30.

### 6.2. Tracking particles initialisation

For all the experiments presented in this paper, the initialisation for tracking was done in a semi-supervised manner. The user first specifies a small set of exemplars, usually

from a short training motion sequence (e.g. quarter cycle in a walking sequence). Given the input image's distance transform image, an exhaustive match is performed over all viewpoints and poses from this set of exemplars. This will provide a simultaneous estimation for the initial pose and viewpoint for the particles in the tracking system. This method was found to be more accurate and effective at locating the correct starting pose and viewpoint than a fully automated initialisation process involving an exhaustive search over all the viewpoints of every exemplar. It was found that such an exhaustive search was more susceptible to matching to wrong poses or viewpoints due to ambiguities in the visual information used. When tracking is performed, it was found that 200 particles were sufficient to successfully track the subject in all test sequences.

### 6.3. Experiment 1: simulated data, turning walk

The first experiment shows the tracking of a rendered 2D video sequence of the generic 3D model walking from an unseen angle. Additionally, the model is made to turn  $90^\circ$  at the middle of the sequence. It is important to note that although the 3D model used is similar to that for generating the training data, its underlying motion is completely unseen with novel movement dynamics. The results can be seen in Fig. 6. The first column (leftmost) shows the input image along with the best 2D edgels overlaid. The second column shows the 5 best particles in the

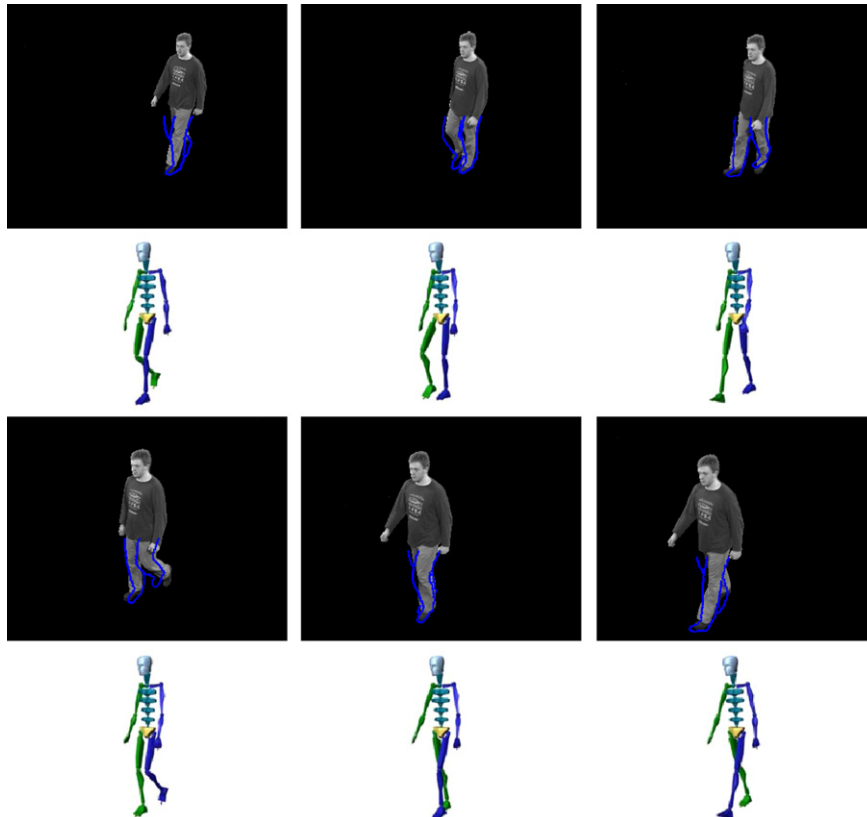


Fig. 8. Results of tracking a subject walking straight from an unseen view-point (46 frames) for frames 10, 18, 22, 32, 39, and 42.

cluster visualisation model. This allows us to see how the particles traverse across the different clusters during the tracking process. The third column shows the 3D skeleton rendered from the best viewpoint associated with the best particle. It should be noted that only the lower body (i.e., legs) is tracked. However, the 3D skeleton shown has correct upper body movements as well. The reason for this is because the 3D information in each exemplar is for the entire body. Thus, although only the legs are tracked, we have the corresponding 3D information for both the upper and lower body.

#### 6.4. Experiment 2: novel video data, straight walk

The purpose of the next experiments is to test the ability for the method to cope not only with unseen motion dynamics (style, speed, and length), but also unseen visual information. The subject used in these video sequences has nothing to do with the motion capture information used in training. The second experiment tests the ability of our method for handling real video sequences of a subject walking in a straight motion.

Two sequences of a subject walking in a straight line were captured from two novel viewpoints in a studio environment. Background subtraction and chroma thresholding was then performed on the sequences to extract the foreground object. Tracking was then performed using the particle filter with an equivalent of 200 particles. The results can be seen in Figs. 7 and 8 for each viewpoint respectively. It can be seen that the tracker is successful in determining the correct viewpoint as well as the 3D pose of the subject.

#### 6.5. Experiment 3: novel video data, turning walk

The third experiment tests the generalisation capability of our tracker for tracking video motion sequences of novel walking trajectories. A video of a subject performing a walking motion with a left turn in the middle of the sequence was captured. Again, background subtraction was also performed and tracking was performed using the same particle filter for the above experiment using 200 particles. The results can be seen in Fig. 9. We found that the tracker managed to dynamically track

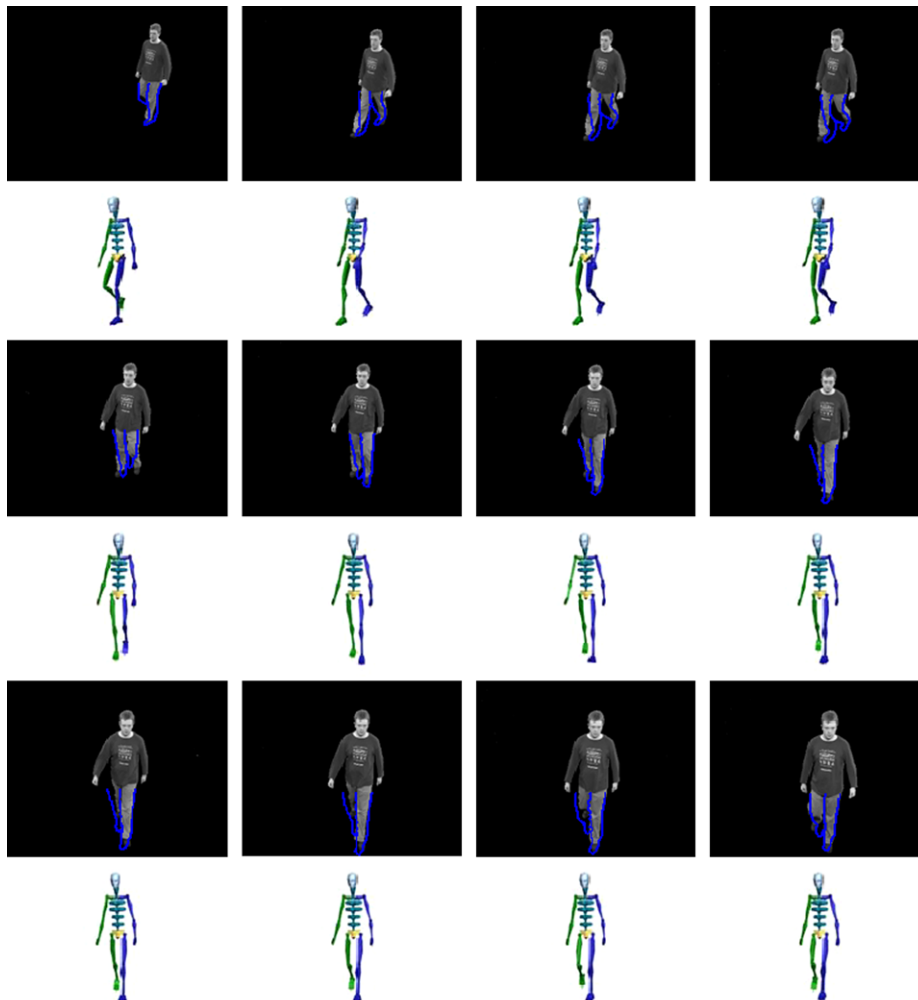


Fig. 9. Results of tracking a subject performing a turning left walking motion (46 frames) for frames 4, 15, 17, 19, 24, 26, 28, 31, 33, 35, 38, and 42.

the changes in 3D pose and viewpoints as the subject performed the motion. However, we note that the tracking from the frontal view is rather ambiguous and is sometimes difficult to determine the exact pose of the subject.

### 6.6. Performance

This section provides a more detailed analysis on the results of the experiments discussed in the previous section. In order to do this, ground truth data in the form of foot positions were first obtained. This was achieved by manually labelling the positions of the centre of the feet for every frame of every test sequence. However, it is noted that because feet can often get occluded from different angles, some amount of guesswork had to be carried out in order to label feet that could not be seen.

Additionally, for the purpose of error analysis, the foot positions for each exemplar were also calculated. This was achieved by separately identifying vertices belonging to the left and right foot on the original 3D model. To obtain the mean position of the foot (left or right), only polygons associated with the required foot were given a colour. All other polygons (including the other foot) were assigned a transparent material. The images were rendered from exactly the same virtual cameras used for obtaining the edgel information. Having obtained the image where the only object present is the foot, simple image processing can be used to extract the mean of the 2D foot position for a particular view and pose.

Using the above information, it is now possible to obtain the foot positions in the image of the tracked results. This was achieved by simply scaling and offsetting the foot positions in the same manner as for the edgels at each frame in the test sequences. The pixel distance from the ground truth position can then be calculated. The visualisation of the estimated and groundtruth foot positions is given in Fig. 10 for all four experiments. Additionally, the normalised mean feet pixel distance error curve was also obtained for each experiment. The error values in each curve are then sorted in descending order and the number of frames normalised to a percentage. This allows us to see the percentage of frames in a sequence with high error rates. We show these sorted curves, one for each experiment in Fig. 11.

Upon closer analysis, we discovered that large errors are mainly caused by the occlusion of one foot. It was found that the position of the visible foot was usually fairly accurate. A few frames after that, the error would reduce down again once both feet were visible again, showing a recovery from problems due to occlusion. Another source of problem arises from the multi-view edgels being sampled from a discrete number of views. An example is when the subject is at view-point between two of these discrete views. The tracking algorithm will match the input to the closest view-point in a particular exemplar. It is this difference between the actual and “closest-matching viewpoint” of an exemplar that causes errors in the estimation of the foot positions.

In terms of computational complexity, our method implemented in unoptimised Matlab takes about 1 s to

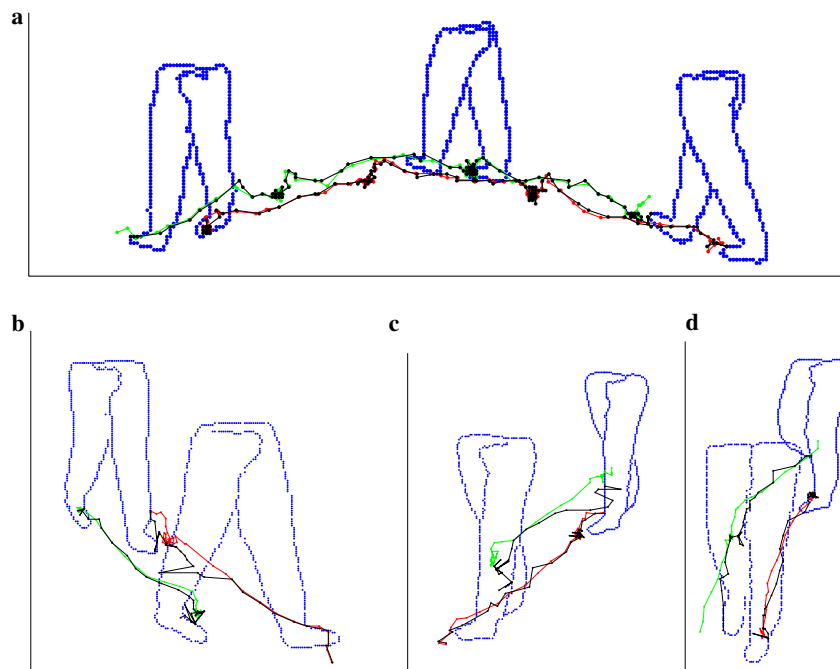


Fig. 10. The visualisation of the estimated (black lines) and groundtruth foot positions (green and red lines) and trajectories for the (a) rendered images for turning walk, (b) and (c) video data for straight walking, and (d) video data for turning walk. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this paper.)

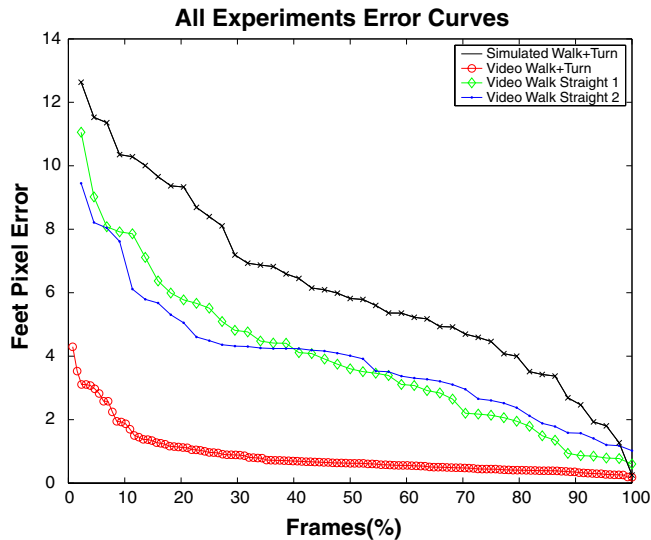


Fig. 11. The sorted error curves for all the experiments. See Section 6.6 for details.

process each frame using 200 particles on a 1 Ghz Pentium 3 processor. The resolution of each frame is  $320 \times 240$  in size. However, we predict that should this method be implemented in C++, a 10-fold gain in efficiency could be achieved.

## 7. Conclusions and future work

In this paper, we proposed a clustered exemplar-based model for performing viewpoint invariant tracking of the 3D motion of a human subject from a single camera. Each exemplar is associated with multiple view visual information of a person and the corresponding 3D skeletal pose. The visual information takes the form of edge pixels obtained from different viewpoints around the subject. The use of edge pixels allows us to exploit internal edges that are useful for differentiating between various ambiguous poses. The inclusion of multi-view information is important for two reasons: viewpoint invariance; and generalisation to novel motions.

The dynamics of the exemplars are modelled in two steps. First, 3D skeletal motions with similar movement are clustered. The inter-cluster and intra-cluster dynamics are modelled by grouping motion segments within each cluster into flow vectors.

Visual tracking of human motion is performed using a particle filter coupled to the dynamics of human movement represented by the exemplar-based model. Experiments were then performed using various sequences that tested the ability for the tracker to handle novel viewpoints, as well as motion sequences with novel trajectories. All experiments performed used only a single view and demonstrate that the exemplar-based models incorporating dynamics generalise to viewpoint invariant tracking of novel movements.

For future work, we will attempt to address the problem of modelling and tracking different types of motions (e.g., walking and running). A useful starting point would be

to draw from work by Arikan et al. [24] for methods that allows one to blend and transition between different types of motions. Additionally, integration of this method into a multi-camera system for more robust tracking will be investigated. The use of more cameras should provide more information that can be used to further constrain the tracking system, allowing for a less ambiguous estimation of the body pose and orientation. In terms of clustering, further analysis will be made to determine if any redundant exemplars and flow vectors within a cluster can be discarded. The result could be that only “salient” exemplars and flow vectors are kept in a particular cluster.

## References

- [1] L. Ren, G. Shakhnoroich, J. Hodgins, H. Pfister, P. Viola, Learning silhouette features for control of human motion, in: Proceedings of the SIGGRAPH 2004 Conference on Sketches and Applications, ACM Press, New York, 2004.
- [2] T. Heap, Learning deformable shape models for object tracking, Ph.D. thesis, School of Computer Studies, University of Leeds, UK (September 1997).
- [3] R. Bowden, T. Mitchell, M. Sarhadi, Reconstructing 3d pose and motion from a single camera view, in: J.N. Carter, M.S. Nixon (Eds.), Proceedings of the British Machine Vision Conference, vol. 2, University of Southampton, 1998, pp. 904–913.
- [4] C. Sminchisescu, A. Jepson, Generative modeling for continuous non-linearly embedded visual inference, in: Proceedings of the 21st International Conference on Machine Learning, Banff, Canada, 2004.
- [5] A. Rahimi, B. Recht, T. Darrell, Learning appearance manifolds from video, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, San Diego, CA, 2005.
- [6] A. Elgammal, C. Lee, Inferring 3d body pose from silhouettes using activity manifold learning, in: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, vol. 2, 2004, pp. 681–688.
- [7] G. Shakhnoroich, P. Viola, T. Darrell, Fast pose estimation with parameter-sensitive hashing, in: Proceedings of the Ninth IEEE International Conference on Computer Vision, 2003.
- [8] B. Stenger, A. Thayananthan, P.H.S. Torr, R. Cipolla, Hand pose estimation using hierarchical detection, in: International Workshop on Human–Computer Interaction, Prague, Czech Republic, 2004, pp. 105–116.
- [9] K. Toyama, A. Blake, Probabilistic tracking in a metric space, in: Proceedings of the ICCV, 2001, pp. 50–59.
- [10] G. Mori, J. Malik, Estimating human body configuration using shape context matching, in: Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark, 2002.
- [11] O. Jenkins, M. Mataric, Performance-derived behaviour vocabularies: data-driven acquisition of skills from motion, International Journal of Humanoid Robotics 1 (2) (2004) 237–288.
- [12] M. Dimitrijevic, V. Lepetit, P. Fua, Human body pose recognition using spatial-temporal templates, in: ICCV Workshop on Modelling People and Human Interaction, Beijing, China, 2005.
- [13] C. Sminchisescu, A. Kanaujia, Z. Li, D. Metaxas, Discriminative density propagation for 3d human motion estimation, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, San Diego, CA, 2005.
- [14] M. Brand, Shadow puppetry, in: Proceedings of the ICCV, 1999, pp. 1237–1244.
- [15] H. Sidenbladh, M.J. Black, L. Sigal, Implicit probabilistic models of human motion for synthesis and tracking, in: Proceedings of the ECCV (1), 2002, pp. 784–800.
- [16] N. Howe, M. Leventon, W. Freeman, Bayesian reconstructions of 3d human motion from single camera video, in: NIPS, 1999.

- [17] R. Urtasun, P. Fua, 3d human body tracking using deterministic motion models, in: Proceedings of the ECCV, 2004.
- [18] A. Agarwal, B. Triggs, Recovering 3d human pose from monocular images, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28 (1) (2006) 44–58.
- [19] D. Ramanan, D. Forsyth, Finding and tracking people from the bottom up, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2003.
- [20] L. Sigal, M. Isard, B. Sigelman, M. Black, Attractive people: assembling loose-limbed models using non-parametric belief propagation, in: *Advances in Neural Information Processing*, 2003, pp. 1539–1546.
- [21] M. Isard, A. Blake, Condensation—conditional density propagation for visual tracking, *International Journal of Computer Vision* 29 (1) (1998) 5–28.
- [22] P. Felzenszwalb, D. Hurrenlocher, Distance transforms of sampled functions, Technical Report TR2004-1963, Cornell Computing and Information Science (2004).
- [23] H. Barrow, J. Tenenbaum, R. Bolles, H. Wolf, Parametric correspondence and chamfer matching: two new techniques for image matching, in: Proceedings of Joint Conference on Artificial Intelligence, 1977, pp. 659–663.
- [24] O. Arikan, D. Forsyth, Interactive motion generation from examples, *ACM Transactions on Graphics* 21 (2002) 483–490.