Articulated Body Posture Estimation from Multi-Camera Voxel Data

Ivana Mikić, Mohan Trivedi, Edward Hunter, Pamela Cosman
Computer Vision and Robotics Research Lab
Department of Electrical and Computer Engineering
University of California, San Diego, La Jolla, CA 92093

Abstract

We present a framework for articulated body model acquisition and tracking from voxel data. A 3D voxel reconstruction of the person's body is computed from silhouettes extracted from four cameras. The model acquisition process is fully automated. In the first frame, body parts are located sequentially. The head is located first, since its shape and size are unique and stable. Other parts are found by sequential template growing and fitting. This initial estimate of body part locations, sizes and orientations is then used as a measurement for the extended Kalman filter which ensures a valid articulated body model. The same filter, with a slightly modified state and state transition matrix, is then used for tracking. The performance of the system has been evaluated on several video sequences with promising results.

1. Introduction

Posture estimation is the problem of extracting the parameters of a model of the human body from video data. The model is usually chosen a priori and the estimation algorithm extracts its parameters. Posture estimation is useful in many applications such as advanced user interfaces [1, 2], intelligent environments [3], entertainment, surveillance systems [4], or motion analysis for sports and medical purposes [5]. In the past few years, the problem of markerless, unconstrained posture estimation using only cameras has received much attention from computer vision researchers [6, 7, 8, 9, 10].

Many existing pose estimation systems require manual initialization of the model and then perform the tracking. The systems that use multiple camera images as inputs, most often analyze the data in the image plane, comparing it with the appropriate features of the model projection [11, 12, 13]. Promising results have been reported in using the depth data obtained from stereo [14, 15] for pose estimation. However, only recently the first attempts at using voxel data obtained from multiple cameras to estimate body pose have been reported [16]. This system used a very simple initialization and tracking procedure that did not guarantee a valid articulated body model, but showed that voxel data can be successfully used for posture estimation.

In this paper we present a system for articulated body model acquisition and tracking from voxel data (Figure 1). Video from four cameras is segmented and a voxel reconstruction of the person's body is computed from the four silhouettes. We are using low resolution data – the four images are 320x240 pixels and the voxel size we chose is 50mm. In the first frame, an automatic model acquisition is performed – the head is found first by template matching and the other body parts by a sequential template growing procedure. To ensure a valid articulated body model, this initial estimate is adjusted by the extended Kalman filter. The same filter with a slightly modified state and state transition matrix is then used for tracking. We also describe a novel framework for posture estimation from voxel data based on the multilevel body model that will enable progressive model acquisition as observations are made from frame to frame.

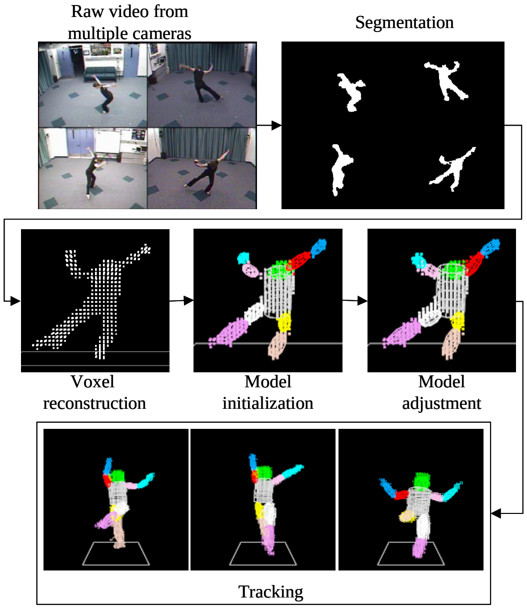


Figure 1 System components

In Section 2 we present our voxel reconstruction algorithm. The description of the model initialization procedure follows in Section 3. The design of the extended Kalman filter that performs model adjustment in the first frame and tracking between frames is outlined in Section 4. Results of our experiments are presented in Section 5. The multilevel posture estimation framework is described in Section 6.

ISBN 0-7695-1272-0/01 $10.00 (C) 2001 IEEE

Concluding remarks follow in Section 7. The two movie files that are referred to in this paper can be found at: http://cvrr.ucsd.edu/~ivana/cvpr.

2. Voxel reconstruction

To compute the voxel reconstruction, we first segment the four camera images using the algorithm described in [17] which eliminates shadows and highlights and produces good quality silhouettes. Based on the centroids and bounding boxes of the 2D silhouettes, a bounding volume of the person is computed. Cameras are calibrated using Tsai's algorithm [18].

Reconstructing a 3D shape (which we will call a 3D silhouette) using 2D silhouettes from multiple images is called voxel carving or shape from silhouettes. Octree [19, 20] is one of the best known approaches to voxel carving. The volume of interest is first represented by one cube, which is progressively subdivided into eight subcubes. Once it is determined that a subcube is entirely inside or entirely outside the 3D silhouette, its subdivision is stopped. Cubes are organized in a tree, and once all the leaves stop dividing, the tree gives an efficient representation of the 3D silhouette. The more straightforward approach is to check for each voxel if it is consistent with all 2D silhouettes. With several clever speed-up methods, this approach is described in [16]. In this system, the person is known to always be inside a predetermined volume of interest. A projection of each voxel in that volume onto each of the image planes is precomputed and stored in a lookup table. Then, at runtime, the process of checking whether the voxel is consistent with a 2D silhouette is very fast since the use of the lookup table eliminates most of the necessary computations.

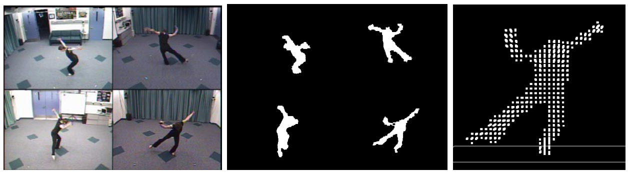


Figure 2 3D voxel reconstruction. Left: original input images. Middle: extracted 2D silhouettes. Right: resulting 3D silhouette

Our goal is to allow a person unconstrained movement in a large space and we also plan to incorporate multiple pan-tilt cameras in the future. Therefore, designing a lookup table that maps voxels to pixels in each camera image is not practical. Instead, we pre-compute a lookup table that maps points from undistorted sensor-plane coordinates (divided by focal length and quantized) in Tsai's model to the image pixels (Equation 1). Then, the only computation that is performed at runtime is mapping from world coordinates to undistorted sensor-plane coordinates (Equation 2). An example frame is shown in Figure 2. The equations are given below, where $\mathbf{x}_w$ is the voxel's world coordinate, $\mathbf{x}_c$ is its coordinate in the camera coordinate system, X_u and Y_u are undistorted sensor-plane coordinates, X_d and Y_d are distorted sensor-plane coordinates and X_f and Y_f are pixel coordinates. The lookup table is fixed for a camera regardless of its orientation (would work for a pan/tilt camera also – only rotation matrix **R** and a translation vector **T** change in this case).

Lookup table computations:

$$\boxed{\begin{array}{c} \left(\frac{X_u}{f},\frac{Y_u}{f}\right)\rightarrow\left(X_f,Y_f\right) \\ \hline X_u = X_d\left(1+\kappa_1\left(X_d^2+Y_d^2\right)\right) \\ Y_u = Y_d\left(1+\kappa_1\left(X_d^2+Y_d^2\right)\right) \\ X_f = d_x^{-1}X_d s_x + C_x \\ Y_f = d_y^{-1}X_d + C_y \end{array}} \quad (1)$$

Run-time computations:

$$\boxed{\begin{array}{c} \left(x_w,y_w,z_w\right)\rightarrow\left(\frac{X_u}{f},\frac{Y_u}{f}\right) \\ \hline \mathbf{x}_c = \mathbf{R}\mathbf{x}_w + \mathbf{T} \\ \frac{X_u}{f}=\frac{x_c}{z_c},\ \ \frac{Y_u}{f}=\frac{y_c}{z_c} \end{array}} \quad (2)$$

3. Model initialization

Figure 5 shows the 10-part body model used in the proposed system. The initialization procedure locates the body parts in the first frame (Figure 4 and Movie 1 illustrate its operation). Due to its unique shape and size, the head is easiest to find and is located first. We create a spherical crust template whose inner and outer diameters correspond to the smallest and largest head dimensions we expect to see. For the head center we choose the location of the template center that maximizes the number of surface voxels that are inside the crust. Then, the voxels that are inside the sphere of the larger diameter, centered at the chosen head center are labeled as belonging to the head, and the true center, size and orientation of the head are recomputed from those voxels. Next, the approximate location of the neck is found as an average over those head voxels which have at least one non-head body neighboring voxel. The template of an average sized torso is then placed with its base at the neck and with its axis going through the centroid of non-head voxels. The voxels inside this template are then used to recompute a new centroid, and the template is rotated so that its axis passes through it (torso is anchored to the neck at the center of its base at all times). This procedure is repeated until the template stops moving, which is accomplished when the template is entirely inside the torso or is well centered over it. Even with an initial centroid that is completely outside the body,

ISBN 0-7695-1272-0/01 $10.00 (C) 2001 IEEE

this procedure converges, since in the area close to the neck, the template always contains some torso voxels that help steer the template in the right direction (see Figure 3).

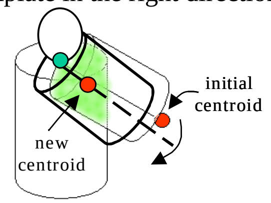


Figure 3 Fitting the torso. Initial torso template is placed so that its base is at the neck and its main axis passes through the centroid of non-head voxels. Voxels that are inside the template are used to calculate new centroid and the template is rotated to align the main axis with the new centroid. The process is repeated until the template stops moving which happens when it is entirely inside the torso or is well centered over it.

The torso template is then shrunk to a small predetermined size in its new location and grown in all dimensions until further growth starts including empty voxels. In the direction of the legs, the growing will stop at the place where legs part. The voxels inside this new template are labeled as belonging to the torso.

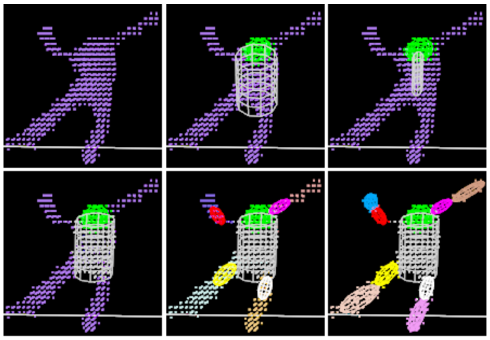
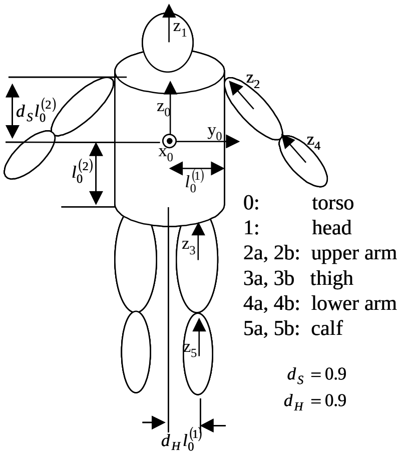


Figure 4 Model initialization. From left to right, top to bottom: (1) 3D silhouette, (2) head located and torso template after the initial positioning process, (3) shrunk torso template, (4) after the growing process, the torso voxels are labeled and the torso model fitted to them, (5) upper arms and thighs after the growing process, (6) lower arms and calves fitted onto the remaining voxels.

Next, the four regions belonging to the limbs are found as the four largest connected regions of remaining voxels, the appropriate joints are located and the same growing procedure described for the torso is repeated for thighs and upper arms. The lower arms and calves are found by locating connected components closest to the identified upper arms and thighs.

Once the voxels belonging to a certain part are labeled in the initialization procedure just described, the location, orientation and size of each body part are computed using the labeled data. From the eigenvalue decomposition of the covariance matrix, the orientation of the body part is available. The computed eigenvalues correspond to the variances for three principal directions (σ_i). We assume uniform distribution of voxels inside the body part. The actual size of the body part ($2\,l_i$) depends on the shape of the part and on σ_i. It can be easily derived that with our assumption of uniform distribution inside the part, for the ellipsoid we have: $l_i = \sqrt{5}\sigma_i$ and for the cylinder: $l_i = 2\sigma_i$ (for the two dimensions of the base) and $l_i = \sqrt{3}\sigma_i$ (for the height of the cylinder). When computing the orientation and the size of a body part, we always order the dimensions in ascending order and orient the local coordinate system accordingly, to maintain constant orientation of the local body part coordinate systems with respect to the corresponding body parts. Also, the orientations of the local coordinate systems have to be checked against the model shown in Figure 5 to ensure consistent orientations (for example, z axis of the head always pointing away from the neck) and rotated if necessary.



neck

$$\mathbf{J}_{01} = \begin{bmatrix} 0 & 0 & l_0^{(2)} \end{bmatrix}^T$$
$$\mathbf{J}_{10} = \begin{bmatrix} 0 & 0 & -l_1^{(2)} \end{bmatrix}^T$$

shoulder

$$\mathbf{J}_{02} = \begin{bmatrix} 0 & \pm l_0^{(1)} & d_S l_0^{(2)} \end{bmatrix}^T$$
$$\mathbf{J}_{20} = \begin{bmatrix} 0 & 0 & l_2^{(2)} \end{bmatrix}^T$$

hip

$$\mathbf{J}_{03} = \begin{bmatrix} 0 & \pm d_H l_0^{(1)} & -l_0^{(2)} \end{bmatrix}^T$$
$$\mathbf{J}_{30} = \begin{bmatrix} 0 & 0 & l_3^{(2)} \end{bmatrix}^T$$

elbow

$$\mathbf{J}_{24} = \begin{bmatrix} 0 & 0 & -l_2^{(2)} \end{bmatrix}^T$$
$$\mathbf{J}_{42} = \begin{bmatrix} 0 & 0 & l_4^{(2)} \end{bmatrix}^T$$

knee

$$\mathbf{J}_{35} = \begin{bmatrix} 0 & 0 & -l_3^{(2)} \end{bmatrix}^T$$
$$\mathbf{J}_{53} = \begin{bmatrix} 0 & 0 & l_5^{(2)} \end{bmatrix}^T$$

Figure 5 Body model. Body model defines how the body parts are connected. Joint locations are expressed in the local body-part coordinate systems, as functions of the body part dimensions.

4. Model adjustment and tracking

The initialization procedure locates body parts well, however, the resulting model is not guaranteed to be valid (body parts do not necessarily touch and locations of joints may not agree with the model). To ensure the valid model in the first frame, we use the obtained initial body part positions as measurements for the extended Kalman filter (EKF), which adjusts the configuration of the articulated model to best fit the data (Figure 6).

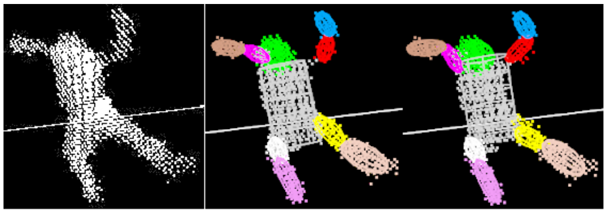


Figure 6 Model adjustment in the first frame. Left: the original voxel reconstruction. Middle: Body parts located in the initialization process. Note that the model is not valid at this stage – both arms are detached from the torso and right hip is at the side of the torso. Right: after the adjustment by the extended Kalman filter, the model is valid and fitted to the data.

The articulated body model we use is shown in Figure 5. The orientations of body parts are represented by quaternions ($\mathbf{q}_i$), i.e. by an axis and an angle of rotation relative to the world coordinate system. Centroids, orientations and sizes for each body part ($\hat{\boldsymbol{\mu}}_i, \hat{\mathbf{q}}_i, \hat{\mathbf{l}}_i, i \in (0,1,2a,2b,3a,3b,4a,4b,5a,5b)$, respectively) are available measurements included in the measurement vector ($\mathbf{z}_k$). To ensure a valid articulated model, we include the non-redundant set of model parameters into the Kalman filter state ($\mathbf{x}_k$): the orientations and sizes of all body parts and the torso centroid ($\mathbf{q}_i, \mathbf{l}_i, i \in (0,1,2a,2b,3a,3b,4a,4b,5a,5b), \boldsymbol{\mu}_0$, respectively). As in [21, 15], the centroids of other body parts are expressed with kinematic chain equations using these parameters. For example, the centroid of the lower arm $\boldsymbol{\mu}_4$ is expressed as a function of the torso centroid $\boldsymbol{\mu}_0$, rotation matrices representing orientations of torso, upper arm and lower arm ($\mathbf{R}_0$, $\mathbf{R}_2$ and $\mathbf{R}_4$ respectively), and shoulder and elbow locations in the local coordinate systems ($\mathbf{J}_{02}$, $\mathbf{J}_{20}$, $\mathbf{J}_{24}$, $\mathbf{J}_{42}$):

$$\boldsymbol{\mu}_4 = \boldsymbol{\mu}_0 + \mathbf{R}_0\mathbf{J}_{02} - \mathbf{R}_2\mathbf{J}_{20} + \mathbf{R}_2\mathbf{J}_{24} - \mathbf{R}_4\mathbf{J}_{42} \qquad (3)$$

As shown in Figure 5, joint locations in the local body-part coordinate systems are expressed as functions of body part dimensions. In the Kalman filter equations [22]:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{F}\mathbf{x}_k + \mathbf{u}_k \\ \mathbf{z}_k &= \mathbf{H}(\mathbf{x}_k) + \mathbf{w}_k \end{aligned} \qquad (4)$$

the only nonlinear parts of the measurement equation are the relationships between the body part centroids and the chosen state parameters, which are linearized around the predicted state in the extended Kalman filter. For example, to linearize Equation 3, we would have:

$$\frac{\partial \boldsymbol{\mu}_4}{\partial \boldsymbol{\mu}_0} = \mathbf{I}_3$$

$$\frac{\partial \boldsymbol{\mu}_4}{\partial \mathbf{l}_0} = \mathbf{R}_0 \frac{\partial \mathbf{J}_{02}}{\partial \mathbf{l}_0} = \mathbf{R}_0 \begin{bmatrix} 0 & 0 & 0 \\ 0 & \pm 1 & 0 \\ 0 & 0 & d_S \end{bmatrix}$$

$$\frac{\partial \boldsymbol{\mu}_4}{\partial \mathbf{q}_0} = \frac{\partial \mathbf{R}_0}{\partial \mathbf{q}_0} \mathbf{J}_{02} = \frac{\partial\left(\exp(\hat{\boldsymbol{\omega}}\theta)\right)}{\partial \mathbf{q}_0} \mathbf{J}_{02}$$

where $\boldsymbol{\omega}$ and θ are the vector along the rotation axis and the rotation angle respectively. If $\mathbf{q}=[q_0,\mathbf{q}_v]^T$, then $\theta = 2\arccos q_0$ and $\boldsymbol{\omega} = \mathbf{q}_v / \sin(\theta/2)$ [23].

For adjustment of the model in the initial frame, we set the state transition matrix **F** to the identity matrix.

The same filter is then used to perform tracking, except that we add the velocity of the torso centroid ($\dot{\boldsymbol{\mu}}_0$) to the state and modify the state transition matrix to include the equation:

$$\boldsymbol{\mu}_0(k+1) = \boldsymbol{\mu}_0(k) + \dot{\boldsymbol{\mu}}_0(k)\Delta T \qquad (5)$$

where ΔT is the sampling period. We currently only model the overall motion of the body, approximately captured by the motion of the torso centroid, and not the motions of the individual body parts in the dynamic (plant) equation of the Kalman filter.

To obtain new measurements in the next frame, the voxels are assigned to body parts by minimizing the Mahalanobis distance from the EKF prediction. Using the labeled voxels, a new measurement is produced as in the first frame, by computing the centroid and the covariance matrix (from which the orientation and dimensions are estimated) for each body part. Again, the orientations of the body parts need to be checked against the model to ensure the orientation of the local coordinate system is consistent with the model. For example, if the computed orientation of the thigh in the new frame has the z axis pointing away from the hip, the local coordinate system has to be rotated by 180 degrees to ensure that the location of the hip joint in the local coordinate system is consistent with the model.

5. Results

In this section, we present the results of model initialization, adjustment and tracking. Two movies illustrating the results can be found at http://cvrr.ucsd.edu/~ivana/cvpr. Movie 1 illustrates the process of model initialization. In our experiments, localization of the head was very robust. In approximately

70% of the 600 frames - 3 sequences of 200 frames each, the head was accurately located. Moreover, in the cases when the head localization fails (usually due to errors in the voxel reconstruction which are caused by segmentation errors and sometimes by the pose that occludes the head), the number of surface voxels inside the crust template is significantly smaller than in the frames where the head is correctly located. A simple threshold can be set up to decide if the head location should be accepted or if the frame should be skipped. If the head is located properly, the torso is found correctly in nearly all of the cases. Of course, if the pose is such that the torso cannot be separated from the limbs, it will be larger than the true torso, but as will be explained in the next section, we consider such torso estimates to be correct. If the body pose reveals other body parts (i.e., if there is at least a slight angle in the knees and elbows or if legs are apart) they will be located also. The system currently expects such a pose, and in the next section we discuss how other cases would be handled.

Adjustment of the initial pose using the extended Kalman filter works very well. The initialization process finds body parts close to their true positions and the adjustment process incorporates these measurements into the valid body model.

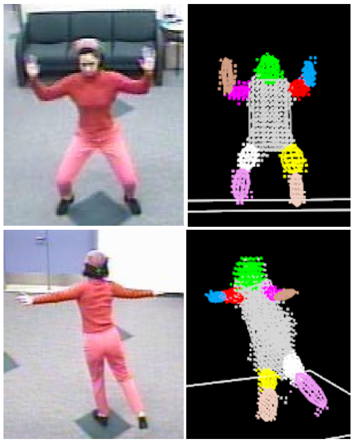


Figure 7 Tracking. Comparison between one of the input images and the configuration of the model. Models of body parts are overlayed on top of the labeled voxels.

The tracking has been tested on several sequences with good results (Figures 7 and 8 and Movie 2). However, the motion has to be slow relative to the frame rate. This is due to the voxel labeling process based on Mahalanobis distance minimization and the fact that the current motion model cannot predict abrupt changes in the body part motion. Combining the predicted model configuration with an approach similar to the initialization process may lead to a more robust tracking performance.

The tracking currently takes about 20 seconds per frame on a 450 MHz Pentium. We have not made any effort to speed up the performance yet. There are several improvements that could be made to increase the speed, such as looking only at surface voxels for Mahalanobis distance computations or taking advantage of the sparseness of the **H** matrix in the EKF equations.

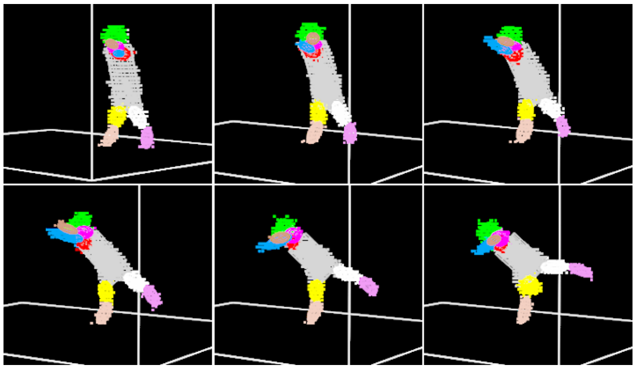


Figure 8 Tracking.

6. Multilevel posture estimation framework

The system presented in this paper is the first stage in the implementation of the framework for posture estimation from voxel data that is the goal of our efforts. We intend to make the model initialization robust to the pose of the person in the beginning of the sequence, and our framework will allow the system to acquire the model progressively as the necessary observations become available. For example, if the person is standing with legs close together and with the arms close to the body, the described initialization procedure would locate only the head and a very large torso (see Figure 9). If the legs are apart, but perfectly straight, the algorithm would detect only one large component for the leg instead of two smaller ones (thigh and calf). We believe that this intrinsic ambiguity in the data should be modeled with a multilevel body model - a hierarchy of models, where at different levels the different amount of detail is captured (Figure 9).

Depending on the available data, the appropriate model configuration will be estimated and progression to a finer model will occur when the data supports it. This progression need not occur at the same time for the whole body. Rather, the different parts of the body will be refined as their configuration reveals necessary detail. The Kalman filtering formulation presented in this paper can be very easily modified to track different models at different levels, by addition and deletion of appropriate rows of matrices **F** and **H** and elements of state and measurement vectors that correspond to different body parts.

ISBN 0-7695-1272-0/01 $10.00 (C) 2001 IEEE

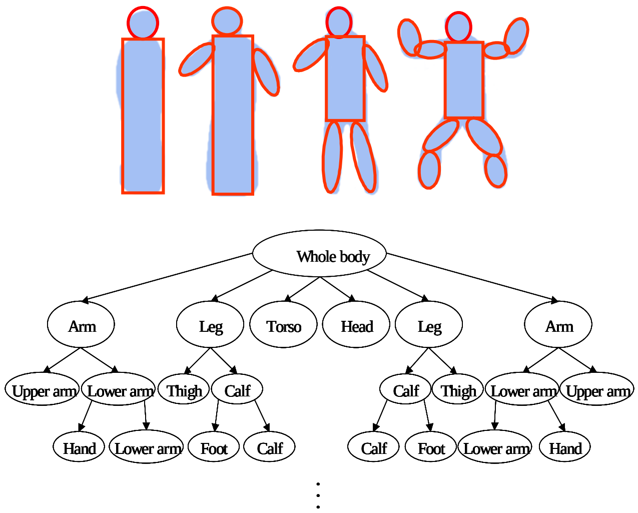


Figure 9 Multilevel body model. Depending on the quality of the available data, different body models are appropriate.

7. Conclusions

As mentioned earlier, we plan to include pan/tilt cameras in the system to allow unconstrained movement in a larger space. The relatively large voxel size (50mm) we chose, makes the algorithm more robust to noise and is appropriate for the image resolution of our input data. With higher resolution input images, smaller voxel sizes could be used which would improve the quality of the voxel reconstruction and should lead to higher quality posture estimates.

In conclusion, we have demonstrated that articulated body posture estimation from voxel data is robust and convenient. Since the voxel data is in the world coordinate system, simple algorithms that take advantage of the knowledge of average dimensions and shape of some parts of the human body are easily implemented. Also, most of the parameters of the articulated body model are directly measurable from the voxel data, making the Kalman filter formulation straightforward.

Acknowledgement

Our research is supported by the California Digital Media Innovation Program (DiMI) in partnership with Sony Electronics, Compaq Computers, Caltrans, and DaimlerChrysler Reseach Division.

References

[1] C. Wren, Understanding Expressive Action, Ph.D. Thesis, Massachusetts Institute of Technology, March 2000

[2] T. Moeslund, Interacting with a Virtual World Through Motion Capture", Interaction in Virtual Inhabited 3D Worlds (L. Qvortrup, ed.), Springer-Verlag, 2000

[3] I. Mikić, K. Huang, M. Trivedi, "Activity monitoring and summarization for an intelligent meeting room", Workshop on Human Motion, Austin, Texas, December 2000

[4] I. Haritaoglu, D. Harwood, L. Davis, "W4: Real-time surveillance of people and their activities", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 22, no. 8, August 2000, pp. 809-830

[5] D. Gavrila, "Visual Analysis of Human Movement: A Survey", Computer Vision and Image Understanding, vol. 73, no. 1, January 1999, p. 82-98

[6] T. Moeslund, E. Granum, "A Survey of Computer Vision-Based Human Motion Capture", Computer Vision and Image Understanding, Vol. 81, No. 3, March 2001, pp. 231-268

[7] I. Kakadiaris, D. Metaxas, "Three-Dimensional Human Body Model Acquisition from Multiple Views", International Journal of Computer Vision, vol. 30, no. 3, 1998, p. 191-218

[8] I. Kakadiaris, D. Metaxas, "Model-Based Estimation of 3D Human Motion with Occlusion Based on Active Multi-Viewpoint Selection", Proc. IEEE International Conference on Computer Vision and Pattern Recognition, San Francisco, CA, June 1996

[9] Q. Delamarre, O. Faugeras, "3D Articulated Models and Multi-View Tracking with Physical Forces", Computer Vision and Image Understanding, Vol. 81, No. 3, March 2001, pp. 328-357

[10] C. Bregler, "Learning and Recognizing Human Dynamics in Video Sequences", IEEE International Conference on Computer Vision and Pattern Recognition, San Juan, Puerto Rico, June 1997

[11] J. Deutscher, A. Blake, I. Reid, "Articulated Body Motion Capture by Annealed Particle Filtering", IEEE Int. Conference on Computer Vision and Pattern Recognition, Hilton Head Island, SC, July 2000

[12] D. Gavrila, L. Davis, "3D model-based tracking of humans in action: a multi-view approach", Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 18-20 June 1996, p.73-80

[13] C. Bregler, J. Malik, "Tracking People with Twists and Exponential Maps", IEEE International Conference on Computer Vision and Pattern Recognition, Santa Barbara, CA, June 1998

[14] M. Covell, A. Rahimi, M. Harville, T. Darrell, "Articulated-pose estimation using brightness- and depth-constancy constraints", IEEE Int. Conference on Computer Vision and Pattern Recognition, Hilton Head Island, SC, June 2000, p. 438-445

[15] N. Jojić, M. Turk, T. Huang, "Tracking Self-Occluding Articulated Objects in Dense Disparity Maps", IEEE Int. Conference on Computer Vision, Corfu, Greece, September 1999

[16] G. Cheung, T. Kanade, J. Bouguet, M. Holler, "A Real Time System for Robust 3D Voxel Reconstruction of Human Motions", Proceedings IEEE Conference on Computer Vision and Pattern Recognition, Hilton Head Island, SC, USA, June 2000, p.714-20, vol.2

[17] T. Horprasert, D. Harwood, and L.S. Davis, "A Statistical Approach for Real-time Robust Background Subtraction and Shadow Detection", Proc. IEEE ICCV'99 FRAME-RATE Workshop, Kerkyra, Greece, September 1999.

[18] R. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses", IEEE Journal of Robotics and Automation, vol. RA-3, no 4, August 1987, p. 323-344

[19] R. Szeliski, "Rapid Octree Construction from Image Sequences", CVGIP: Image Understanding, Vol. 58, No. 1, July, pp. 23-32, 1993

[20] L. Davis, E. Borovikov, R. Cutler, D. Harwood, T. Horprasert, "Multi-Perspective Analysis of Human Action", Proc. 3^{rd} Intl. Workshop on Cooperative Distributed Vision, Kyoto, Japan, November 1999

[21] E. Hunter, "Visual Estimation of Articulated Motion using the Expectation-Constrained Maximization Algorithm", PhD Dissertation, University of California, San Diego, 1999

[22] Y. Bar-Shalom, T. E. Fortmann, Tracking and Data Association, Academic Press, 1987

[23] R. Murray, Z. Li, S. S. Sastry, A Mathematical Introduction to Robotic Manipulation, CRC Press, 1993