Hierarchical Part-Based Human Body Pose Estimation

R. Navaratnam * A. Thayananthan * P. H. S. Torr † R. Cipolla *

* University of Cambridge
Department of Engineering
Cambridge, CB2 1PZ, UK
http://mi.eng.cam.ac.uk/∼cipolla

† Oxford Brookes Univeristy
Department of Computing
Oxford, OX33 1HX, UK
http://cms.brookes.ac.uk/staff/PhilipTorr/

Abstract

This paper addresses the problem of automatic detection and recovery of three-dimensional human body pose from monocular video sequences for HCI applications. We propose a new hierarchical part-based pose estimation method for the upper-body that efficiently searches the high dimensional articulation space. The body is treated as a collection of parts linked in a kinematic structure. Search for configurations of this collection is commenced from the most reliably detectable part. The rest of the parts are searched based on the detected locations of this anchor as they all are kinematically linked. Each part is represented by a set of 2D templates created from a 3D model, hence inherently encoding the 3D joint angles. The tree data structure is exploited to efficiently search through these templates. Multiple hypotheses are computed for each frame. By modelling these with a HMM, temporal coherence of body motion is exploited to find a smooth trajectory of articulation between frames using a modified Viterbi algorithm. Experimental results show that the proposed technique produces good estimates of the human 3D pose on a range of test videos in a cluttered environment.

1 Introduction

Many applications such as Human Computer Interaction (HCI), sign language recognition and avatar animation require pose estimation of the human body from video sequences. A system that estimates body pose for these applications should be able to self initialize (i.e. can recover when it loses track, for example, if the individual leaves the view), track independent of activity, identify the configuration of the body as accurately as possible and be computationally efficient. For convenience, such a system needs to be able to run in a standard PC with single (optical) camera.

Tracking human body articulation is a difficult problem due to the high dimensionality of the state space and the inherent ambiguity that arises from using 2D image features to estimate 3D pose parameters. Traditionally, tracking involves searching for the current pose based on the pose estimate from the previous frame guided by a dynamic model. Single hypothesis tracking such as the Kalman filter [16] invariably fails due to the ambiguity involved. Particle filters have been used in the past to overcome ambiguity by maintaining multiple hypotheses [4]. However, a large number of particles is needed

BMVC 2005 doi:10.5244/C.19.47

for tracking the full range of human motion. The main disadvantage of these traditional tracking paradigms is that they can neither self-initialize nor recover from a loss of track.

One way of addressing the above problems is to treat tracking as object detection in every frame. The problem of pose estimation is decoupled from temporal dependency and hence the problem of initialization and recovery are solved within this *tracking as detection* framework. The temporal information is only used to smooth the motion and to overcome ambiguous situations. This clearly divides the tracking problem into two distinct problems: (1) Estimate a set of possible configurations from a single frame (Detection) (2) Combine these configurations from each frame to obtain smooth multiple trajectories (Tracking).

The techniques for solving the detection problem can be divided into two categories namely top-down, full- template matching approaches and bottom-up part-based approaches. Gavrila's pedestrian detection [7] and Stenger et al's [21] hand tracking frameworks are typical examples of top-down full template matching methods. A number of 2D templates are created which cover the entire state space and are organized into a template hierarchy to facilitate a coarse to fine search strategy. However, the number of templates needed to find the correct object pose increases exponentially with the number of dimensions and this is impractical with more than 7 or 8 dimensions.

Bottom-up approaches handle the high dimensionality efficiently by focusing on parts. These part-based approaches try to identify body parts individually (in contrast to a full templates) in an image and then assemble a configuration that has maximum support from image observations. The parts are detected independently of each other based on generic rectangular shape templates and simple appearance models. Possible body configurations are assembled from the detected collection of parts using Data-Driven Markov Chain Monte Carlo framework [12] or versions of belief propagation [6, 15, 19] or simple heuristics [13]. The spatial relationship between parts is lost when they are detected individually which potentially leads to a large number of false positives making the problem of re-assembling them into meaningful configurations very complex.

In this paper we present a 'hierarchical part-based' detection scheme with the aim of combining the best of both top down and part based methods. The main idea is to view the whole body template as a hierarchy of parts. First we search for the root part. Then we explore each branch of this hierarchy sequentially by searching for a child part based on identified locations of the parent part. We avoid the difficult problem of re-assembling the parts by retaining the kinematic links between parts while searching. At the same time we circumvent the curse of dimensionality by essentially searching for one part at a time. Moreover, unlike other part-based approaches [6], our part-templates inherently encode the 3D pose information, thus we are able to extract 3D articulation parameters directly.

In a pure detection-based approach, the pose with the highest confidence in each frame can be selected as the best configuration. However this does not guarantee a smooth transition of pose between frames. In the real world, a body part can physically move only to an adjacent location in space from one instance to the next. In terms of pose parameters, this implies smooth transitions, or '*temporal coherence*'. Restricting the motion being tracked to particular actions [11, 16, 17] or explicitly modelling the manifolds of pose parameters from training data [2, 5, 18, 23] would not only reduce the dimensionality of state space, but also provide a means to smooth the motion between frames. But these approaches are difficult to generalize to arbitrary movements because one would require a large amount of data to model/learn all possible motions.

In our framework, the temporal coherence is imposed by modelling the detected configurations of each frame as states in a Hidden Markov Model (HMM) and applying the *Viterbi* algorithm to estimate a track of smooth pose transitions (see section 3). A simple dynamical model that penalizes large pose transitions is assumed. This framework inherently propagates multiple hypotheses for body configurations between frames.

The next section explains how we take advantage of the hierarchical arrangement of body parts to reduce the number of templates required for pose estimation while extracting accurate 3D pose information.

2 Hierarchical part-based detection

In template-based approach one needs to search over a large number of templates to estimate all possible poses. For example, if the articulation parameters are discretised at 10 degree intervals, over 10^{10} templates are needed to represent all possible poses of the upper body. We exploit the hierarchical nature of the kinematic structure to reduce this.

Consider the following 'parts' or 'combination of parts': (a) head and torso, (b) upper arms and (c) lower arms and hand. For a front-facing upper body, the number of degrees of freedom for each of these parts is 2 for scale and rotation of torso, 3 for two articulations and scale of upper arm and 3 for two articulations and scale of lower arm. Hence the number of templates needed to represent the state space for each of these are considerably less than for a complete upper body (less than 10 000 in total).

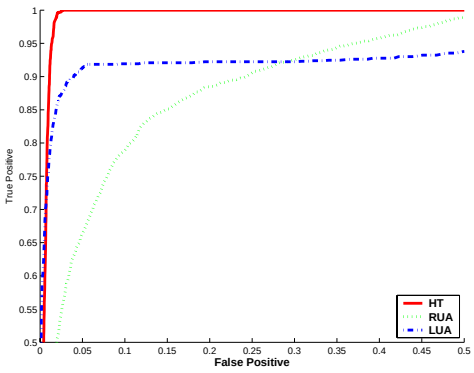

Figure 1: **ROC curves for individual detectors.** *Illustrated are the ROC curves of three individual part detectors: Head-and-torso (red solid), Right-upper-arm (green dotted) and Right-lower-arm blue (dashdot). It is evident from these that the head-and-torso detector is the most reliable one.*

Now the challenge is to combine these representations in a meaningful way and extract the pose of the upper body. One approach is to detect individual parts separately and assemble configurations that are supported by the image [6]. Instead, we propose to initially search for the head-and-torso part and then search for other parts in a breadth first order of the kinematic hierarchy.

There are many reasons to search for the head-and-torso part first. In HCI applications, the user generally faces the camera. So it is reasonable to assume that the head and torso are visible most of the time. Moreover, the head-and-torso part generally occupies a larger image region compared to other parts when the whole upper body is captured in the image. Hence it contains more information for that part detector to exploit. The reliability of the head-and-torso part compared to the upper-arm and lower-arm-and-hand parts is illustrated through ROC curves in figure 1. Searching for parts in this hierarchical manner has precedent in the past literature.

[1] HT - head and torso, RUA, LUA - right and left upper arm, RLA, LLA - right and left lower arm.

In [15], Ramanan and Forsyth initially detect the possible torso locations to narrow the search for other body parts. Gavrilla [8] takes a similar approach, which he terms as 'search space decomposition', where he localizes the torso using color cues and uses this information to constrain the search for the limbs.

2.1 Formulation

Now we formulate a posterior for configurations detected in an image. The state vector $\mathbf{x}$ of body configurations is a concatenation of sub-state vectors of the body parts, $\mathbf{x} = \{x_{HT}, x_{RUA}, x_{LUA}, x_{RLA}, x_{LLA}\} = \{x_i\}$. The sub-state vectors contain articulation and positional parameters. The posterior probability of a pose $(\mathbf{x})$ given the image observations $(\mathbf{z})$ is formulated as,

$$p(\mathbf{x}|\mathbf{z}) = \frac{p(x_{HT})p(x_{RUA}|x_{HT})p(x_{LUA}|x_{HT})p(x_{RLA}|x_{RUA})p(x_{LLA}|x_{LUA})p(\mathbf{z}|\{x_i\})}{p(\mathbf{z})} \quad (1a)$$

$$\propto \prod_i p(x_i|x_{parent(i)}) \prod_i p(\mathbf{z}|x_i). \quad (1b)$$

The prior $p(x_{HT})$ is taken to be uniform over all articulation (within allowed range of joint angles) and positional state parameters of x_{HT}. The conditional prior $p(x_{RUA}|x_{HT})$ is also modelled as uniform distribution. Similar arguments/assumptions are made for the other priors, $p(x_{LUA}|x_{HT}), p(x_{RLA}|x_{RUA})$ and $p(x_{LLA}|x_{LUA})$. Since detection in each frameis performed without prior knowledge from previous frames, it is reasonable to assume that all poses (with joint angles in valid range) are equally possible.

The likelihood scores obtained during the template matching of individual body parts are independent of each other. So we approximate the likelihood as $p(\mathbf{z}|\{x_i\}) \propto \prod_i p(\mathbf{z}|x_i)$ where $p(\mathbf{z}|x_i)$ is the likelihood of a single part. This is a standard approximation adopted by many [6, 13] in the past.

The image observations are based on the chamfer score $\mathbf{z}_{cham}$ and the appearance score $\mathbf{z}_{app}$. The joint likelihood of $\mathbf{z} = (\mathbf{z}_{cham}, \mathbf{z}_{app})$ is approximated as

$$p(\mathbf{z}|x_i) = p(\mathbf{z}_{cham}|x_i)p(\mathbf{z}_{app}|x_i) \quad (2)$$

assuming that the observations are independent. The next section describes how these likelihoods are computed.

2.2 Template search

Even though the high dimensionality of the body can be effectively handled by decomposing into parts, the templates[2] that represent the parts still need to be searched for efficiently. Template trees provide an efficient method to search with a large number of templates. By constructing a hierarchy of templates (for each part separately, i.e. 5 template trees), where each node is a prototype for all its children, and searching the children

[1]The term '$parent(i)$' represent the parent of part i, for example $parent(RUA) = HT$.

[2]Templates are created from projections of a 3D model built from 18 truncated quadrics in a similar manner to [20]. The dimensions of this model are generic and not adopted to fit each person

only if the matching error for the parent is below a threshold leads to an efficient algorithm which is orders of magnitude faster than a brute force search of all templates [7, 21].

The templates in the first level are matched at coarse grid locations of a given input image. Then, only the sub-trees under templates that have given matching error below a threshold are further explored. During this, the templates at each node being investigated are matched at an ever decreasing neighborhood of the parent's matched locations. Hence, subsequent templates are tested at locations where they are highly likely to match, thus reducing the total number of template evaluations needed to search through the complete set. Each template is matched to image based on shape and appearance similarity.

Edge features: Similarity between image edge features and template edge features is measured using *chamfer distance* [1]. A robust measure is obtained by dividing the image edges into 8 groups based on their orientation and calculating the *distance transform* (DT) for each of them separately. The template points with corresponding orientation are matched with the relevant DT image. The similarity measure of a template is found from the mean sum (d_{cham}) of individual costs from all edge groups as follows,

$$p(\mathbf{z}_{cham}|x_i) = \frac{1}{Z_{cham}} \exp(-\lambda_{cham} d_{cham}). \tag{3}$$

In the above formulation, Z_{cham} is the partition function and λ_{cham} is the weight assigned for the edge-based similarity.

Appearance features: In a continuous sequence of images, it can be safely assumed that the person will not change suddenly in appearance. Hence, each image carries important information about that person in terms of his/her appearance such as clothing and visible skin areas. This is exploited by learning appearance models for each body part using RGB histograms[1] from the initial 30 frames of a video sequence. Using these, given an RGB triplet, an appropriately weighted confidence measure can be obtained. The input image is segmented using mean shift [3]. The probability of each region belonging to a particular part is computed based on the collective probability of all the pixels in that region. If that probability is above a threshold then the whole region[2] is assigned to that body part. Thus a binary image for each part is extracted from the input image. Integral and row-sum images are computed from these binary images and a correlation type similarity score is used to match each template efficiently,

$$p(\mathbf{z}_{app}|x_i) = \frac{1}{Z_{app}} \exp(-\lambda_{app}(1 - d_{app})). \tag{4}$$

The appearance similarity score, d_{app}, ranges between 1 and -1. The higher the value the better the similarity, hence the term $(1 - d_{app})$ in the exponential. In the above formulation, Z_{app} is the partition function and λ_{app} is the weight assigned for the appearance-based similarity.

These (Eq.3 and Eq.4) are rather simplistic and computationally inexpensive approximations for feature likelihoods. We obtained reasonable detection rates and relatively small classification error with these approximations. The negative log likelihood is

$$-\log(p(\mathbf{z}|x_i)) = \lambda_{cham} d_{cham} + \lambda_{app}(1 - d_{app}) + K. \tag{5}$$

[1] We used 32 bins in each dimension for the joint histogram in RGB [10].

[2] This is better than assuming each pixel is independent and classifying them individually.

The weight factors, λ_{cham} and λ_{app}, and the constant K are computed empirically such that the maximum likelihood classification error is minimized for all templates[1].

The posterior computed for each body configuration found through template matching in each frame is a sample of the posterior distribution. We find the modes of this distribution using a simple heuristic based on the distance between the configurations in the state space and their posterior probability.

Each frame has been dealt with in isolation until now. We describe a tracking paradigm for a sequence of frames in the next section.

3 Enforcing Temporal Coherence

As noted in section 1, temporal coherence in the movements of body parts translates to smooth transitions in configuration space. Sets of configurations are detected for each image in a video sequence and an optimal path through these needs to be found such that the articulation is smooth. This can be easily found using a Hidden Markov Model (HMM)[2]. The detected mode-configurations of each frame are treated as the states of the HMM at a time instant. Since we do not assume any strict dynamic model, the transition between these states from one time instant to next is modelled by single Gaussian on each parameter of the state space. The mean for the distributions are equal to the parameters of state at $(t-1)$ and the standard deviation depends on the partition size of the state space used in creating the templates. The probability of each configuration at the initial frame is assumed to be uniform. The Viterbi[2] algorithm is used to find the best path that enforces smooth pose transition between frames. At each iteration step of the Viterbi algorithm, multiple hypothesis for paths are propagated. This is essential as there is depth ambiguity of body parts in monocular sequences.

4 Experiments

Some experiments were performed to assess how reliably the head-and-torso part can be detected compared to other parts. As shown in Fig.2 and Table 1, lower arm part produced many more false positive matches than the head-and-torso part. Moreover, in the examples illustrated in Fig.2, the right lower arm was not detected at all at the given threshold. Compared to a simple torso part, the combined head-and-torso part is much more reliably detected. Similarly, the lower-arm-and-hand combined part produces less classification error compared to a simple lower-arm part.

Note that the number of detections for the same threshold does not vary much across different images for the head-and-torso part. For example, in table 1, for a low threshold the numbers of head-and-torso parts detected in two different images are 61 and 56 whereas the detections for the lower arm parts are 44,993 and 13,199. Hence the same threshold can be safely used with head-and-torso part detection for various image sequences.

We tested the proposed approach for pose estimation in many video sequences. Each video sequence contained approximately 500 frames. Since we did not have ground-truth

[1] A principled approach to compute these might be to use the PDF Projection Theorem as proposed in [22]

[2] For a detailed explanation of HMM and Viterbi algorithm, refer the tutorial by Rabiner [14]

Threshold	Low (0.81)				High (0.84)			
Part	HT	LA	HT	LA	HT	LA	HT	LA
Image No.	2(a)	2(e)	2(b)	2(f)	2(c)	2(g)	2(d)	2(h)
Locations	61	44 993	56	13199	4	5 616	1	849

Table 1: **Templates and detected locations:**
Two different parts - namely, head-and-torso (HT) and lower arm (LA) - are searched for independently and the number of detections are tabulated. Their locations on the image are illustrated in Fig.2. Consider the first and second rows of the above table. The head-and-torso part has been detected at 61 locations whereas the lower arm part has been detected at over 40 000 locations for the same likelihood threshold. Moreover, the lower arm part failed to detect the right lower arm in the examples given in Fig.2

(a) (b) (c) (d) (e) (f) (g) (h)

Figure 2: **Templates and detected locations:** *Head-and-torso part (a,b,c,d) and lower arm part (e,f,g,h) at their detected locations are overlaid (in white) on the image as explained in the Table 1. Figures a,b,e and f are for a low threshold and c,d,g,and h are for a high threshold of likelihood. As illustrated, the head-and-torso part can be more reliably detected compared to lower arm part.*

data, the identified configurations were visually evaluated with some tolerance for depth ambiguity.

4.1 Detection in a single frame

Figure 3 shows examples of poses detected based on the approach of section 2. The results illustrated here are obtained without exploiting any temporal information.

In the sub-figures (b),(d),(f),(g),and (h) of figure 3, the upper arm of the person in the image is fully occluded and in sub-figures (b),(f) and (g), the lower arm is pointing directly at the camera. These poses are correctly identified because the parts are searched for in the hierarchical order of the kinematic structure. These would have been very difficult to handle in other part-based strategies as proposed in [6, 12] and [13]

In cases where the highest ranked pose (see section 2.2) did not match well the pose of the human in the image, the best pose is typically found within the first 20 highest ranked poses (in 98% of such instances) as illustrated in figure 4. This suggests that the correct pose will possibly be identified by considering the temporal coherence of articulation between frames.

4.2 Tracking

When using an HMM and applying the Viterbi algorithm as explained in section 3, as one backtracks (Fig.6) from time instant $t = T$, typically all trajectories collapse to a single path for $t < T - 10$, as also observed in [9]. The Viterbi algorithm is designed to give only

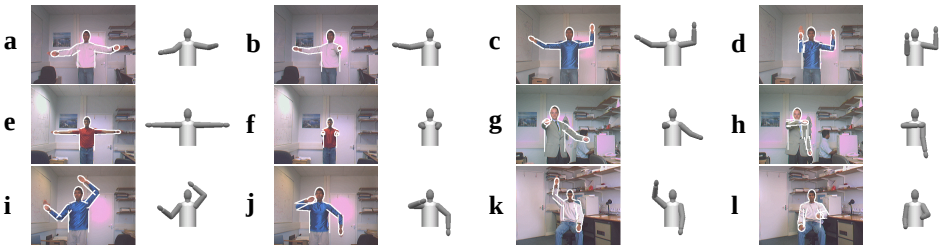

Figure 3: **Pose Detection:** *Results of detection-based pose estimation in individual frames from 6 different sequences. On average, the body pose was correctly estimated in 80% of the frames in our experiments. In cases where the highest ranked pose did not match well the pose of the human in the image, the best pose is typically found within the first 20 highest ranked poses in 98% of such instances.*

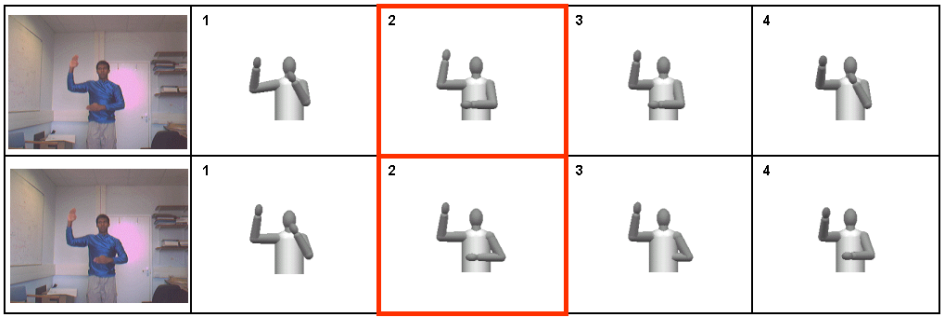

Figure 4: **Ranked Samples:** *The ranked samples from detection-based pose estimation. The highlighted samples are better estimation of pose of the human in the image compared to the first ranked configuration. The examples shown in here are frames 9 and 14 in the sequence at figure 5.*

one best path when backtracked. This can be useful for batch processing a sequence of frames off-line. A useful modification of this method for real-time pose estimation could be a time delayed output, i.e. the system could output an estimate of pose for frame $t-\tau$ ($\tau \geq 10$) by backtracking from the input frame t. We follow this approach to stay within the tracking paradigm.

In this approach, application of the Viterbi algorithm takes only a fixed amount of computation for each new frame. As each new frame is processed, paths need to be propagated only between the current (t) and the previous $(t-1)$ frames. The best configuration for $(t-\tau)^{th}$ frame is then computed by backtracking from the t^{th} frame.

Figure 5 illustrates how the temporal coherence helped to find smooth pose transitions between frames where the pose detection at each frame separately produced poor results. The two failed situations shown in figure 4 are frames 490 and 495 in the sequence at figure 5. In that sequence during simple pose detection, the body pose was misidentified in frames between 485 and 495. This was rectified by imposing temporal coherence.

On average this method produced 90% correct identification of full upper-body pose. In 95% of the frames the pose was correct except for one arm. Our implementation performed at a frame rate of 0.7Hz on a 2.4 GHz Pentium^{TM} 4 machine. We are currently

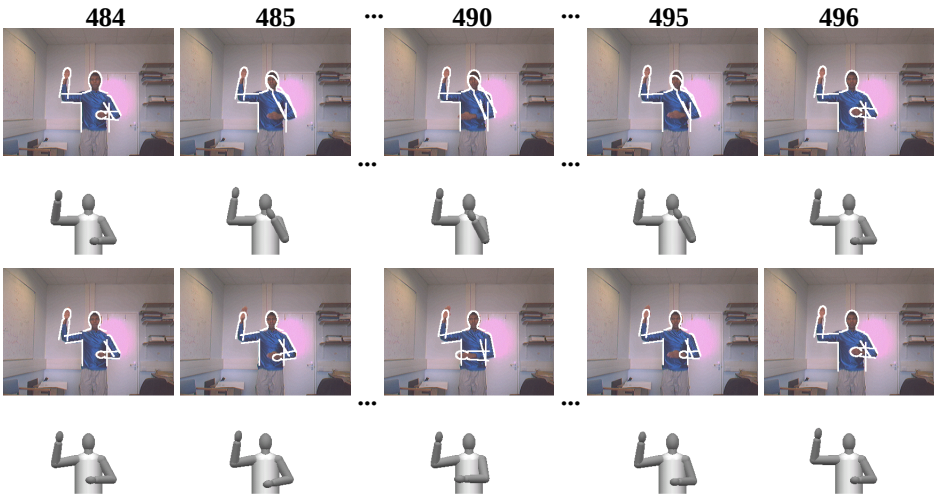

Figure 5: **Applying the Viterbi algorithm helps to recover better pose estimation.** *The first tow rows of images show the best pose estimate from pure detection in each frame in a sequence of 500 frames. The bottom two rows of images show the pose estimation obtained by backtracking as explained in section 4.2. As illustrated, the wrongly identified poses in frames between 485 and 495 are correctly estimated by exploiting temporal coherence.*

optimizing the implementation in order to enable to track in realtime.

5 Conclusions

In this paper we have presented a viable method for human body tracking and pose estimation for HCI applications. A hierarchical part-based approach that extracts 3D articulation parameters using templates has been introduced. As a result of the hierarchical search, some partially occluded poses were identified in this approach. Though only results from upper-body pose estimation are presented here, the approach can be easily extended to apply for full body.

Some improvements are possible in this framework. Imposing joint endstop constraints would reduce false positive configurations. In very long input sequences, illumination variation starts to degrade the appearance-based similarity measure. This could be remedied by updating the appearance online from reliably detected poses in occasional frames.

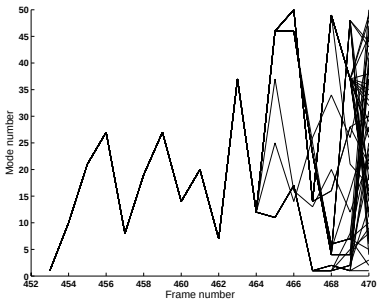

Figure 6: **Backtracked paths using the Viterbi algorithm:** *This is an example of all the backtracked paths from the very last frame in a sequence. Note how all the trajectories collapse to a single path within 10 frames from the end.*

Simple gestures could be recognized by assigning semantics to regions of articulation space. If only a fixed set of gestures are to be tracked, then the dynamics of each gesture and transitions between them could be used to impose constraints on pose further reducing the false positives.

Acknowledgments

The authors would like to thank Toshiba Corporation, Gates Cambridge Trusts and ORS for their support.

References

[1] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. *Proc. 5th. Int. Joint Conf. in AI*, 1977.

[2] M. Brand. Shadow puppetry. In *ICCV*, 1999.

[3] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *PAMI*, 2002.

[4] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *CVPR*, 2000.

[5] R. Fablet and M.J. Black. Automatic detection and tracking of human motion with a view-based representation. In *ECCV*, 2002.

[6] P.F. Felzenszwalb and D.P. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 2005.

[7] D. M. Gavrila. Pedestrian detection from a moving vehicle. In *ECCV*, 2000.

[8] D. M. Gavrila and L. S. Davis. 3-D model-based tracking of humans in action: a multi-view approach. In *CVPR*, 1996.

[9] M. Isard and A. Blake. A smoothing filter for condensation. In *ECCV*, 1998.

[10] M.J. Jones and J.M. Rehg. Statistical color models with application to skin detection. *IJCV*, 2002.

[11] X. Lan and D. P. Huttenlocher. A unified spatio-temporal articulated model for tracking. In *CVPR*, 2004.

[12] M.W. Lee and I. Cohen. Human upper body pose estimation in static images. In *ECCV*, 2004.

[13] G. Mori, X. Ren, A. A. Efros, and J. Malik. Recovering human body configurations: Combining segmentation and recognition. In *CVPR*, 2004.

[14] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 1989.

[15] D. Ramanan and D.A. Forsyth. Automatic annotation of everyday movements. In *NIPS*. 2004.

[16] K. Rohr. Towards model-based recognition of human movements in image sequences. *CVGIP: Image Understanding*, 1994.

[17] H. Sidenbladh, M. J. Black, and D. J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *ECCV*, 2000.

[18] H. Sidenbladh, M.J. Black, and L. Sigal. Implicit probabilistic models of human motion for synthesis and tracking. In *ECCV*, 2002.

[19] L. Sigal, M. Isard, B. H. Sigelman, and M. J. Black. Attractive people: Assembling loose-limbed models using non-parametric belief propagation. In *NIPS*. 2004.

[20] B. Stenger, P. R. S. Mendonça, and R. Cipolla. Model based 3D tracking of an articulated hand. In *CVPR*, 2001.

[21] B. Stenger, A. Thayananthan, P. H. S. Torr, and R. Cipolla. Filtering using a tree-based estimator. In *ICCV*, 2003.

[22] A. Thayananthan, R. Navaratnam, P. H. S. Torr, and R. Cipolla. Likelihood models for template matching using the pdf projection theorem. In *BMVC*, 2004.

[23] R. Urtasun and P. Fua. 3d human body tracking using deterministic temporal motion models. In *ECCV*, 2004.