Combining 2D Feature Tracking and Volume Reconstruction for Online Video-Based Human Motion Capture

Christian Theobalt Marcus Magnor Pascal Schüler
Hans-Peter Seidel
Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85, Saarbrücken, Germany
{theobalt|magnor|schueler|hpseidel}@mpi-sb.mpg.de

Abstract

The acquisition of human motion data is of major importance for creating interactive virtual environments, intelligent user interfaces, and realistic computer animations. Today's performance of off-the-shelf computer hardware enables marker-free non-intrusive optical tracking of the human body. In addition, recent research shows that it is possible to efficiently acquire and render volumetric scene representations in real-time. This paper describes a system to capture human motion at interactive frame rates without the use of markers or scene-intruding devices. Instead, 2D computer vision and 3D volumetric scene reconstruction algorithms are applied directly to the image data. A person is recorded by multiple synchronized cameras, and a multilayer hierarchical kinematic skeleton is fitted to each frame in a two-stage process. We present results with a prototype system running on two PCs.

1 Introduction

Recently, the field of human motion capture has brought together researchers from computer vision and computer graphics. The acquisition of human motion data is a prerequisite for the control of artificial characters in virtual reality and augmented reality applications [1], as well as in computer animation and video games [22]. The analysis of human motion, e.g. gesture recognition, can be used for intelligent user interfaces and automatic monitoring applications [7].

Existing optical motion capture systems only work in a very constrained scene setup. The person to be tracked has to wear markers, and many cameras have to observe the scene from different viewpoints to prevent occlusions [9, 12]. The first marker-free vision-based motion capture systems have only recently become feasible thanks to increasing computational power of off-the-shelf hardware. Non-real time approaches [13, 8, 26] use features extracted from video frames to fit simple kinematic skeleton models with volumetric limb representations to human body poses. Image differencing [14] and silhouette skeletonization [10] are also used to fit simple kinematic models to video streams. The use of TV image sequences for the acquisition of articulated motion is presented in [32]. In [24] an implicit-surface human body model is fitted to video material. More recently, Bregler et. al. use the combination of optical flow, a probabilistic region model, and the twist parameterization for human body joints to fit a kinematic model to video footage [3]. Existing real-time systems use comparably simple models, such as probabilistic region representations and probabilistic filters for tracking [31], or combine feature tracking and dynamic appearance models [11]. Unfortunately, these approaches cannot support sophisticated human body models like kinematics skeletons or dynamic body representations.

At the same time, a new method for the acquisition and efficient rendering of volumetric scene representations obtained from multiple camera views, known as shape-from-silhouette or the visual hull [17], has been proposed. Early approaches in the field construct discrete three-dimensional grids of volume elements (voxels) from a set of silhouette images of a scene, a method known as voxel carving or volume intersection [28, 25, 5]. More recently, it was shown that a polyhedral representation of the visual hull can be acquired and rendered in real-time [20]. An image-based approach to visual hull construction samples and textures visual hulls along a discrete set of viewing rays [21]. State-of-the-art graphics hardware can be used to accelerate the construction of slices of the visual hull [18]. Most work focuses on improving the quality of the reconstructed scene [15].

Only recently, methods have been presented that use real-time volume reconstruction to capture human motion.

In [4], an Expectation-Maximization-like ellipsoid fitting procedure is used, and in [19], a force-field exerted by the voxels is used to find the configuration of a kinematic chain model.

In this paper, we present a new approach to full-body human motion capture which combines efficient color-based optical tracking of human body features with the voxel-based reconstruction of the person's visual hull from multiple camera views. The system consists of an online and an off-line component. In the online component, the visual hull is reconstructed from four camera views, and the 3D positions of the head, hands and feet are automatically identified and tracked. In the off-line component, a simplified humanoid kinematic skeleton is fitted to these 3D positions using a saved stream of visual hulls and 3D feature locations.

A second layer extends the skeleton by more detailed arm and leg representations and cylindrical volume samples for modeling the volumetric extent of the extremities. The more detailed model is fitted to the visual hull in order to recover the exact pose of arms and legs. This hybrid approach benefits from the strengths of both sources of information (features+volume model) and compensates for individual weaknesses of the separate techniques.

The structure of the paper follows the structure of our system. Sect. 2 starts with an overview of the system architecture. The real-time component of our system is described in Sect. 3 The off-line component is described in Sect. 4. Results with the prototype implementation are presented in Sect. 5. The paper concludes with a summary and a discussion of future work in Sect. 6.

2 System Overview

2.1 Software architecture

The online component of our system is distributed on two PCs. The software is implemented as a distributed client-server application (Fig. 1). Currently, there are 2 clients, each of which is running on a 1 GHz single processor Athlon PC connected to two Sony™ DFW-V500 IEEE1394 video cameras that run at a resolution of 320x240 in color mode. Both clients perform a background subtraction (Sect. 3.2), as well as the computation of a partial visual hull for the 2 connected camera views in real-time. Additionally, the client controlling the two front view cameras identifies and tracks the positions of hand, head and feet at interactive frame rates (see Sect. 3.2 and Sect. 3.3). The partial visual hulls from both clients are transferred to the server application which builds the full visual hull and renders it using OpenGL. The server also sends the trigger signals to the cameras for synchronization. The software architecture scales easily to more cameras and more clients.

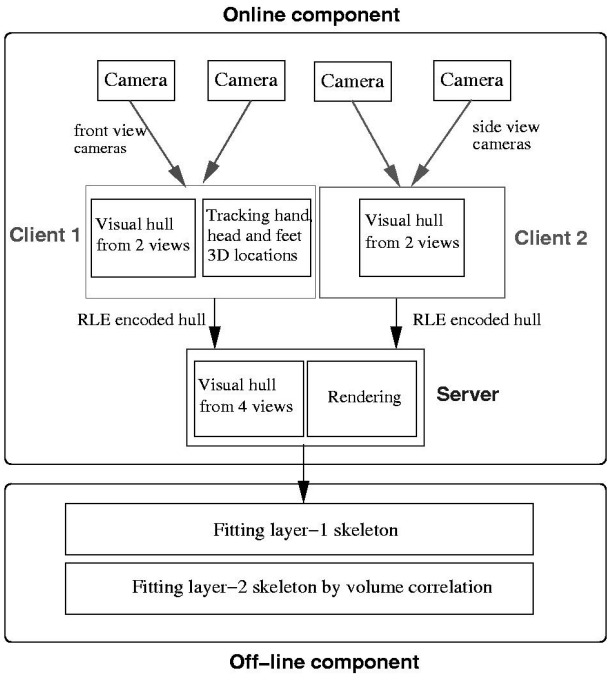

Figure 1. System architecture

The model fitting is currently implemented as a separate application which works with recorded visual hulls and 3D locations acquired with the online system.

2.2 Scene setup

The person to be tracked is supposed to move inside a confined volume. The scene is observed from four synchronized cameras from different directions. We require that there are two cameras observing the person from two nearby positions in front (Fig. 2). The person moves barefooted and needs to face these cameras allowing only limited rotation around the vertical body axis. The cameras are calibrated using Tsai's method [30].

3 Online System

3.1 Initialization

In the first frame, the person is supposed to stand in an initialization position, facing the two frontal cameras, with both legs next to each other and spreading the arms horizontally away to the side at maximal extent.

3.2 Silhouette Segmentation

The segmentation step consists of two parts. First, the person's silhouette is separated from the background in each

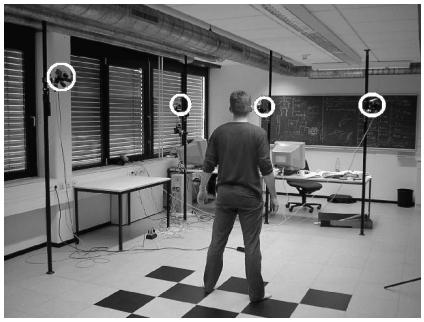

Figure 2. Scene setup: Camera studio, calibration pattern on the floor, 4 cameras marked by circles

camera perspective. Then, the silhouettes obtained from the front-view cameras are segmented in order to identify hand, feet and head. The former step is performed for every time step, the latter is performed for the initial frame only.

Separating the person from the background is done by using a background distribution for each camera perspective consisting of a mean image $\mu(x, y)$ and a standard-deviation image $\sigma(x, y)$. These are generated from several consecutive video frames of the static background scene. For the

Figure 3. Video frame after background subtraction (l) and corresponding silhouette (r)

silhouette extraction a method originally proposed in [4] is used which proves to be robust against shadows cast by the person on the floor and the walls. If a pixel differs in at least one color channel by more than an upper T_u threshold from the background distribution

$$\|p(x, y) - \mu(x, y)\| > T_u \cdot \sigma(x, y)$$

it is classified as foreground. If its difference from the background statistics is smaller than the lower threshold T_l in all channels it is surely a background pixel. All pixels which fall in between these thresholds are possibly in shadow areas. Shadow pixels are classified by a large change in intensity but only small change in hue. If $p(x, y)$ is the color vector of the pixel to be classified, and $\mu(x, y)$ is the corresponding background pixel color vector, their difference in hue is

$$\Delta = \cos^{-1}\left(\frac{p(x, y) \cdot \mu(x, y)}{\|p(x, y)\|\|\mu(x, y)\|}\right)$$

If $\Delta > T_{angular}$ the pixel is classified as foreground, else as shadow. A 0/1-silhouette image for each camera is computed this way.

The binary silhouette images of the person standing in the initialization position seen from the two front view cameras are segmented using a Generalized Voronoi Diagram (GVD) decomposition (see Fig. 5). Often used in free space segmentation of cognitive topological maps of mobile robots [27, 29, 16], the Generalized Voronoi Diagram is the set of all points in the silhouette which is equidistant to at least two silhouette boundary points.

The GVD point set can be used to segment the silhouette into distinct regions by searching for critical points, i.e. points locally minimizing the clearance to the silhouette boundary. These points are used as centers for border lines between adjacent regions in the silhouette. These lines connect the two boundary pixels closest to the critical point (Fig. 4). Since in the silhouette the boundaries to the head, hand and feet are identified by constrictions, the algorithm nicely segments these parts from the rest of the body. This way, the position and the regional extent of these body parts are extracted.

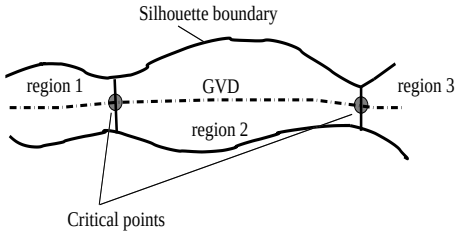

Figure 4. GVD with critical points

The connectivity of the recovered silhouette regions can be represented by a graph connecting the region centers. For the case of the human silhouette in the initialization position, the five terminating nodes in the connectivity graph correspond to the head, the hands and the feet of the person.

3.3 Tracking head, hands and feet

To track the motion of body parts in 2D, we implemented a fast tracking strategy. We use a continuously adaptable mean-shift algorithm which is capable of tracking the mean of dynamically changing probability distributions, originally developed for face tracking [2, 6]. From

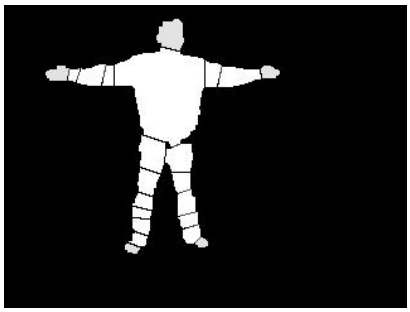

Figure 5. Silhouette segmented by Generalized Voronoi Diagram decomposition

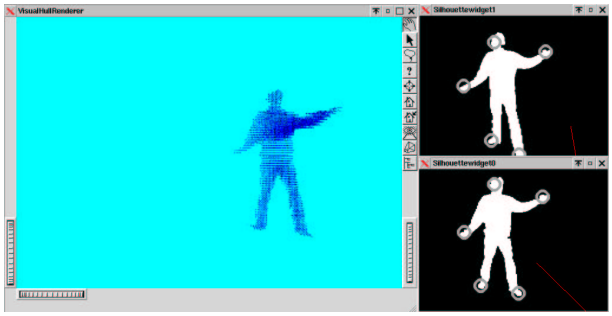

Figure 6. Screen-shots of server showing the visual hull (l) and silhouettes with tracked feature locations (r)

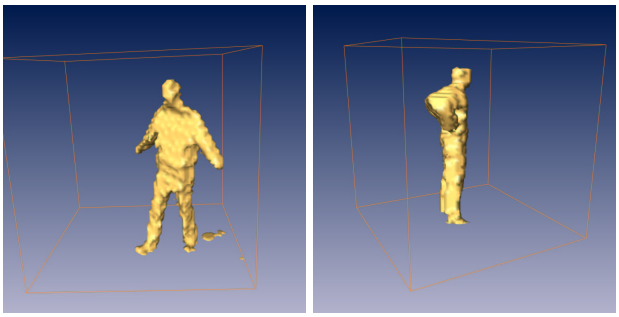

Figure 7. Visual hulls from 4 camera views with scene bounding box

the segmentation step, it is known which pixels belong to the head, the hands and the feet for both front camera views at $t = 0$. The HSV color is the principal cue used for tracking body parts. The color range of human skin in the camera view is different depending on lighting conditions and camera adjustment. Since the locations and extent of the head, the hands and the feet are known, their color values in the image plane can be used to compute an average skin color for each frontal camera view, C_1 and C_2. These values are used to define tolerance intervals in color space, $[C_1 - tolerance\_low, C_1 + tolerance\_high]$ and $[C_2 - tolerance\_low, C_2 + tolerance\_high]$. For the colors in these intervals, color histograms H_1 and H_2 are computed based on the video frames with the person in initialization position.

After the first video frames, the algorithm proceeds as follows. For each new frame, an intermediate gray-scale image is computed that contains for each pixel an approximation to the probability of belonging to one of the desired regions. This can be done by back-projecting the histograms H_1 and H_2 into the corresponding video frames after background subtraction. Alternatively, we can simply filter out all pixels in the allowed color interval and set all pixels passing the test to the maximum pixel value. In practice, this leads to fast convergence of the tracking algorithm.

We use a separate continuously adaptable mean shift tracker for each of the five body parts in both front views that takes the intermediate gray-scale images as input. Within a limited rectangular search window, gradient information is used to iteratively converge to the mean of the probability distribution (see [2] for details). Starting with the mode position in the previous frame, the center of the search window after convergence is taken as the new body part position in the current frame. At time step $t = 0$ the trackers are initialized with the center positions found during the Voronoi decomposition step.

The whole procedure is run for each video frame acquired from the two front view cameras. Fig. 6 shows a screen-shot of our system where the tracked body parts are marked by circles. We assume that the colors of the head, the hands and the feet are sufficiently different from the colors of the clothes that the person wears. Head, hands and feet colors need to be similar in HSV space for our method to work properly. Requiring that the person moves barefooted is the easiest way to fulfill this constraint. The drawback of the method is that in case of overlapping body parts, the trackers can be mislead.

Once their locations in the front camera views are determined, the 3D positions of the body parts are computed by triangulation. We assume that the tracked centroids of the hands correspond to the projected wrist joint locations, the centroids of the feet to the ankle joint locations, and the centroid of the head to the model root joint.

3.4 Volume reconstruction

From the silhouettes of the moving person, we reconstruct a voxel-based approximation to the visual hull at

every time step. Our approach adapts the voxel carving method and is similar to the algorithms presented in [4] and [19].
The box in space in which the person is allowed to move is subdivided into a regular grid of volume elements.

In our distributed implementation, each voxel is simultaneously projected into the views of the two cameras connected to one client computer. If it re-projects into the silhouette of the person in both views, it is classified as occupied space. The partial hulls from each client i, $\mathcal{V}_i$, are run-length-encoded and transferred to the server application via LAN. On the server, the complete visual hull $\mathcal{VH}$ is constructed by intersecting the volumes, $\mathcal{VH} = \mathcal{V}_1 \bigcap \mathcal{V}_2$. The intersection can be efficiently implemented using bitwise boolean operators. The voxel projections can be precomputed for each static camera view. Two example visual hulls reconstructed from four camera views can be seen in Fig. 7.

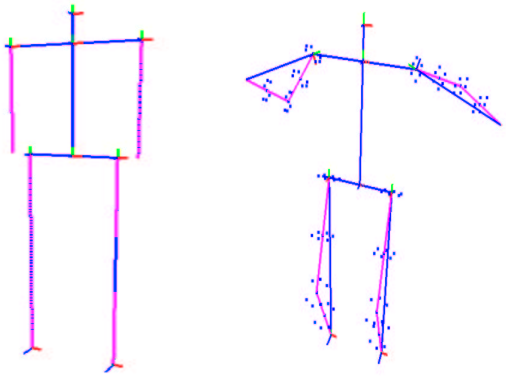

Figure 8. Skeleton layer 1 (l), skeleton layer 2 (r)

4 Off-line system

4.1 Initialization

The model fitting application (off-line component in Fig. 1) takes visual hulls and 3D feature locations that are saved by the online system as input. The dimensions of the kinematic skeleton need to be adjusted to the body dimensions of the moving person. This is either done by manually measuring the limb lengths and loading them into the application, or by an interactive step. In this step the user marks shoulder, hip, elbow and knee locations in the two camera frames showing the person in the initialization position from front. Together with the tracked positions, the 3D locations of all joints can be computed and the lengths of the body segments are derived. The thicknesses of the arms and legs are set by the user.

4.2 The Skeleton

The model uses a 2-layer kinematic skeleton to which volume samples representing the extension of body parts are attached. The first layer of the skeleton consists of a structure of 10 bone segments and 7 joints (including the root). The rotation parameters for the joints and the translation of the model sum up to 20 degrees of freedom for the skeleton.
The second layer extends the layer-1 structure by upper arm and forearm segments, as well as upper leg and lower leg segments (Fig. 8). Each of these segments consists of two bones attached to a layer-1 arm or leg segment. These bones are connected by a 1-degree-of-freedom revolute joint that serves as a simplified model for the elbow and the knee joint. As an example, this structure is shown for the arm in Fig. 9. The lengths of the upper arm l_{upper} and lower arm l_{lower} are fixed and obtained from the initialization step. During model fitting, the length of the layer-1 segment l_{whole} varies, and the joint angle ϕ of the elbow is fully determined by the cosine theorem (Fig. 9) . As an additional degree of freedom, the rotation ρ around the axis along the layer-1 segment is introduced. For the whole layer-2 skeleton, 4 additional degrees of freedom result since the ϕ-angles are fully determined by the layer-1 skeleton.
To model the extent of limbs, volume samples attached to the arm and leg structures on layer 2. These are point samples taken from a cylindrical volume around the extremities (see Fig. 8). This construction for legs and arms is the precondition for our model fitting strategy on layer 2.

The skeleton structure is hierarchical, and each segment has its own local coordinate system. Revolute joints are parameterized by one angle. Higher degrees-of-freedom joints can be parameterized by ZYZ-Euler angles or Quaternions [23], or the joint transformation can be set directly in form of the corresponding matrix (Sect. 4.3.1). In order to apply our model parameters to a skeleton containing the full set of joints and bones on just one layer (e.g. H-Anim), an additional step has to be taken. The rotation matrices defined by the shoulder and hip joints have to be multiplied by matrices rotating the layer-1 arm or leg segments onto the corresponding upper arm and leg segments in the local coordinate system.

4.3 Model fitting

The model is fitted for each video frame in a two-stage process which is described in the following.

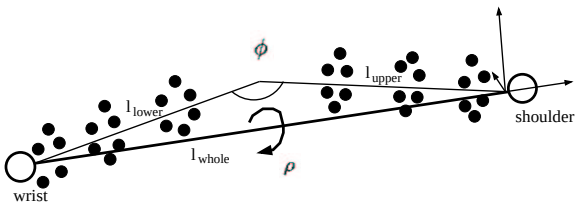

Figure 9. Arm structure of model layer 2

4.3.1 Fitting the first skeleton layer

The 2D feature tracking (Sect. 3.3) reports a set of 3D goal locations for the head, the hands and the feet. From the initialization, the skeleton dimensions are known. By assuming that the person is moving with an upright upper body and with only slight rotation around the vertical body axis, all joint locations for the layer-1 skeleton are known for every time step. For each video frame, the first layer of the kinematic skeleton is fitted into this point set. This is done by translating the model root (located at the head) to the current 3D head position. The distances between the left and right shoulder and wrist as well as the left and right hip and ankle are computed, and the lengths of the corresponding layer-1 segments are adapted to these values. The skeleton is represented as a hierarchical kinematic chain. Each joint corresponds to a rotation $\mathcal{R}_j$ and a translation $\mathcal{T}_j$, which can be represented as a combined matrix $\mathcal{A}(R_j, T_j)$ in homogeneous coordinates. Knowing the coordinates of a point in the joint coordinate system P_j, its world coordinates can be found by $P_w = \left(\prod_{i=0}^{n-1} \mathcal{A}(\mathcal{R}_i, \mathcal{T}_i)\right) \cdot P_j$, i.e. by multiplying P_j by the preceding n joint transformations in the skeleton hierarchy. In our case, all the joint transforms apart from the shoulder and hip transforms are known. For the latter, only the translations $\mathcal{T}_j$ are known. The unknown rotations can be easily computed which is shown for the example of the left arm (Fig. 10). Let l_{whole} be the length of the left arm segment on layer-1 and P_w be the world coordinates of

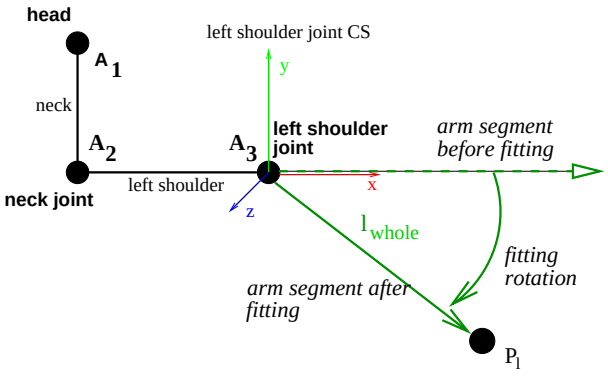

Figure 10. Layer-1 fitting

the left hand. By assuming that initially the rotation of the shoulder is the identity matrix $\mathcal{I}$, the coordinates of the left hand in shoulder space P_j can be easily found. If in case of no rotation the left shoulder segment is coincident in direction with the x-axis of the shoulder coordinate system, the current rotation of the shoulder R_j is identical to the rotation matrix which rotates the shoulder coordinate system's x-axis onto the vector $\vec{P_j}$ from the shoulder origin to the current left hand coordinates. This matrix is straightforward to compute. The whole layer-1 fitting is performed in real-time (Sect. 5).

4.3.2 Fitting the second skeleton layer

Once the model parameters are found for the first skeleton layer, the additional degrees of freedom of the second model layer are recovered by using the visual hull information. During the fitting step of model layer 1, the lengths of arm and leg segments are recomputed for each time step. Knowing the lengths of the additional two segments of arms and legs enables computing the elbow and knee joint angles (ϕ in Fig. 9) directly using the cosine theorem. In order to find the additional angle of the layer-1 arm and leg segments (see also 4.2), a maximal overlap between the set of volume samples attached to the layer-2 model and the voxel data obtained from the visual hull is searched. This step is performed for those arm and leg segments with a noticeable bending of the elbow and knee joints, i.e. only if the length of the corresponding layer-1 segment is below a threshold. The search procedure is as follows, using the arm segment as an example:

Making use of the temporal coherence, we start with the rotation of the arm in the previous frame, $\rho(t-1)$, and rotate the arm segment to ν equidistant angles ξ_l in the interval $[\rho(t-1) - s, \rho(t-1) + s]$, with s defining the search neighborhood size. For each such orientation, ξ_l, a quality measure for the overlap between the volume samples and the visual hull, $match_l$, is computed which is the higher the better the model fits to the voxel set. For each volume sample, the corresponding voxel it lies in is computed (see Fig. 11) . If n is the number of these voxels which belong to the visual hull (i.e. are filled), then n^k is the overlap match score for the current configuration ξ_l, where a value of $k = 4$ is used for best performance.

Using the set of ν match scores, the final rotation $\rho(t)$ of the arm segment is found by computing the center of gravity of the set $\Xi = \{\xi_l \times match_l \mid l = 1, \ldots, \nu\}$, the set of angles ξ_l each multiplied by its corresponding match score

$$\rho(t) = \frac{1}{\sum_{l=0}^{\nu-1} match_l} \sum_{l=0}^{\nu-1} \xi_l \times match_l.$$

The procedure for the leg segments is the same. Although the difference between match scores for neighboring ξ_l can

be very small, this approach still allows us to recover small changes in rotation from $t-1$ to t.

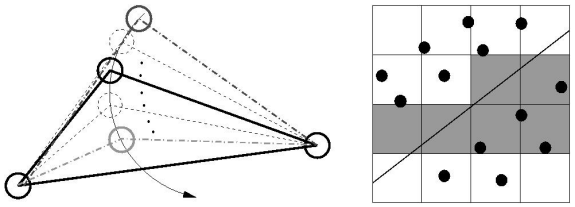

Figure 11. (l) Testing rotations between search interval bounds (stippled lines), (r) slice through voxel volume showing overlap between samples and voxels

The positions of the cameras are crucial for the quality of the visual hull. Typical artifacts due to bad camera positions are occlusion artifacts observable as too thick arms or legs. These artifacts form thin voxel planes in which the arm or leg must lie. Our approach can still recover the correct arm and leg configurations in the presence of these visual hull errors. A camera looking at the scene from the top is not required, and even with our 4 cameras looking from the side, robust fitting is possible.
For each video frame, the 2-step fitting procedure results in a set of model parameters describing the body pose. These rotation and translation parameters can be easily used to animate any artificial character based on a similarly structured skeleton (see also Sect. 4.2).

5 Results

The system is tested on several sequences of a person moving in the camera capture setup shown in Fig. 2. Currently, two Athon 1GHz single-processor PCs are used. One PC runs a client and the server application simultaneously, resulting in high workload on this machine.

Fig. 12 shows two frames out of a sequence of 170 frames. In both pictures, the spheres mark the tracked 3D locations of the head, the hands and the feet. One can see that the complete layer-2 skeleton is nicely fit into the visual hulls. Using 4 cameras, the combined visual hull reconstruction and feature tracking runs at 4 fps for a $64\times64\times64$ volume and approximately 6.5 fps for a $32\times32\times32$ voxels. Measurements show that currently feature tracking consumes over 30% of total computation time. Furthermore, we experience a network overhead in our current implementation, since the frame rates of one client running independently without sending data to the server can reach up to 19 fps (measured using the internal camera trigger).

The model fitting process which works on recorded sequences takes 0.8s (for all four layer-2 arm and leg segments) for one visual hull using 256 volume samples and 15 angular search steps for arms and legs. Fitting the layer-1 model can be done at almost no cost ($<$1 ms). It turns out that even with only 4 cameras, the system can robustly fit the kinematic skeleton to the motion data. Even visibility artifacts resulting in too thick arms don't mislead the model fitting. The knowledge of correct head, hands and feet positions also makes possible correct model fitting in cases that are problematic for other approaches, such as if the arms are very close to the body. A dynamic motion model for the tracked features will further extend the range of allowed motions. More results including videos of the system in action can be found at http://www.mpi-sb.mpg.de/∼theobalt/VisualHullTracking.

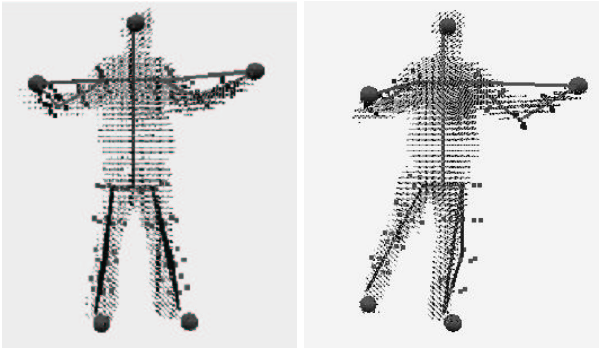

Figure 12. Skeleton fitted to visual hulls (rendered as point sets) of a moving person

6 Conclusion

In this paper we present a method that combines color-based feature tracking and 3D scene reconstruction from silhouettes for human motion capture. The algorithm enables fast fitting of a simplified kinematic model to the video footage. Additional degrees of freedom that are hard to recover using pure feature tracking are computed by using a second layer of the kinematic model. This layer features a special representation for arm and leg segments including volume samples attached to the skeleton. The presented method uses the reconstructed volumetric visual hull to find the correct configuration of the kinematic skeleton at every time step. First results of a prototype implementation capturing the motion of a human performer demonstrate the system's ability to fit the skeleton in real-time and a more detailed skeleton at near interactive frame rates. This hybrid approach of combining feature tracking and volume reconstruction is found to be capable of correctly finding human

body configurations even in the presence of typical visibility artifacts in the visual hull.

In the future, the model fitting step and the visual hull reconstruction will be integrated into one real-time application. The use of a dynamic motion model for feature tracking is also another area of our research. Moreover, we look into using the visual hull for the recovery of a wide range of torso orientations to allow complex upper body motion. Furthermore, the application of this new approach for character animation and the control of avatars using H-Anim models will be investigated.

References

[1] C. Babski and D. Thalmann. Real-time animation and motion capture in web human director (WHD). In *Web3D*, pages 139–145, 2000.

[2] G. Bradski. Computer vision face tracking as a component of a perceptual user interface. In *IEEE Workshop of Applications of Computer Vision*, pages 214–218, 1998.

[3] C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *Computer Society Conference on Computer Vision and Pattern Recognition 98*, pages 8–15, 1998.

[4] K. M. Cheung, T. Kanade, J.-Y. Bouguet, and M. Holler. A real time system for robust 3D voxel reconstruction of human motions. In *Proceedings of the 2000 IEEE Conference on Computer Vision and Pattern Recognition (Computer Society Conference on Computer Vision and Pattern Recognition 2000)*, volume 2, pages 714 – 720, June 2000.

[5] P. Eisert, E. Steinbach, and B. Girod. Automatic reconstruction of stationary 3-D objects from multiple uncalibrated camera views. *IEEE Transactions on Circuits and Systems for Video Technology: Special Issue on 3D Video Technology*, 10(2):261–277, Mar. 2000.

[6] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, Boston, 1990.

[7] D. Gavrila. The visual analysis of human movement. *Computer Vision and Image Understanding*, 73(1):82–98, January 1999.

[8] D. Gavrila and L. Davis. 3D model-based tracking of humans in action: A multi-view approach. In *Computer Society Conference on Computer Vision and Pattern Recognition 96*, pages 73–80, 1996.

[9] M. Gleicher. Animation from observation: Motion capture and motion editing. *Computer Graphics*, 4(33):51–55, November 1999.

[10] Y. Guo, G. Xu, and S. Tsuji. Tracking human body motion based on a stick-figure model. *Journal of Visual Communication and Image Representation*, 5(1):1–9, 1994.

[11] I. Haritaoglu, D. Harwood, and L. Davis. W4: Who? When? Where? What? A real time system for detecting and tracking people. In *Conference on Automatic Face and Gesture Recognition 98 (Tracking and Segmentation of Moving Figures)*, pages 222–227, 1998.

[12] L. Herda, P. Fua, R. Plaenkers, R. Boulic, and D. Thalmann. Skeleton-based motion capture for robust reconstruction of human motion. In *Proceedings of Computer Animation 2000*. IEEE CS Press, 2000.

[13] D. Hogg. Model-based vision : a program to see a walking person. *Image and Vision Computing*, 1(1):5–20, 1983.

[14] Y. Kameda, M. Minoh, and K. Ikeda. Three dimensional motion estimation of a human body using a difference image sequence. In *Proceedings of the Asian Conference On Computer Vision '95*, pages 181–185, 1995.

[15] K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. Technical Report TR692, 1998.

[16] J. C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[17] A. Laurentini. The visual hull concept for silhouette-based image understanding. *Pattern Analysis and Machine Intelligence*, 16(2):150–162, February 1994.

[18] B. Lok. Online model reconstruction for interactive virtual environments. *Symposium on Interactive 3D Graphics, pp. 69-72, 2001*, 2001.

[19] J. Luck and D. Small. Real-time markerless motion tracking using linked kinematic chains. In *Proceedings of the International Conference on Computer Vision, Pattern Recognition and Image Processing 2002 (CVPRIP02), in cooperation with JCIS 2002*, 2002.

[20] W. Matusik, C. Buehler, and L. McMillan. Polyhedral visual hulls for real-time rendering. In *Proceedings of 12th Eurographics Workshop on Rendering*, pages 116–126, 2001.

[21] W. Matusik, C. Buehler, R. Raskar, S. J. Gortler, and L. McMillan. Image-based visual hulls. In *Siggraph 2000, Computer Graphics Proceedings*, pages 369–374, 2000.

[22] A. Menache. *Understanding Motion Capture for Computer Animation and Video Games*. Morgan Kaufmann, 1995.

[23] Murray, R. M., Li, Zexiang, Sastry, and S. Shankar. *A mathematical introduction to robotic manipulation*. CRC Press, 1994.

[24] R. Plankers and P. Fua. Tracking and modeling people in video sequences. *Computer Vision and Image Understanding*, 81(3):285–302, March 2001.

[25] M. Potmesil. Generating octree models of 3d objects from their silhouettes in a sequence of images. *Computer Vision, Graphics and Image Processing*, 40:1–20, 1987.

[26] K. Rohr. Incremental recognition of pedestrians from image sequences. In *Computer Society Conference on Computer Vision and Pattern Recognition 93*, pages 8–13, 1993.

[27] P. Rowat. *Representing the Spatial Experience and Solving Spatial Problems in a Simulated Robot Environment*. PhD thesis, University of British Columbia, 1979.

[28] R. Szeliski. Rapid octree construction from image sequences. *Computer Vision, Graphics, and Image Processing. Image Understanding*, 58(1):23–32, 1993.

[29] S. Thrun. Learning maps for indoor mobile robots. *Artificial Intelligence*, 99(1):21–71, 1998.

[30] R. Tsai. An efficient and accurate camera calibration technique for 3d machine vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'86)*, pages 364–374, June 1986.

[31] C. R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.

[32] J. Zheng and S. Suezaki. A model based approach in extracting and generating human motion. In *Proceedings of the International Conference on Pattern Recognition*, 1998.