

# Markerless Motion Capture from Monocular Videos

Vishal Mamania Appu Shaji Sharat Chandran  
Department of Computer Science & Engineering  
Indian Institute of Technology Bombay  
Mumbai, India 400 076  
{vishalm, appu, sharat}@cse.iitb.ac.in

## Abstract

*We present a method to determine the 3D spatial locations of joints of a human body from a monocular video sequence of a Bharatanatyam dance. The proposed method uses domain specific knowledge to track major joints of the human in motion from the two dimensional input data. We then make use of various physical and motion constraints regarding the human body to construct a set of feasible 3D poses. A dynamic programming based method is used to find an optimal sequence of feasible poses that represents the original motion in the video.*

## 1. Introduction

Markerless motion capture requires solutions to two problems; a tracking problem of identifying and disambiguating individual body parts from the rest of the image, and a reconstruction problem of estimating the 3D pose of the figure from 2D data. The challenge in tracking is to deal with background clutter and ambiguities in image matching, while the challenge in reconstruction is to compensate for the loss of 3D information that happens during recording. *In this work, we have developed motion capture techniques for a specific domain of Bharatanatyam dance, a classical dance form of India.*

There are two major human motion tracking methods: multi-view vision, where more than one camera is present, and monocular vision where only a single camera is used. *We try to tackle the harder problem using only a single camera.*

*Tracking of the human body in two-dimensional space is a well studied topic. However, tracking of individual body parts remains an ill-conditioned problem. Various hurdles haunting the problem include irregular shape of the human body, self-occlusion, clothings and makeup, shadows, and a high number of degrees of freedom (DOFs). Because of these constraints, it is currently not possible to track differ-*

*ent human body parts from a video automatically in a reliable manner. Hence the general problem remains largely unsolved. However, it is possible to solve the problem for specific applications using domain specific knowledge. We develop a semiautomatic method based on the uniformity of traditional dress of Bharatanatyam to track the body parts using skin color detection. The tracked data thus obtained is not always accurate and manual intervention is required in some cases.*

The main challenge in the *reconstruction* of articulated body motion is the large number of degrees of freedom to be recovered. A realistic articulated motion of the human body usually has at least 28 degrees of freedom. Search algorithms – deterministic or stochastic – that search such a space without constraints, fall foul of exponential computational complexity. *This paper uses additional constraints for solving the problem in realistic time.* We make use of constraints in the form of prior assumptions, motion estimation techniques, and view constraint restriction to break down the problem to a tractable algorithm.

More specifically, since our input comprises of data from a single camera, there exist fundamental ambiguities in the reconstruction of the 3D pose. The well known reflective ambiguity under orthographic projection results in a pair of solutions for the rotation of a single link out of the image plane. Once the actual length of each link is known, these ambiguities reduce to twofold *forward-backward flipping* ambiguities. The full model thus has  $2^{\#links}$  possible solutions. We make use of simple inverse kinematics to systematically generate the complete set of such configurations and hence to investigate the full set of associated cost minima. A dynamic programming algorithm is proposed to traverse this configuration tree and pick one of the most likely motions. More scene constraints are introduced so as to prune inconsistent configuration and thereby speeding up the search.

The rest of the paper is organized as follows. Section 2 summarizes some of the major work in the area of motion capture. Section 3 looks at the tracking aspect of the problem.

In Section 4, we propose a graph based algorithm which is augmented with probabilistic model for reconstructing 3D model from tracked data. We present our results in Section 5. Section 6 concludes the paper.

## 2. Related Work

Previous work in the field of motion capture has been mainly dependent on cues like markers or multiple views, while little work has been done in the field of single view markerless motion capture.

Tracking which forms an important ingredient of our method has been a well studied topic. Blob trackers, contour trackers and optical flow based tracking methods are most widely used tracking techniques. Blob trackers [22] extract low level information like color and pixel intensity. This information is subsequently grouped or interpreted according to the higher level knowledge about the scene. Our approach is similar to one used in [22]. The optical flow based algorithms [3, 17] extract a dense velocity field from an image sequence assuming that image intensity is conserved during the displacement. This conservation law is expressed by a spatio-temporal differential equation which is solved under additional constraints of different form. The feature based techniques [18] extract local regions of interest (features) from the images and identify the corresponding features in each image of the sequence. Contour-based object tracking [11, 2] requires object detection only once. Tracking is performed by finding the object contour given an initial contour from the previous frame.

An extensive survey of human motion capture techniques has been done by [14], which has references to more than 130 major publications in the field. Considerable amount of work has already been carried out for motion capture system involving multiple cameras. [12] generates 3D voxel data from multiple cameras placed at strategic locations to estimate pose. [1] uses a learning based method for recovering 3D human body pose from single images and monocular image sequences. They recover pose by direct non-linear regression against shape descriptor vectors extracted automatically from image silhouettes, whereas [7][13] use silhouettes generated from multiple views to track an articulated body in 3D.

[6] introduces an interactive system which combines constraints on 3D motion with input from a human operator to reconstruct sequences in 3D. They use an iterative batch algorithm which estimates the maximum a posteriori trajectory based on 2D measurements subject to a number of constraints, like kinematic constraints, joint angle limits, dynamic smoothing, and intermediate frames specified by the user. [15] makes use of a motion library to resolve the depth ambiguity in recovering the 3D configurations from



Figure 1. Some examples of the traditional Bharatanatyam dress.

2D features. [3] uses exponential maps and twist motions to extract 3D human configurations from a single-camera video sequence.

The weakness of kinematic constraint in monocular tracking can be addressed by using dynamic models to constrain the motion, and complex statistical methods to jointly represent the ambiguity in registration and reconstruction[20]. [19] uses particle filtering with important sampling based on either a learned walking model or a database of motion snippets, to focus search in the neighborhood of known trajectory pathways. [8] proposes an annealing framework in a multiple camera settings. During annealing, the search for parameters is driven by noise proportional with their individual variances.

One of the most major work of recovering structure from motion is the *factorization method* developed by [21]. However this method assumes orthography, and is applicable only for rigid body. Considerable amount of subsequent work has been done to extend factorization method to non rigid and non-orthographic cases. [16][5].

## 3. Tracking

In our work, we have tackled the tracking problem for the domain of Bharatanatyam, a classical dance form of India. We make use of specific information about the costume of the dancer.

The traditional dress worn by the dancer covers her entire body except the face, forearms, and the feet. Fig.1 shows a couple of examples of such dress. One important feature of the dress is the golden belt (seen in color print out) around the waist region. This belt is a part of the traditional dress and is always present.

In our implementation, we also made some standard assumptions such as only a single person, who is always in the view of camera, is present in the scene. The background is almost static and the camera is assumed to be stationary. Severe lighting changes are prohibited. These assump-

tions make the task of background subtraction easy. The distance between camera and the dancer is large. Thus we are justified in assuming the orthographic projection for reconstruction phase. We next discuss the human model used and selection and tracking of key body features.

### 3.1. Human Model

We can use a modelless approach for tracking in which case the tracking is done solely on the basis of shape and silhouettes, or we can use a 2D or 3D human model. Advantages of using a human model are that the state of the system at any point of time can be easily stored and accessed, and also the incremental addition of information becomes easy.

Volumetric 3D model would be a weak model as we use monocular video sequences. Using silhouettes makes it difficult to handle self-occluding body parts, especially arms. Stick figure representation, where the major joints are represented as points and the bones connecting them as lines, as shown in Fig. 2(a), can very well model the human body in 3D. We use the manipulated 2D data to feed the model. In our case, we are restricting our model to the upper part of the body. Fig. 2(b) shows the model we use.

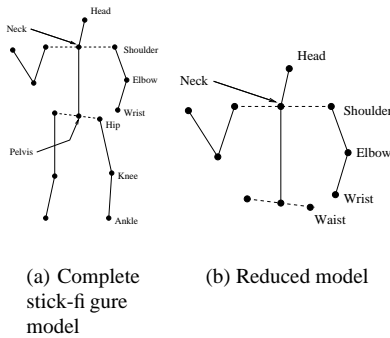


Figure 2. The human model's stick figure representation.

### 3.2. Key features

We mainly use *skin color model* to locate important features. We track the location of head, neck, shoulders, wrists, and belly to an acceptable level of accuracy using this model. The position of elbows and shoulders are approximated by making use of anthropometric information. It should be noted that the end effectors have the highest priority. End effectors are the end points of human body parts like arms, legs, etc. Examples are head, hands, and feet. The reason for giving a high priority to the end effectors is that the configuration of intermediate body parts can be approximately calculated from the end effector configuration, while the reverse is not true.

### 3.3. Feature Tracking

We use the skin color model similar to the one used by [9] for obtaining skin color regions. After we get the skin regions, we label the detected regions as the corresponding parts of the body. Incidentally the golden belt around the waist of the dancer closely resembles the skin color and is categorized as skin region. The output of skin detection is post-processed by morphological operations which produces the blobs of skin regions. If the corresponding body parts are well apart, these blobs would be separated. Each of these blobs can then be approximated as an ellipse. The endpoints of the major axis of each ellipse gives the endpoints (joints) of the corresponding link.

Problem occurs when the blobs get broken or merged. Blobs may break because of bad image processing, while they may merge because of proximity or occlusion. Broken blobs can be reunited while fitting ellipses by making use of local proximity information. Merged blobs can be separated by keeping track of orientation of the merging ellipses. If the major axes of both the occluding ellipses coincide, it is very difficult to separate them apart. However, this is a rare event in case of Bharatanatyam dance.

Using the above technique, we obtain the positions of head, neck, elbows, wrists, and waist. However, the position of the shoulders can not be obtained using this method. To locate the shoulders, we use a simple heuristic. It is observed that in most of the cases, except when the body is tilted, the position of the shoulders is exactly above the waist region endpoints and in horizontal line with the lower end of the neck. Using this heuristic, we can estimate the position of shoulders too, thus filling the entire human model as desired. The tracked data obtained by the above method is not very accurate. Hence we also need some manual intervention. In addition, we make use of a fixed-lag Kalman smoother [10] to filter out the erroneous data.

## 4. Reconstruction

After we get the 2D positions of all the joints, the next task is to estimate the 3D information that is lost during recording. Since we assume orthography for input data, we make use of foreshortening as a clue coupled with some additional heuristics to retrieve the lost information.

To make the problem simpler and tractable, we make use of the following assumptions.

1. The camera is stationary and calibrated.
2. The camera is sufficiently away from the subject such that we can safely assume orthographic projection. In our test data, the camera is placed at around 10 meters away from the subject.

3. The initial pose of the subject is also known.

### 4.1. Estimating absolute depths

In order to use foreshortening, we need to know the actual 3D lengths of all the links. This can either be done manually or using anthropometric data. We have tried to find the actual length from the video sequence itself. It is based on the observation that given a sufficiently long video sequence, each link will become parallel (or nearly parallel) to the plane of screen at least once. We further ensure the consistency of our method by normalizing it with respect to anthropometric data. For example, for left and right fore-arms, we use maximum length of the two.

Once we have the true length of a link, potentially we can decompose the orientation of the link to two possible cases. One endpoint of the link will be displaced from its reference plane attached to the other endpoint, by a relative value which is proportional to the difference of their z-components. This difference, however, suffers from orthographic reflective ambiguity. That is, the actual depth of the link may be in positive or negative direction. In both the cases, the 2D projection would exactly be the same as seen in figure 3.

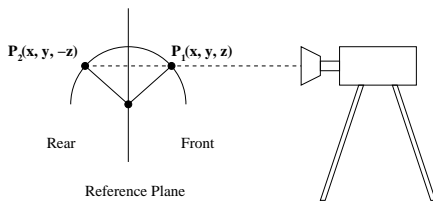


Figure 3. Reflective ambiguity under orthographic projection: Two 3D points can have same 2D projection.

### 4.2. Pose generation

Since each of the links considered has two possible 3D configurations for a given 2D configuration and since adjacent links are joined, there are  $2^{\#links}$  permutations of the overall body configuration. We have to explore all these possibilities for each frame. Figure 4 shows the formulation of these  $2^{\#links}$  permutations. We consider the neck as the root of the skeleton, which will not undergo any change in configuration throughout. Left shoulder may be in front or rear of the reference plane attached to the neck. Same is the case with the right shoulder, making a total of four possibilities considering only shoulders. At the lowest level of the tree, we will have  $2^{\#links}$  leaves.

Of course, not all of these configurations are physically feasible. We need to add constraints, which will allow only those poses which are physically attainable by humans, in order to prune this exponentially huge set of permutations.

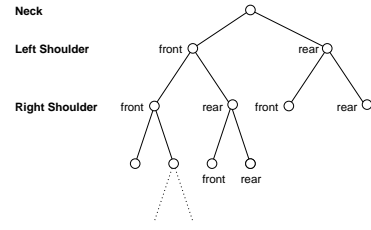


Figure 4.  $2^{\#links}$  possible permutations: At each joint, we have 2 possibilities.

We use various kinds of constraints viz. model, joint angle limit, and collision constraints.

#### 4.2.1. Model constraints

Model constraints enforce connectivity between adjacent links and link length constancy. These constraints are very basic. To enforce them, we have to use a 3D kinematic model which satisfies these constraints. Using such 3D kinematic model for the given 2D measurements itself will restrict the number of possible solutions to a finite number.

#### 4.2.2. Joint Angle Limits

Each joint of the human body has a minimum and a maximum limit of angle of bend that is possible to achieve. For instance, it is not possible to bend the arm at the elbow joint below 10 degrees. However, such a pose might have been represented in some of the permutations. This is taken care of by joint angle limits. All poses, which have at least one joint whose angle crosses either the maximum or the minimum limit, are marked invalid and are not considered for further processing.

#### 4.2.3. Collision Constraints

Because of solid nature of the body, one part of the body cannot penetrate through another part. Collision constraints ensure this behavior by checking whether any link collides with any other link. To enforce these constraints, the stick figure model is not enough; we need to represent each of the link by a 3D structure like a cylinder or an ellipsoid.

Using the above constraints, we find all invalid poses and remove them from any further processing. In our experiments, we found that almost 70% of the  $2^{\#links}$  poses were invalidated by joint angle and collision constraints. Hence, though the complexity of the algorithm will be exponential, the main time-consuming processing will be done on only a limited number of cases.

### 4.3. Graph Formulation

All the processes done till now work on individual frames and will produce a set of all possible valid poses for each frame. However, if we want to establish a valid 3D pose sequence for some time duration, it is necessary to consider the temporal dimension of the input. Given a valid pose  $x$  in frame  $i$  and a valid pose  $y$  in frame  $i + 1$ , it is not always possible for a person to change body pose from  $x$  to  $y$  within the time duration of one frame. We formulate this problem as a graph problem, since it is very easy to visualize a pose as a node and a transition between two poses as an edge between corresponding nodes.

We form the graph in the following way. The graph basically has a layered structure with each frame being represented by a layer. Each valid pose at each frame is represented as a node in the corresponding layer. Edges are established between nodes  $A$  and  $B$  in adjacent layers, if it is possible to change pose from  $A$  in first frame to  $B$  in next frame. Each of the edge carries a weight which represents some metric of the transformation between the poses. Various metrics that are possible are change in angles, change in depth, and angular velocity.

Now our problem reduces to finding a minimum weight path from a node in first layer to a node in the last layer. This can be done using standard dynamic programming techniques like Viterbi algorithm

#### 4.3.1. Calculating weights

Whenever the body moves, there is change in configuration of some joints. This change may be in orientation, angle, velocity, acceleration, or any combination of them. When the body moves swiftly, these changes should not be sudden. We are exploiting an interesting observation that the motion involved during the snippet where a link crosses its reference plane is generally smooth [4]. Hence we can assume acceleration associated with that link to be nearly zero during this time interval.

#### 4.3.2. Change in Velocity

During the movement, each joint angle has some angular velocity associated with it. For smooth motion, these velocities should not change drastically. i.e. the acceleration should be as small as possible. Thus we can use sum of accelerations as a merit to weigh the edges of the graph.

Let  $\mathbf{A}(\phi_k)$  denote the position vector of joint  $\mathbf{A}$  at frame  $k$ .  $\mathbf{AB}(\phi_k)$  denote the vector from joint  $\mathbf{A}$  to joint  $\mathbf{B}$  on a segment  $\mathbf{AB}$ , all in body configuration  $\phi_k$ . Assume the interval between two consecutive frames is  $\Delta t$ . Then the relative translation velocity of the segment  $\mathbf{AB}$  from body configu-

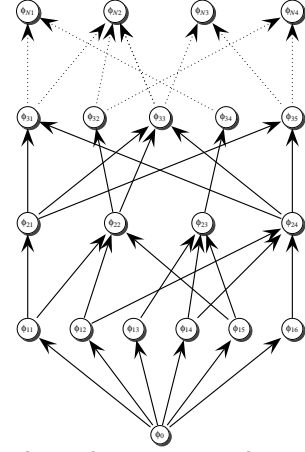


Figure 5. Graph used to represent the poses and their transitions.

ration  $\phi_k$  in frame  $k$  to another body configuration  $\phi_{k+1}$  in frame  $k + 1$  is defined as

$$\begin{aligned} \mathbf{V}_{\mathbf{AB}}(\phi_k, \phi_{k+1}) &= \frac{\{[\mathbf{B}(\phi_{k+1}) - \mathbf{B}(\phi_k)] - [\mathbf{A}(\phi_{k+1}) - \mathbf{A}(\phi_k)]\}}{\Delta t} \\ &= \frac{[\mathbf{AB}(\phi_{k+1}) - \mathbf{AB}(\phi_k)]}{\Delta t} \end{aligned}$$

The relative angular velocity and acceleration of segment  $\mathbf{AB}$  are defined as

$$\boldsymbol{\omega}_{\mathbf{AB}}(\phi_k, \phi_{k+1}) = \mathbf{AB} \times \mathbf{V}_{\mathbf{AB}}(\phi_k, \phi_{k+1}) \quad (1)$$

and

$$\boldsymbol{\alpha}_{\mathbf{AB}}(\phi_k, \phi_{k+1}, \phi_{k+2}) = \frac{|\boldsymbol{\omega}_{\mathbf{AB}}(\phi_{k+1}, \phi_{k+2}) - \boldsymbol{\omega}_{\mathbf{AB}}(\phi_k, \phi_{k+1})|}{\delta t} \quad (2)$$

respectively. A smooth angular motion of a body segment during walking indicates a nearly zero angular acceleration. An angular acceleration function associated with body configuration  $\phi_k, \phi_{k+1}, \phi_{k+2}$  can be defined as

$$f_k(\phi_k, \phi_{k+1}, \phi_{k+2}) = \sum_{\mathbf{AB}} |\boldsymbol{\alpha}_{\mathbf{AB}}(\phi_k, \phi_{k+1}, \phi_{k+2})| \quad (3)$$

where the summation is taken over all body segments and the magnitudes of angular accelerations are used for simplicity. Our aim is to minimize this measure of angular accelerations.

However, this increases the complexity of the algorithm very much. Since in this case we have to consider all possible transitions between adjacent frames to calculate acceleration.

### 4.4. Velocity based estimation

Instead of calculating the absolute difference between two quantities (depths or angles) of two poses, we can use the known velocity at one frame to estimate the position of each of the joint in the next frame.

$$\mathbf{A}_{\text{est}}(\phi_{k+1}) = \mathbf{A}(\phi_k) + \mathbf{V}_A \Delta t \quad (4)$$

Now, we find the difference between the estimated values and the observed values. This gives us an estimate of error function at each frame. This error function is used as weights for edges.

$$\mathbf{e} = \mathbf{A}(\phi_{k+1}) - \mathbf{A}_{\text{est}}(\phi_{k+1}) \quad (5)$$

## 5. Results

We present some of the results obtained by us, in this section. The first row of Fig. 6 shows some sample input frames. The skin regions are extracted and largest blobs are retained which are labeled to corresponding body parts, as seen in the second row. Ellipses are fitted around each of these regions as can be seen in the third row. The major axes of these ellipses give the joint locations. Using these joint locations, the stick-figure representation of the body is made. The fourth row shows the stick-figures. These figures are in 2D. The reconstruction step outlined in the paper estimates 3D coordinates for each of the joint creating a 3D model. The last row of the figure shows the same frames from a different viewpoint to confirm its 3-dimensionality.

## 6. Final Remarks

We have presented a computer vision based method to use the domain specific knowledge to obtain 3D configuration of a human body from monocular video sequence. We have implemented the system only for the upper body. However similar concepts hold for the entire body. The cues required for tracking the lower limbs may be different. Especially, it is difficult to obtain the position of knees because of loose-fitting dress. The reconstruction algorithm remains the same for the full body motion capture, though its running time may increase heavily because of non-linear nature of the algorithm.

## Acknowledgments

We would like to thank Aarthi and Guru Padmini Radhakrishnan of Soundarya Natya Kalalaya for their kind cooperation in allowing us to record the Bharatanatyam performance and use it for the purpose of this research. We thank Nilesh and Kashyap for their assistance in implementation of the system.

## References

[1] A. Agarwal and B. Triggs. 3d human pose from silhouettes by relevance vector regression. In *International Conference on Computer Vision & Pattern Recognition*, pages II 882–888, Washington, June 2004.

[2] A. Blake, R. Curwen, and A. Zisserman. A framework for spatio-temporal control in the tracking of visual contours. In *International Journal of Computer Vision*, volume 11, 1993.

[3] C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, page 8. IEEE Computer Society, 1998.

[4] Z. Chen and H.-J. Lee. Knowledge guided visual perception of 3d human gait from a single image sequence. *IEEE Transactions on systems, man and cybernetics*, 22(2), March 1992.

[5] J. Costeira and T. Kanade. A multi-body factorization method for motion analysis. pages 1071–, 1995.

[6] J. M. David E.DiFranco, Tat-Jen Cham. Reconstruction of 3-d figure motion from 2-d correspondences. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2001.

[7] Q. Delamarre and O. Faugeras. 3d articulated models and multi-view tracking with silhouettes. *Proc. International Conference on Computer Vision*, 2:716–721, 1999.

[8] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2:126–133, 2000.

[9] R. Feris, T. Campos, and R. Junior. Detection and tracking of facial features in video sequences. volume 1793, pages 127–135, April 2000.

[10] R. Kalman. A new approach to linear filtering and prediction problems. *Transactions of ASME - Journal of basic engineering*, pages 35–45, 1960.

[11] M. Kass, D. Terzopoulos, and A. Witkin. Snakes:active contour models. 1987.

[12] I. Mikic, M. Triverdi, E. Hunter, and P. Cosman. Articulated body posture estimation from multi-camera voxel data. *Proc. IEEE Computer Vision and Pattern Recognition*, 2001.

[13] A. Mittal, L. Zhao, and S. Larry. Human body pose estimation using silhouette shape analysis. *IEEE Int Conf on Advanced Video and Signal Based Surveillance*, July 2003.

[14] T. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, 2001.

[15] M. J. Park, M. G. Choi, and S. Y. Shin. Human motion reconstruction from inter-frame feature correspondences of a single video stream using a motion library. pages 113–120, 2002.

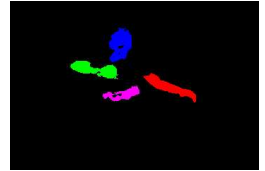
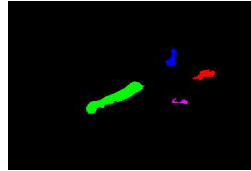
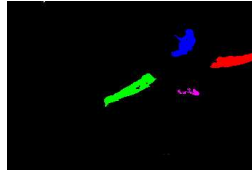
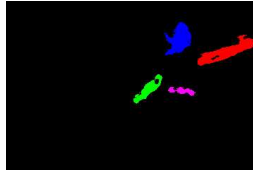
[16] C. J. Poelman and T. Kanade. A paraperspective factorization method for shape and motion recovery. *IEEE transactions on pattern analysis and machine intelligence*, 19:206–218, Mar 1997.

[17] J. Rehg and T. Kanade. Model-based tracking of self-occluding articulated objects. *Proc. International Conference on Computer Vision*, pages 612–617, Jun 1995.

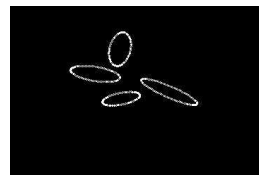
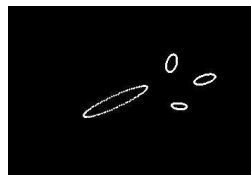
[18] J. Shi and C. Tomasi. Good features to track. Technical report, 1993.



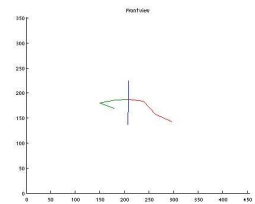
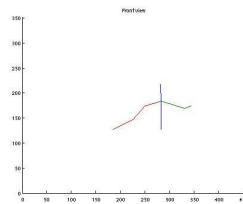
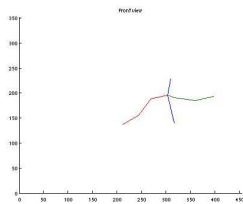
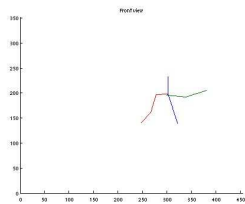
(a) A few selected frames from the original video sequence



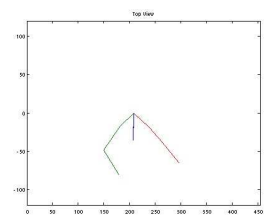
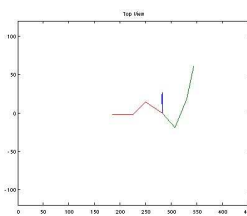
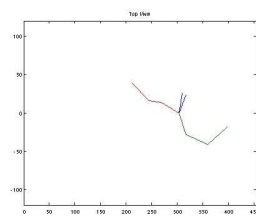
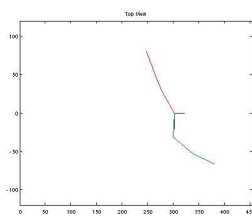
(b) The blobs of skin color, color coded for easy identification



(c) An ellipse is fitted around each blob of skin color region



(d) Stick figure model fitted to the human body based on ellipses



(e) The model as viewed from top.

Figure 6. Some sample results of our implementation

[19] H. Sidenbladh, M. J. Black, and D. J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. pages 702–718, 2000.

[20] C. Sminchisescu and B. Triggs. Kinematic jump processes for monocular 3d human tracking. In *International Conference on Computer Vision & Pattern Recognition*, pages 169–76, June 2003.

[21] C. Tomasi and T. Kanade. Shape and motion from image

streams under orthography — a factorization method. *International Journal on Computer Vision*, Nov 1992.

[22] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: real-time tracking of the human body. In *Proceedings of the 2nd International Conference on Automatic Face and Gesture Recognition (FG '96)*, page 51. IEEE Computer Society, 1996.