Shape-From-Silhouette Across Time Part II: Applications to Human Modeling and Markerless Motion Tracking

Kong-man (German) Cheung, Simon Baker and Takeo Kanade

The Robotics Institute
Carnegie Mellon University

$*$ $*$ $*$ **Systems and Experiments Paper** $*$ $*$ $*$

Abstract

In Part I of this paper we developed the theory and algorithms for performing Shape-From-Silhouette (SFS) across time. In this second part, we show how our temporal SFS algorithms can be used in the applications of human modeling and markerless motion tracking. First we build a system to acquire human kinematic models consisting of precise shape (constructed using the temporal SFS algorithm for rigid objects), joint locations, and body part segmentation (estimated using the temporal SFS algorithm for articulated objects). Once the kinematic models have been built, we show how they can be used to track the motion of the person in new video sequences. This marker-less tracking algorithm is based on the Visual Hull alignment algorithm used in both temporal SFS algorithms and utilizes both geometric (silhouette) and photometric (color) information.

Keywords: Human Kinematic Modeling, Markerless Motion Capture, Articulated Human Tracking, 3D Reconstruction, Shape-From-Silhouette, Visual Hull, Stereo, Temporal Alignment.

Contact Information

Second Revision

Paper Title: Shape-From-Silhouette Across Time Part II:
Applications to Human Modeling and Markerless Motion Tracking

Paper Type: Systems and Experiments Paper

Authors: Kong-man (German) Cheung, Simon Baker and Takeo Kanade

Contact Name: Simon Baker

Contact Address:

Carnegie Mellon University,
5000 Forbes Avenue, Robotics Institute
EDSH 213, Pittsburgh, PA 15213, USA

Contact Email: simonb@cs.cmu.edu

Contact Phone: +1 (412) 268-5746

Contact Fax: +1 (412) 268-5571

Author Emails: german+@cs.cmu.edu, simonb@cs.cmu.edu, tk@cs.cmu.edu

Number of Manuscript Pages: 37

Number of Supplementary Videos: 1 (SFSAT_Applications.mpg)

1 Introduction

Human kinematic modeling and motion tracking are difficult problems because of the complexity of the human body. Despite the difficulties, these problems have received a great deal of attention recently due to the large number of applications. Having a precise 3D kinematic (shape and joint) model of specific human is very useful in a variety of different situations. For example, such a model could be used in the garment/furniture manufacturing industry to make clothes/furniture that are tailored to the body shape and motion range of the individual. A collection of such models can be used to generate valuable statistics of body shape information (such as arm length, shape, etc.) of people from different races for anthropological studies. Likewise, accurate human motion tracking is essential in a wide variety of applications. For example, in intelligent environments such as smart offices or households [SKB^{+}98, Coe98, LZG98], tracking human motion and recognizing gestures is a natural way for the computer to understand the action and intention of humans. In the field of automatic surveillance and security, it is important for computers to be able to observe suspicious people and track their actions over time. For sports science and medicine, the ability to track the body parts of athletes is critical for improving their performance during competition or for injury rehabilitation. Last but not least, the entertainment industry is another area where there is an increasing need for better human modeling and motion tracking algorithms. Accurate human kinematic models, precise tracking data are essential components for making animated virtual characters more human-like in both computer games and motion picture production.

Although there are a variety of complete systems [CYB, TTI] and algorithms [ACP03] for human body shape acquisition using laser-scanning devices, most of these systems are expensive and do not estimate the important joint information. Similarly, almost all commercial motion capture systems [MET, VIC] attach optical or magnetic markers on the person whose motion is to be tracked and use triangulation on the positions of the markers to achieve tracking. Although these systems generally produce very good results, they are invasive and difficult to use. In applications such as security, surveillance and human-computer interaction, these systems are not applicable because placing markers on the person is either impossible or undesirable. In view of these limitations of existing systems, the study of non-invasive, vision-based human modeling and tracking is vital. There are many advantages of using a vision-based approach. For example, cameras are low-cost, easily reconfigurable and non-invasive. Moreover, camera images contain both shape and texture information of the person. Finally instead of using two separate systems for human modeling and motion tracking, one multi-camera system can be used for both tasks.

In recent years researchers have proposed a variety of vision-based systems to capture the 2D and 3D shapes of human body parts [LY95, KMB94, JBY96, KM98, PFD99, FHPB00, BK00, OBBH00,CKBH00,KYS01,FGDP02,SMP03]. Moreover there are also a large number of systems for tracking human motions in video sequences [RK95, GD96, BM97, BM98, YSK$^+$98, HHD98, JTH99,DCR99,CR99a,CR99b,PRCM99,DF99,CKBH00,SDB00,SBF00,DBR00,DCR01,DC01, LC01, SC02, MTHC03, CTMS03] using a variety of model-based approaches. An extensive survey of vision-based motion tracking systems can be found in [MG01]. Among the above systems, silhouette information has been used extensively [CA96,WADP97,KM98,CA98,BK99,CKBH00, MTHC03, CTMS03] since silhouettes are easy to extract and provide valuable information about the position and shape (posture) of the person. In particular, many human shape modeling and motion tracking systems (such as [MTG97, KM98] and more recently [CKBH00, Mat01, MHTC01]) use Shape-From-Silhouette to construct 3D estimates of the body shape for modeling and tracking. None of these systems have considered combining SFS temporally, however.

In Part I of this paper [CBK04], we developed the theory of and proposed algorithms for performing Shape-From-Silhouette (SFS) across time for both rigid and articulated objects (see [CBK03b,CBK03a,Che03,CBK04][1] for the details of the algorithms). In this second part we apply our temporal SFS algorithms to build human kinematic modeling and motion tracking systems. Our systems differ from the previous work in several aspects. First, our kinematic modeling system estimates the precise 3D shape and complete skeletal information of the person using multiple camera views while most of the other systems either use monocular images to reconstruct view-

[1] [CBK04] is available at http://www.cs.cmu.edu/˜german/research/Journal/IJCV/Theory/SFSAT_Theory.pdf.

dependent 2D shape and joint models [KMB94, BK00, KYS01] or only recover imprecise 3D shape [CKBH00, MTHC03] and partial joint information [KM98, PFD99]. Secondly since we use person-specific models to perform motion tracking in new videos, our system is more accurate than other model-based systems which use generic shapes (e.g. rectangles or ellipses in 2D, cylinders or ellipsoids in 3D) to model the body parts of the person. Finally our tracking algorithm incorporates both silhouette and color information at the same time instead of using only one of the two cues [DF99, SDB00, SBF00, CTMS03].

The remainder of this paper is organized as follows. In Section 2 we describe our human kinematic modeling system. The joint skeleton of the person is first estimated using the articulated temporal SFS algorithm. The 3D body shape (voxel model) of the person is then estimated using the rigid temporal SFS algorithm and combined with the joint skeleton to form the kinematic model. In Section 3 the acquired kinematic model is used to perform marker-less motion capture of the same person in new video sequences using an image-based articulated object tracking algorithm very similar to the temporal SFS algorithms. Finally a discussion and several suggestions for future work is included in Section 4.

2 Human Kinematic Modeling

In this section we describe how to use our temporal SFS algorithms for both rigid and articulated objects to build a vision-based 3D human kinematic modeling system. The system consists of three steps: (1) constructing a joint skeleton of the person, (2) acquiring detailed shape information and (3) merging the shape and joint information to build a kinematic model. Each task in our system is described in details in Sections 2.2, 2.3 and 2.4, together with the results of applying the system to three people: SubjectE, SubjectG and SubjectS.

2.1 Related Work

The work most related to our vision-based human body kinematic information acquisition system is by Kakadiaris and Metaxas in [KM95]. They used deformable templates to segment the 2D body parts in a silhouette sequence. The segmented 2D shapes from three orthogonal view-points are then combined into a 3D shape by SFS. Although our idea of estimating the joint locations individually instead of all at once is partly inspired by their system, here we address the acquisition of motion, shape and articulation information, while [KM95] focuses mainly on shape estimation. Besides the 2D work by Krahnstoever et al. in [KYS01,KYS03] (which we have already discussed in Part I of this paper), the research group led by Fua addressed the problem of 3D human body modeling using a three-camera system [PFD99, PF01, FGDP02]. They first extract dense feature points on the surface of the body parts by manual initialization and stereo matching. The feature points are then tracked across the video sequences using a template matching technique. A flexible but complex human model consisting of deformable metaballs [Bli82] as shape primitives is then used to fit the tracked feature points through a least square framework. Though they have not demonstrated the modeling of a complete body, their approach is able to handle non-rigid deformation of the body parts. Sand et al. have also captured the non-rigid deformation of the human body skin using silhouette images [SMP03]. However, marker-based motion capture data is used in their system to estimate the joint skeleton and track the motion of the person.

2.2 Joint Skeleton Acquisition

The first task of our modeling system is to locate the joint positions of the person using the articulated object temporal SFS algorithm proposed in Part I of this paper [CBK04]. Once the joint locations have been recovered, they are aligned and registered with each other to form a complete joint skeleton of the person.

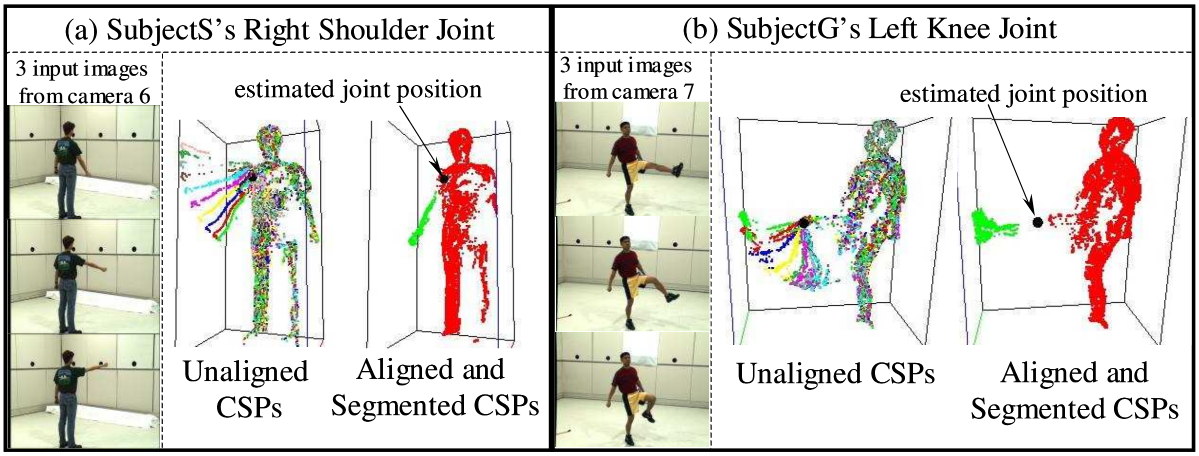

(a) SubjectS's Right Shoulder Joint
3 input images from camera 6
estimated joint position
Unaligned CSPs
Aligned and Segmented CSPs
(b) SubjectG's Left Knee Joint
3 input images from camera 7
estimated joint position
Unaligned CSPs
Aligned and Segmented CSPs

Figure 1: (a) Input images and results for (a) the right shoulder joint of SubjectS and (b) the left knee joint of SubjectG. For each joint, the unaligned Colored Surface Points (CSPs) from different frames are drawn with different colors. The aligned and segmented CSPs are shown in two different colors to show the segmentation. The estimated articulation point (joint location) is indicated by the black sphere.

2.2.1 Estimating Individual Joint Positions

Although theoretically we can estimate all of the joint positions of a person at the same time, in practice this approach suffers from local minimum due to the high dimensionality of the problem. Instead we take a sequential approach and model the joints one at a time. The person is asked to treat their body as a one-joint articulated object by moving their joints one at a time while keeping the rest of their body still. For each person, eight joint locations: left/right shoulder, elbow, hip and knee are estimated. For each joint, Colored Surface Points (CSPs) are first extracted from the video sequences. CSPs are essentially 3D color points on the surface of the object and are extracted by combining the Shape-From-Silhouette and Stereo principles (the details of how to extract CSPs and their properties can be found in [CBK04] or in [Che03]). The CSPs are then used to recover the motion of the moving body part, its segmentation and the joint location using the articulated temporal SFS algorithm described in Sections 5.5 and 5.6 of Part I [CBK04]. Some of the input images and the results for SubjectS's right shoulder joint and SubjectG's left knee joint are shown in Figures 1(a) and (b) respectively. Moreover the joint estimation results for the right leg of SubjectG and the left arm of SubjectS are shown in the movie clips **SubjectG-joints-**

rightleg.mpg and **SubjectS-joints-leftarm.mpg**[2]. Generally, the estimation of the (shoulder and elbow) joints of the arms are more accurate than the (hip and knee) joints of the legs because it is more difficult to keep the rest of the body still when moving the leg. In our system, ankle and wrist joints are not modeled (nor tracked) because the image pixels of the feet and hands are too small in our current 640x480 image resolution for accurate modeling. With cameras of higher image resolution, the ankle and wrist joints can be estimated using the methods described above.

2.2.2 Joint Registration

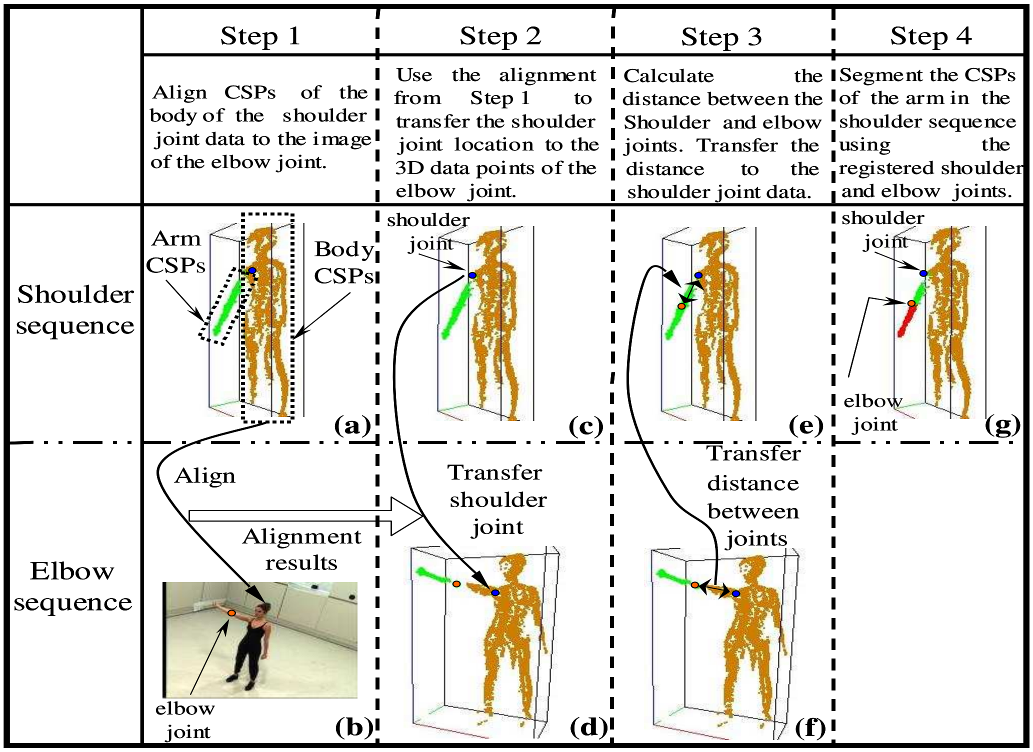
After the joints and the associated body parts (described by CSPs) are recovered individually, they are registered with respect to a reference frame to form an articulated model of the body. The registration process consists of two procedures. The first procedure involves aligning joints within each separate limb while the second procedure performs a global registration of all of the joints and body parts with respect to the reference frame. Both procedures are described below.

A. Limb Joints Alignment

Before registering all of the joints to the reference frame, the two joints of each separate limb are first aligned with each other. The limb joints alignment procedure is illustrated graphically using the right arm of SubjectE in Figure 2. The same procedure applies to the leg replacing the shoulder and elbow joints with the hip and knee joints. The idea of the procedure is to align the shoulder and elbow joints with respect to the shoulder sequence with the arm being straight. As will be seen shortly, having the joints registered with the arm being straight reduces the complexity of the subsequent global registration procedure.

Consider the shoulder and elbow joints of the right arm shown in Figure 2. The shoulder joint is estimated in a sequence of the person rotating her arm around the shoulder with the elbow joint straight while the elbow joint is estimated in a sequence of the person bending her arm at the elbow. We assume that the elbow sequence contains one frame with the elbow straight. In Step 1 (Fig-

[2] All movie clips can be found at http://www.cs.cmu.edu/~german/research/Journal/IJCV/Applications/. Lower resolution versions of some of the movies are also included in the supplementary movie SFSAT_Applications.mpg.



Step 1: Align CSPs of the body of the shoulder joint data to the image of the elbow joint.

Step 2: Use the alignment from Step 1 to transfer the shoulder joint location to the 3D data points of the elbow joint.

Step 3: Calculate the distance between the Shoulder and elbow joints. Transfer the distance to the shoulder joint data.

Step 4: Segment the CSPs of the arm in the shoulder sequence using the registered shoulder and elbow joints.

Figure 2: The four steps of the Limb Joints Alignment Procedure illustrated using the right arm of SubjectE. The same procedure applies to the legs replacing the shoulder and elbow joints with the hip and knee joints. See text for details.

ures 2(a) and (b)) we compute the 6 DOF transformation of the body from the shoulder sequence to the elbow sequence by taking the shoulder model and aligning it to the straight arm image in the elbow sequence using the rigid temporal SFS algorithm [CBK04]. In Step 2 (Figures 2(c) and (d)) we map the shoulder joint location from the shoulder sequence to the elbow sequence. In step 3 (Figures 2(e) and (f)) we compute the relative position of the elbow and shoulder joints in the elbow sequence and map it back into the shoulder sequence so that the elbow joint location is estimated in the shoulder sequence. Finally in Step 4 we segment the forearm in the shoulder sequence using the known elbow position.

B. Global Registration

Once the joints within each limb have been aligned, global registration is performed to build the final joint skeleton. The global registration for all four limbs is illustrated in Figure 3(a) and the procedure for one limb is explained using the right arm of SubjectE in Figure 3(b). For each

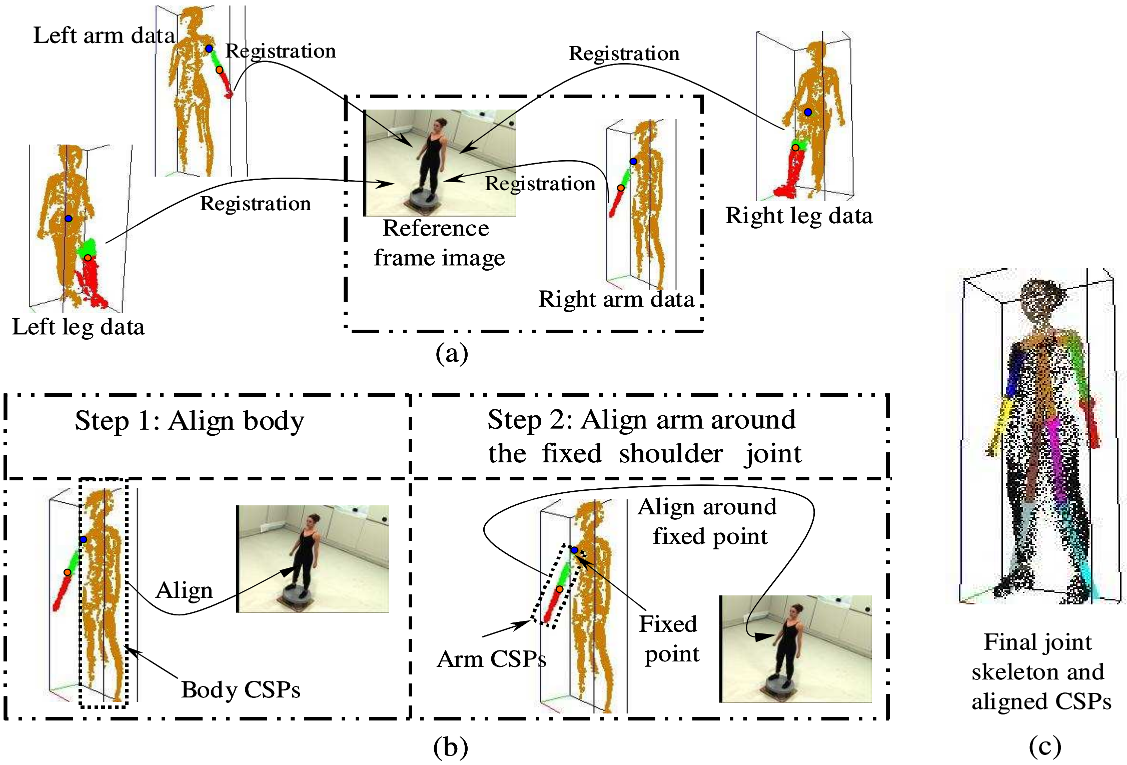


Figure 3: (a) Global joint registration. (b) For each limb, two steps are required to register the joints. (c) The final registered joint skeleton and the aligned CSPs

limb, the global registration procedure consists of two steps. The first step aligns the body CSPs against a reference frame using the rigid temporal SFS algorithm. Once the 6D motion of the body has been recovered, the position of the first limb joint (shoulder/hip) is calculated. The second step involves the alignment of the limb itself. To simplify this step, we assume that the reference frame is chosen such that the images at the reference frame are captured with all of the person's limbs straight (the choice of a good reference frame will become apparent in Section 2.3). Since the joints within each limb are already registered with the limb being straight (in the limb joint alignment procedure), the straight limb assumption of the reference frame images enables us to treat the whole limb as one rigid object rather than an articulated object with two parts. In other words, we can ignore the second limb joint (elbow/knee) and the problem becomes alignment of a rigid object around a fixed point with only 3 DOF (the rotation around the joint). The details of this algorithm are included in Section 3.2.3). Figure 3(c) illustrates the final joint skeleton of SubjectE and the registered CSPs obtained after the global registration procedure.

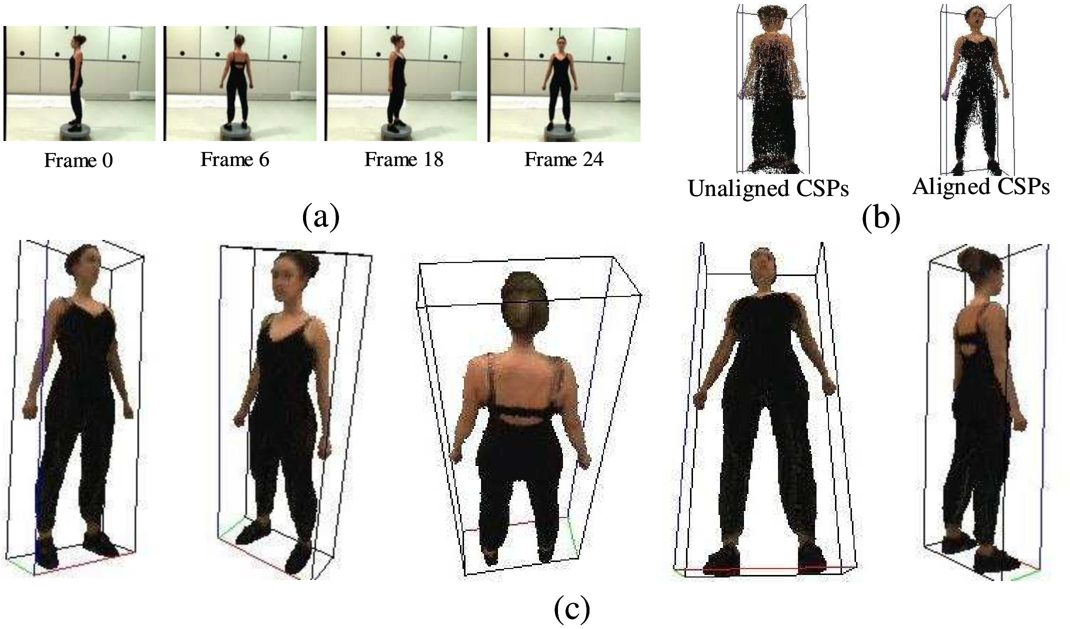
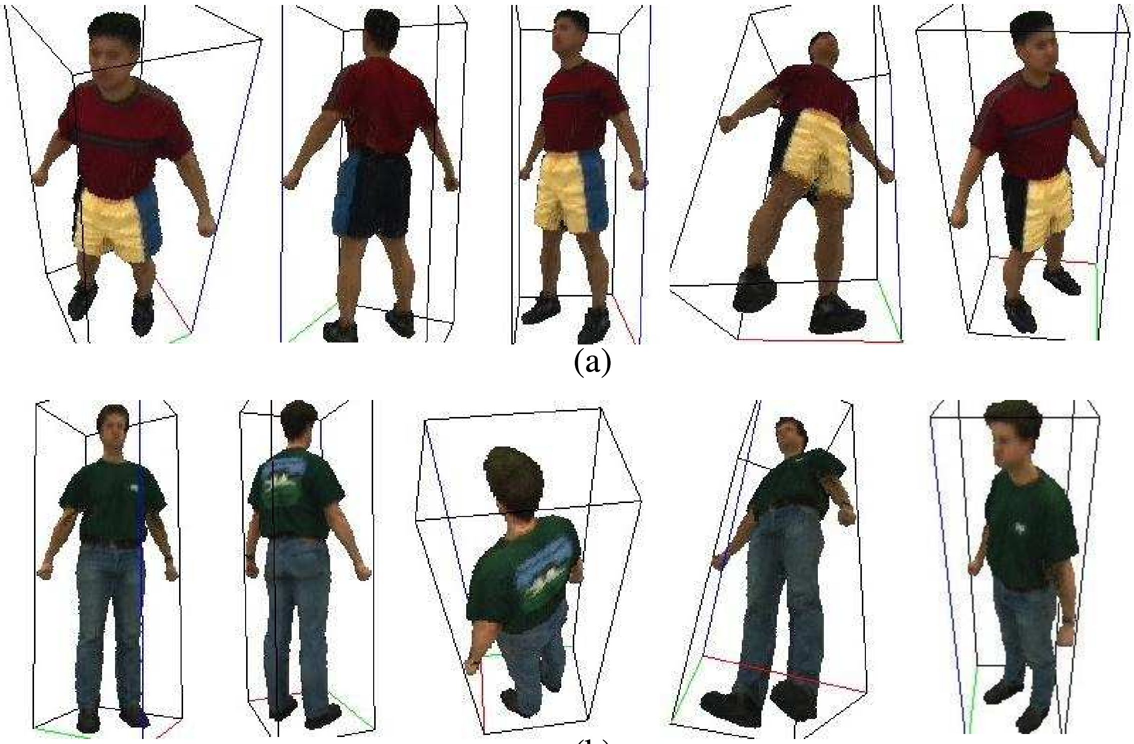


Figure 4: Results of body shape acquisition for SubjectE. (a) Four input images, (b) unaligned and aligned colored surface points from all frames, (c) refined Visual Hull of the body displayed from several views.

2.3 Body Shape Acquisition

The next task is to acquire the shape of the body. One possible choice is to use the CSPs extracted from the sequences used to estimate the individual joints. We do not use these CSPs to represent the body shape of the person because they are not uniformly distributed over the different body parts (most of the CSPs come from the torso). This non-uniformity poses a severe disadvantage when using the model in motion tracking. Moreover, due to errors in all the alignment and registration procedures, the CSPs obtained after the global registration do not represent the actual shape of the body accurately enough (see Figure 3(c)). Hence instead we build an accurate and detailed voxel model of the person using the rigid object temporal SFS algorithm proposed in [CBK04]. The centers of the *surface* voxels of the voxel model are then extracted and used to represent the shape of the person. There are two advantages of using this approach. Since the voxel model is reconstructed using a large number of silhouettes, the model is very accurate and the surface voxel centers are close approximations to points on the surface of the actual person. Moreover since the voxel centers lie on a 3D grid, they are uniformly distributed.

To build these voxel models, video sequences of the person standing on a turn table were


(a)

(b)

Figure 5: Refined voxel models of (a) SubjectG, (b) SubjectS.

captured by eight cameras with thirty frames (roughly equal to a whole revolution of the turntable) per camera. Note that there is no need to calibrate the rotation axis and speed of the turntable because our rigid body temporal SFS algorithm is able to recover the 6 DOF motion of the person on the turntable fully automatically. The person is asked to remain still throughout the capture process to satisfy the rigidity assumption. Moreover, the person is also told to keep their limbs straight so that the first frame of the sequence can be chosen as the reference frame for the global body joints registration discussed in Section 2.2.2. After applying the rigid object temporal SFS algorithm to recover the motions, a refined voxel model of the person is built using the Visual Hull refinement technique as described in [CBK04]. The centers of the surface voxels of the model are extracted and colored by back-projecting them into the color images. Some of the input images, the unaligned/aligned CSPs and the 3D refined voxel model of SubjectE are shown in Figure 4 and in the video clip **SubjectE-bodyshape.mpg**. Figure 5 illustrates the 3D models of SubjectG and SubjectS. It can be seen that excellent shape estimates of the human bodies are obtained.

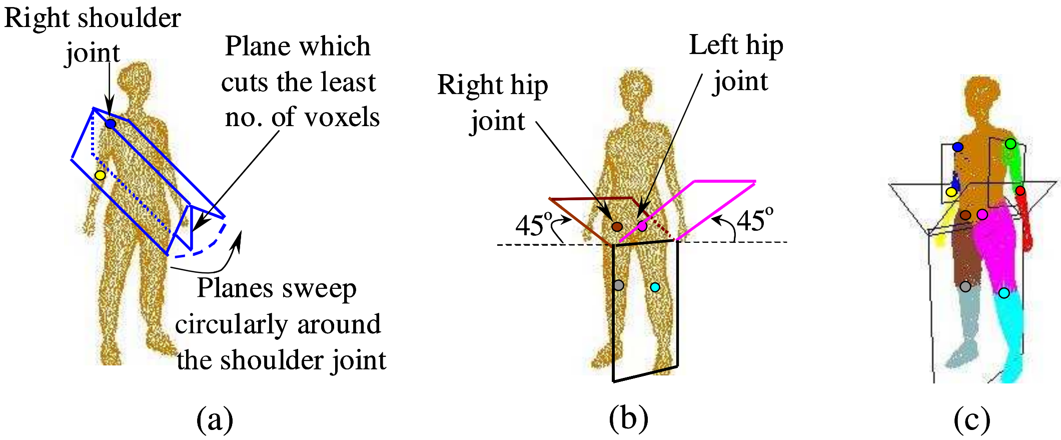

Right shoulder joint
Plane which cuts the least no. of voxels
Planes sweep circularly around the shoulder joint
Right hip joint
Left hip joint
45°
45°
(a)
(b)
(c)

Figure 6: Segmenting the voxel centers to the appropriate body parts. (a) The arm cutting planes are found by sweeping a plane circularly around the shoulder joints. The plane which cuts the least number of voxels is chosen. (b) The leg cutting planes are formed by two planes passing through the hips joints at a 45 degree angle with the horizontal, and a vertical plane which separate the legs from each other. (c) Example results with the joints, the cutting planes and the segmented voxels of the model.

2.4 Merging Shape and Joint Information

The last task is to merge the joint and shape information. Before the merge, slight modifications are made to the joint positions to enforce left and right symmetry of the joint skeleton (the asymmetry is caused by errors in joint estimation and registration). Two rules are applied: (1) The left and right shoulder joints have the same height above the ground. The same applies to the two hip joints. (2) The distance between the shoulder and elbow joints on the left arm is equal to that on the right arm. The same applies to the distances between the hip and knee joints on the legs. These two rules are reasonable because the person is told to stand upright on the turntable when the reference frame is captured. The rules can be enforced by averaging the corresponding values for the left and right sides of the body. Once the joint positions have been adjusted, they are transfered to the voxel model. Since the joints are registered with respect to the reference image used to create the voxel model, the transfer is straightforward.

The only task remaining is to segment the voxel centers to the corresponding body parts. Figure 6 illustrates an algorithm to do this based on the joint locations. First, five cutting planes are found to separate the four limbs away from the body (Figure 6(c)). Once the limb has been segmented, it can be divided into the upper and lower parts using the elbow/knee joint location. The

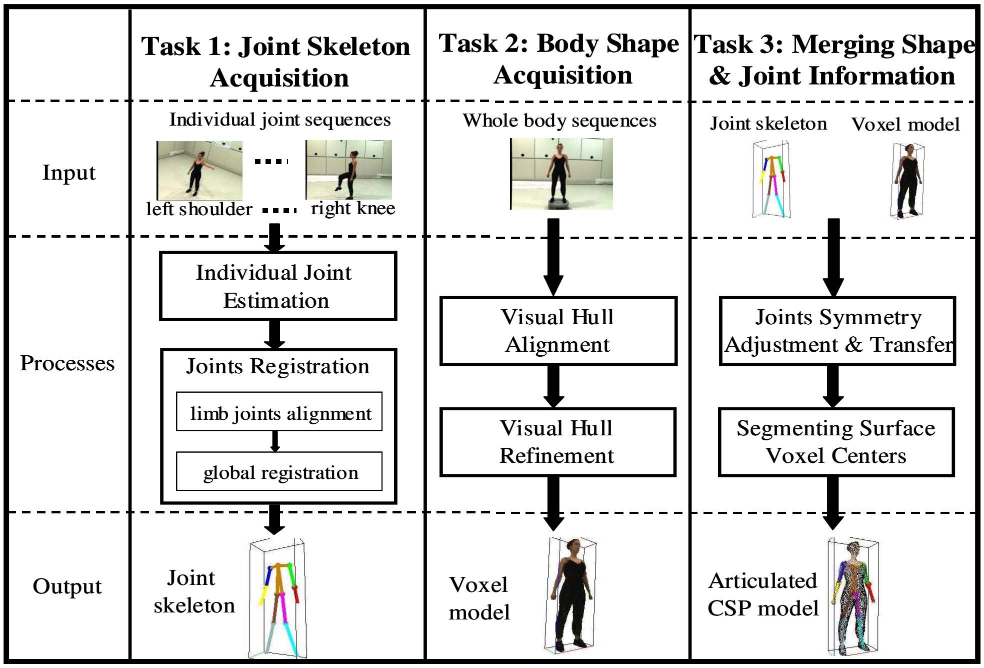


Figure 7: Flow chart illustrating the three tasks in our human kinematic modeling system.

ideal cutting plane for the arm would be the one which passes through the shoulder joint and the arm pit. To find this plane, a plane is swept circularly around the shoulder joint across the body as shown in Figure 6(a). The plane which cuts the least number of voxels is chosen to be the arm cutting plane. To separate the legs from the each other and from the body, three planes are used. The first plane passes through the right hip joint, the second plane passes through the left hip joint, each of the planes making a 45 degree angle with the horizontal. The third plane is a vertical plane which make a “Y” with the first two planes, as shown in Figure 6(b). With a slight abuse of terminology, hereafter we treat the surface voxel centers as if they are CSPs and call the merged model an articulated CSP model of the person. As a summary, Figure 7 illustrates the three component tasks in our vision-based human kinematic modeling system. Detailed implementations of each component of our system can be found in [CBK04].

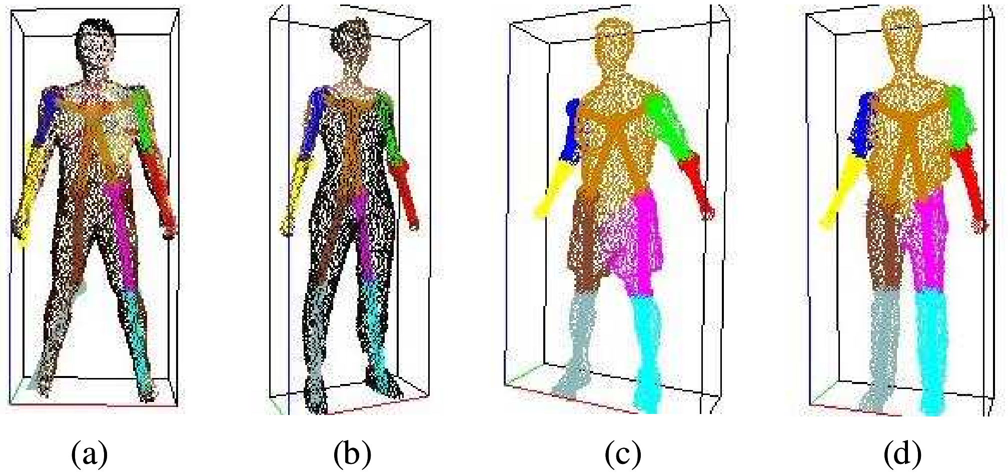


Figure 8: Articulated model of (a) synthetic virtual person, (b) SubjectE, (c) SubjectG and (d) SubjectS. In (a) and (b), the CSPs are shown with their original colors. In (c) and (d), the CSPs of different body parts are shown with different colors. For display clarity, the CSPs drawn are down-sampled in the ratio of one in two in total number of points.

2.5 Experimental Results

Articulated CSP models of a synthetic virtual person (see [CBK04]), SubjectE, SubjectG, and SubjectS are shown in Figures 8(a)(b)(c) and (d) respectively. The video clip **Subject-EGS-kinematicmodels.mpg** shows 3D fly-around views of the models of SubjectE, SubjectG and SubjectS. Note that the articulated CSP model can be turned into an articulated voxel model by substituting the center points by solid voxels (3D cubes). Table 1 shows the approximate timing for each step in our modeling system (see Figure 7 for a flow chart of the system). The data processing time is obtained from a 1.2GHz Pentium PC. It can be seen that our modeling is not real-time. The steps to recover the motion of the person on the turntable (Visual Hull Alignment) and processing the data from all eight joints are currently the bottlenecks of the system.

3 Human Articulated Tracking

In this section we show how the kinematic model of a person obtained using the system described in Section 2 can be used to track the motion of the person in new video sequences. The formulation of

Tasks	Time
Task 1: Joint Skeleton Acquisition	
Data Capture	10 seconds per joint
Data Processing: (a) Joint Estimation	$\approx$ 30 minutes per joint
(b) Joints Registration	$\approx$ 1 hour for 8 joints
Task 2: Body Shape Acquisition	
Data Capture	30 seconds
Data Processing: (a) Visual Hull Alignment	$\approx$ 2 hours
(b) Visual Hull Refinement	$\approx$ 5 minutes
Task 3: Merging Shape & Joint Information	
Data Processing: (a) Symmetry Adjustments & Transfer	$\approx$ 1s
(b) Segmenting Surface Voxel Centers	$\approx$ 2 minutes

Table 1: The approximate timing of each step of our modeling system. The data processing time is obtained on a 1.2GHz Pentium PC.

our motion tracking algorithm is similar to the 3D CSPs/2D image alignment principle used in both temporal SFS alignment algorithms proposed in Part I of this paper [CBK04]. The main addition is the incorporation of joint constraints into the motion equations as described in Section 3.2.

3.1 Related Work

Among all of the model based approaches to track human motion, the work by Sidenbladh et al. in [SDB00, SBF00], that by Delamarre and Faugeras in [DF99], that by Carranza et al. in [CTMS03] and that by Mikic et al. in [MTHC03] are the most related to our tracking algorithm.

Sidenbladh et al. [SBF00] perform human motion tracking by first modeling the person using articulated cylinders as body parts. Each body part is projected into a reference image to create an appearance model [SDB00]. Using a particle filtering framework [DBR00], the articulated 3D appearance model is then used to track the motion [SBF00]. As pointed out by the authors themselves, their model works well for tracking a single body part but is too weak for constraining the motion of the entire body without using specific motion models. Hence their approach is restricted to tracking simple motions such as walking or running for which motion model can be created by collecting examples [SBF00].

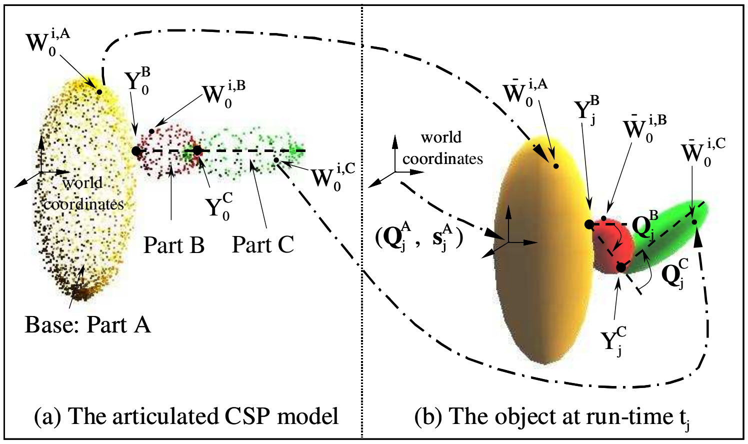
In [DF99], silhouette contours from multiple cameras are used to constraint the articulated model (which consists of geometric primitives such as cylinders or truncated cones) of a person. The way of generating "forces" to align 2D contours of the projected model with the silhouette boundary is similar to the geometric constraints we use in our tracking algorithm. In [CTMS03], Carranza et al. first render a human model using graphics hardware and then compare the rendered images (using pixel-wise XOR) with the silhouette images extracted from video sequences to track human motion. Although it is unclear exactly how their XOR errors are formulated as driving forces for optimizing the motion parameters, their grid-search initialization procedure provides a good way to reduce the problem of local minima. Mikic et al. also use multiple-view silhouettes in [MTHC03] for motion tracking, although their body part fitting is done in 3D space and is closely related to our previous work in [CKBH00]. None of the above work uses color information, unlike in our algorithm.

3.2 Image-Based Articulated Object Tracking

We consider the problem of tracking an articulated object in (color and silhouette) video sequences using a known articulated model of the object. We assume the articulated model is constructed using the human kinematic modeling system described in Section 2. The model consists of rigid parts with known shape described in terms of CSPs. The rigid parts are connected to each other at known joint locations.

3.2.1 Problem Scenario

Figure 9(a) depicts an articulated CSP model of an object consisting of three rigid parts A, B and C with part A being the base of the object. Without loss of generality, we assume that the model is at its reference configuration which means the rotation angles of the joints and the translation of the base part A are all zero. Hereafter we represent the 3D position and color of the i^{th} CSP of part Z at time t by $W_t^{i,Z}$ and $\mu^{i,Z}$ respectively, where $t = 0$ denotes the model frame. Now assume



(a) The articulated CSP model

(b) The object at run-time t_j

Figure 9: (a) The articulated CSP model of an articulated object with three rigid parts A, B and C. (b) The object itself at run-time t_j. The articulated CSP model in (a) is used to estimate the motion parameters of the object at t_j.

the shape information of the model is given as sets of CSPs represented by $\{W_0^{i,A}, \mu_0^{i,A}; \quad i = 1, \cdots\cdots, L_0^A\}$, $\{W_0^{i,B}, \mu_0^{i,B}; \quad i = 1, \cdots\cdots, L_0^B\}$, $\{W_0^{i,C}, \mu_0^{i,C}; \quad i = 1, \cdots\cdots, L_0^C\}$ for the parts A, B and C respectively and the joint locations of the model are known and denoted by Y_0^B and Y_0^C. Furthermore, we assume the model color and silhouette images $\{I_0^k, S_0^k; \quad k = 1, \cdots\cdots, K\}$ that were used to construct the model are available.

Suppose we have imaged the articulated object by K cameras at each of J time instants with the color and silhouette images represented by $\{I_j^k, S_j^k; \quad k = 1, \cdots\cdots, K; \quad j = 1, \cdots\cdots, J\}$. Also assume that we have extracted from these images J sets of (unsegmented) CSPs $\{W_j^i, \mu_j^i\}$ (see Section 4.2.1 in [CBK04]). If we represent the positions and orientations (with respect to the reference configuration at the model frame) of the base part A at time t_j be $(\boldsymbol{Q}_j^A, \boldsymbol{s}_j^A)$ and the rotations of parts B and C about their joints as $\boldsymbol{Q}_j^B, \boldsymbol{Q}_j^C$ respectively, the goal of image-based articulated object tracking can then be stated as:

The Image-Based Articulated Object Tracking

Given the above input information, estimate $(\boldsymbol{Q}_j^A, \boldsymbol{s}_j^A)$ of the base part A and $\boldsymbol{Q}_j^B, \boldsymbol{Q}_j^C$ of the articulated joints at time t_j for all $j = 1, \cdots\cdots, J$.

3.2.2 Tracking Principle

We explain the tracking principle using the j^{th} frame data captured at run-time t_j (see Figure 9(b)). We assume the articulated object has already been tracked at t_{j-1}, i.e. we have initial estimates of the parameters $\boldsymbol{Q}^A_{j-1}$, $\boldsymbol{s}^A_{j-1}$, $\boldsymbol{Q}^B_{j-1}$ and $\boldsymbol{Q}^C_{j-1}$. As a recap, we have the following information as the input data:

1. Model data: (1a) segmented model CSPs $\{W_0^{i,A}, \mu_0^{i,A}, W_0^{i,B}, \mu_0^{i,B}, W_0^{i,C}, \mu_0^{i,C}\}$,
   (1b) known model joint positions Y_0^B and Y_0^C,
   (1c) model color and silhouette images $\{I_0^k, S_0^k\}$ used to construct the model.
2. Data at t_j: (2a) run-time unsegmented CSPs $\{W_j^i, \mu_j^i\}$,
   (2b) run-time color and silhouette images $\{I_j^k, S_j^k\}$,
   (2c) estimated parameters $\boldsymbol{Q}^A_{j-1}$, $\boldsymbol{s}^A_{j-1}$, $\boldsymbol{Q}^B_{j-1}$ and $\boldsymbol{Q}^C_{j-1}$ from previous frame.

Just as when aligning two Visual Hulls [CBK04], we pose the problem of estimating $\boldsymbol{Q}^A_j$, $\boldsymbol{s}^A_j$, $\boldsymbol{Q}^B_j$ and $\boldsymbol{Q}^C_j$ as the problem of minimizing the geometric and color errors caused by projecting the 3D CSPs into the 2D images. To be more specific, there are two types of errors we can use:

1. The *forward* geometric and photometric errors of projecting (respectively) the segmented model CSPs $\{W_0^{i,Z}, \mu_0^{i,Z}\}$ into the run-time silhouette $\{S_j^k\}$ and color images $\{I_j^k\}$.
2. The *backward* geometric and photometric error of projecting (respectively) the run-time CSPs $\{W_j^i, \mu_j^i\}$ into the model silhouette $\{S_0^k\}$ and color images $\{I_0^k\}$.

Given estimates of $\boldsymbol{Q}^A_j$, $\boldsymbol{s}^A_j$, $\boldsymbol{Q}^B_j$ and $\boldsymbol{Q}^C_j$, the forward errors are obtained by applying the motions to the already segmented model CSPs and projecting them into the run-time images. To calculate the backward errors, however, an extra step is required. In order to apply the correct motion transformations (for part A, B or C) to the run-time CSPs, we have to decide for each run-time CSP W_j^i, which part of the articulated object it belongs to. In other words, we have to segment the set of CSPs $\{W_j^i, \mu_j^i\}$ into parts A, B and C. Segmenting a set of 3D points is a difficult problem, and a variety of approaches have been used under different situations. Two approaches for segmenting the run-time CSPs based on the known shape information of the model and the estimated motion parameters from the previous frame are discussed in [Che03]. Once the run-time CSPs have been segmented, the backward errors can be calculated and added to the forward errors.

Theoretically it is sufficient to just include the forward errors in the optimization equations. However, the advantage of including the backward errors is that the motion parameters are then more highly constrained. With the addition of the backward errors, tracking is less likely to fall into local minimum, especially when two parts of the articulated object are very close to each other (see Section 3.3.3 for more details). The disadvantage of including the backward errors is the extra step that is required to segment the run-time CSPs. The backward errors should not be used if the segmentation of the run-time CSPs is not reliable.

3.2.3 Incorporating Joint Constraints into the Optimization Equations

In this section we describe how to incorporate joint constraints into the calculation of the forward and backward errors. For the forward errors, let $\bar{W}_0^{i,A}$, $\bar{W}_0^{i,B}$ and $\bar{W}_0^{i,C}$ be the positions of $W_0^{i,A}$, $W_0^{i,B}$ and $W_0^{i,C}$ at run-time t_j (see Figure 9(b)). Using the joint constraints between the parts, we have the following equations relating the transformed model CSPs and the joint positions (Y_j^B and Y_j^C) at t_j with the motion parameters:

$$Part\ \ A: \qquad \bar{W}_0^{i,A} = \boldsymbol{Q}_j^A W_0^{i,A} + \boldsymbol{s}_j^A\ , \tag{1}$$

$$Part\ \ B: \qquad Y_j^B = \boldsymbol{Q}_j^A Y_0^B + \boldsymbol{s}_j^A\ , \qquad \bar{W}_0^{i,B} = \boldsymbol{Q}_j^A \boldsymbol{Q}_j^B (W_0^{i,B} - Y_0^B) + Y_j^B\ , \tag{2}$$

$$Part\ \ C: \quad Y_j^C = \boldsymbol{Q}_j^A \boldsymbol{Q}_j^B (Y_0^C - Y_0^B) + Y_j^B\ , \qquad \bar{W}_0^{i,C} = \boldsymbol{Q}_j^A \boldsymbol{Q}_j^B \boldsymbol{Q}_j^C (W_0^{i,C} - Y_0^C) + Y_2^C\ . \tag{3}$$

Using the above equations, the forward errors are written as

$$e_{2,1} = \sum_{Z=A,B,C} \left[\sum_{i=1}^{L_0^Z} \sum_{k} \left\{ \tau * d_j^k(\bar{W}_0^{i,Z}) + [\boldsymbol{c}_j^k(\bar{W}_0^{i,Z}) - \mu_0^{i,Z}]^2 \right\} \right]\ , \tag{4}$$

where $d_j^k(\bar{W}_0^{i,Z})$ represents the distance between the 2D projection of $\bar{W}_0^{i,Z}$ and the silhouette image S_j^k, and $\boldsymbol{c}_j^k(\bar{W}_0^{i,Z})$ denotes the projected color of $\bar{W}_0^{i,Z}$ on the color image I_j^k with τ being a weighing constant (see Section 4.2.3 of [CBK04]). Note that the error of a model CSP with respect to the k^{th} run-time color and silhouette image is calculated only if the CSP is visible in that camera. Since in this case, the object consists of articulated rigid parts, the "reverse approach" described in [CBK04] for testing visibility is not applicable. An alternative method for determining visibility for articulated object tracking is presented in [Che03].

To calculate the backward errors $e_{1,2}$, we first express the positions of the (now assumed segmented) run-time CSPs with respect to the model images in terms of the motion parameters $\boldsymbol{Q}_j^A, \boldsymbol{s}_j^A, \boldsymbol{Q}_j^B$ and $\boldsymbol{Q}_j^C$ by inverse transforming the set of motion relations in Equations (1) to (3). Then the transformed run-time CSPs are projected into the model silhouette and color images to get the geometric and photometric errors. Combining the backward and forward error terms (Equation (4)), the optimization equation becomes:

$$\arg \min_{\boldsymbol{s}_j^A, \boldsymbol{Q}_j^A, \boldsymbol{Q}_j^B, \boldsymbol{Q}_j^C} \left[e_{2,1} + e_{1,2} \right] \ , \qquad (5)$$

which can be solved using the Levenberg-Marquardt (LM) algorithm described in [Che03].

Although we have described the tracking algorithm using an example articulated object consists of three parts, it can be easily extended to articulated objects with N parts. In the special case where the motion (rotation and translation) of the base (part A in our example) is known, or if it is static, the problem degenerates to tracking a multi-link object around a fixed point. An example would be the situation we discussed in Section 2.2.2 for globally registering the joints of the limbs. Note that in such cases our algorithm still applies, the only difference being that $(\boldsymbol{Q}_j^A, \boldsymbol{s}_j^A)$ are known constants instead of parameters to be optimized in Equation (5). The Image-Based Articulated Object Tracking Algorithm is summarized below:

Image-Based Articulated Object Tracking Algorithm

1. Initialize the motion parameters in the first frame t_1.

2. For $j = 1, \cdots\cdots, J$, estimate the motion parameters at t_j by the following procedures:

   (a) Initialize the motion parameters at t_j with those estimated at t_{j-1}.

   (b) Segment the run-time CSPs at t_j.

   (c) Apply the Iterative LM algorithm [Che03] to Equation (5) to minimize the sum of forward errors and backward errors with respect to the motion parameters $\boldsymbol{Q}_j^A$, $\boldsymbol{s}_j^A$, $\boldsymbol{Q}_j^B$ and $\boldsymbol{Q}_j^C$ until convergence is attained or for a fixed maximum number of iterations.

3.3 Tracking Full Body Human Motion

3.3.1 The Articulated Human Model

The articulated CSP models used to track human motion are the same as those built in Section 2 (see for example Figure 8). Each model consists of nine body parts: torso, right/left lower/upper arms and legs, connected by eight joints: right/left shoulder, elbow, hip and knee joints. Each body part is assumed to be rigid with the torso being the base. The shoulder and hip joints have 3 degrees-of-freedom (DOF) each while there is 1 DOF for each of the elbow and knee joints. Including translation and rotation of the torso base, there are a total of 22 DOF in the model.

3.3.2 Hierarchical Tracking

The most straightforward way to use the Image-Based Articulated Object Tracking Algorithm for human motion tracking is to apply it directly to all the body parts (with a total of 22 DOF) at the same time. In practice, however, this all-at-once approach is prone to local minima because of the high dimensionality. To reduce the chance of falling into local minimum, we instead use a two-step hierarchical approach: first fit the torso base and then fit each limb independently. This approach makes use of the fact that the motion of the body is largely independent of the motion of the limbs which are largely independent of each other. The first step of our hierarchical approach involves recovering the global translation and orientation of the torso base. This can be done using the 6 DOF temporal SFS algorithm for rigid objects (see [CBK04]). Once the global motion of the torso

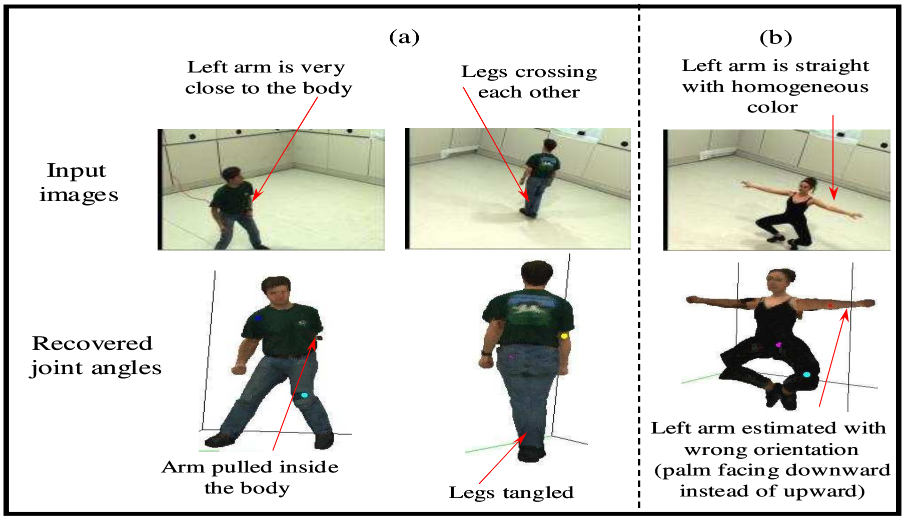

(a)
(b)
Left arm is very close to the body
Legs crossing each other
Left arm is straight with homogeneous color
Input images
Recovered joint angles
Arm pulled inside the body
Legs tangled
Left arm estimated with wrong orientation (palm facing downward instead of upward)

Figure 10: Two situations where our tracking algorithm is particularly vulnerable to local minima. (a) The body parts are very close to each other. (c) The arm is straight and of homogeneous color in which there is ambiguity around the shoulder joint.

has been estimated, the four joint positions: left/right shoulders and hips are calculated. In the second step, the four limbs are aligned separately around these fixed joint positions just as in the special case mentioned at the end of Section 3.2.3. Using such a hierarchical approach not only reduces the chance of falling into local minimum, but also reduces the processing time as there are only four unknowns to be optimized for each limb.

3.3.3 Dealing with Local Minimum

As common to all methods which use an error minimization formulation, our human motion tracking algorithm is prone to the problem of local minima, especially since the human articulated body has a very large number of DOF. Though we have used the hierarchical approach (discussed in Section 3.3.2) to reduce the dimensionality, the problem of local minima cannot be completely avoided.

Empirically we found that there are two situations when our tracking algorithm is particularly vulnerable to local minima. The first situation occurs when the body parts are very close to each other. In this situation, there is a good chance that the optimization gets trapped in a local minimum and the body parts penetrate each other such as the examples shown in Figure 10(a).

The second situation happens when the arm is straight and there is not enough color (or texture) information on the arm to differentiate the rotation angle of the shoulder joint about the axis along the length of the arm. An example is illustrated in Figure 10(c) where the palm of the left arm of SubjectE is facing upward but the recovered joint angles have the palm of the arm facing downward (i.e. the joint angles of the left shoulder joint is rotated around the axis along the arm by 180 degrees). Note that the local minima in the first situation is only a valid solution in the solution space but not a valid solution in the physical world while the local minima in the second situation is valid in both the solution space and the physical world, although it is not the correct solution.

To cope with the first situation, collision detection and reinitialization is added to our algorithm. In each frame, after all the joint angles have been estimated, the body parts are checked for collision. If a collision is detected between a limb and the body, within each limb (e.g. collision of upper and lower arm) or between limbs, the joint angles of the limbs involved in the collision are reinitialized and re-aligned. To reinitialize, instead of using only the joint angles estimated from the previous one frame, those from the previous three frames are used to predict the initial guess. To increase the chance of climbing out of the local minimum, a small random perturbation is also added to the initial guess. Although this heuristic is sufficient to avoid some of the local minima, it still fails occasionally. For a failed frame, to avoid propagating the wrong estimates to the next frame, the joint angles are set to be those estimated from the previous frame, hoping that the local minimum problem will be resolved in the next frame. For cases where a limb is totally lost in the tracking, manual reinitialization is required to restart the tracking of that limb.

The second situation is difficult to deal with because the geometric constraints are unable to resolve the ambiguity due to the symmetry of the arm. In cases when there is no texture on the arm (as in the case of SubjectE), the photometric constraints are also unable to correct the misalignment. Although currently we have not implemented a satisfactory solution to this situation, the tracking generally recovers by itself once the arm is bent (when the ambiguity can be resolved by the geometric constraints).

Another possible way to reduce the problem of local minima in both situations is to impose angle and velocity limits on each joint during tracking, similar to the search grid idea used by Carranza et. al. in [CTMS03]. Although not implemented in our current system, we are planning to incorporate the joint/velocity limit into our system in the near future (see Section 4.2 for more details).

3.4 Experimental Results

To test our tracking algorithm, two types of data are used: (1) synthetic sequences with ground-truth are generated using OpenGL to obtain a quantitative evaluation and (2) sequences of real people with different motions are captured to obtain qualitative results. On the average, the tracking takes about 1.5 minutes per frame on a 1.2GHz Pentium PC.

3.4.1 Synthetic Sequences

Two synthetic motion video sequences: KICK (60 frames) and PUNCH (72 frames) were generated using the synthetic human model used in Part I [CBK04] with a total of eight cameras per sequence. The articulated model shown in Figure 8(a) is used to track the motion in these sequences. Figure 11 compares the ground-truth and estimated joint angles of the left arm and right leg of the body in the KICK sequence. It can be seen that our tracking algorithm performs very well. The movie **Synthetic-track.mpg** illustrates the tracking results on both sequences. In the movie, the upper left corner shows one of the input camera sequences, the upper right corner shows the tracked body parts and joint skeleton (rendered in color) overlaid on the (gray-scale version of the) input images. The lower left corner depicts the ground-truth motion rendered using an avatar and the lower right corner represents the tracked motions with the same avatar. The avatar renderings show that the ground-truth and tracked motions are almost indistinguishable from each order.

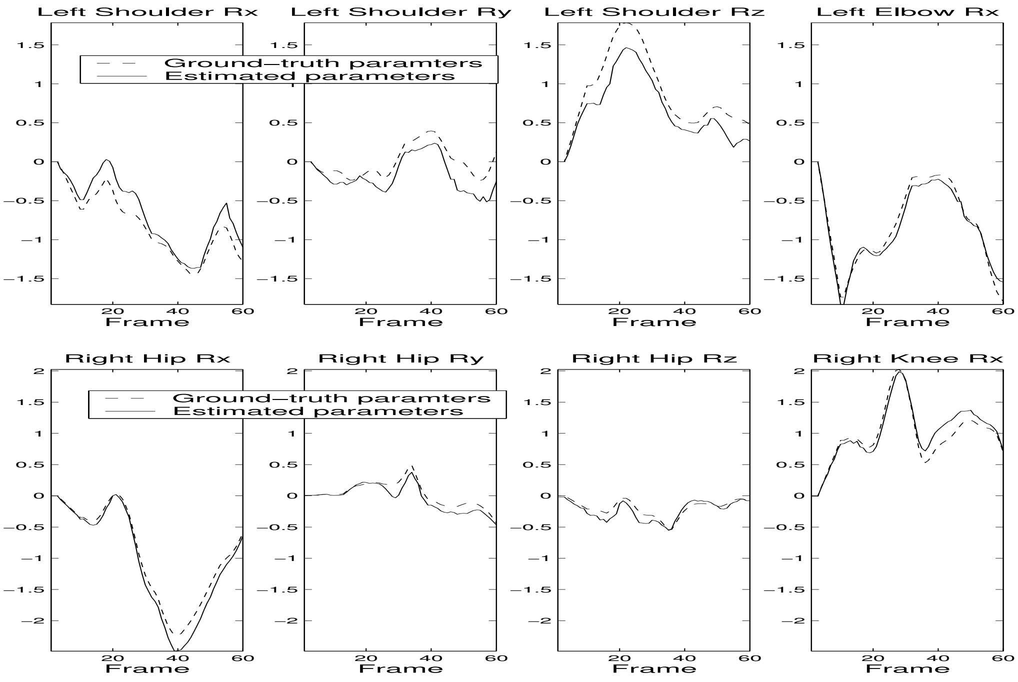


Figure 11: Graphs comparing ground-truth and estimated joint angles of the left arm and right leg of the synthetic sequence KICK. The estimated joint angles closely follow the ground-truth values throughout the whole sequence. The tracking results of the KICK sequence can be seen in the movie **Synthetic-track.mpg**.

3.4.2 Real Sequences

We also tested our tracking algorithm on a variety of sequences of real human subjects performing a wide range of motions. For SubjectG, three video sequences: STILLMARCH (158 frames) , AEROBICS (110 frames) and KUNGFU (200 frames) were captured to test the tracking algorithm with eight cameras used in each sequence. Figures 12 and 13 show the tracking results on the AEROBICS and KUNGFU sequences respectively. Each figure shows selected frames of the sequence with the (color) tracked body parts and the joint skeleton overlaid on one of the eight camera input images (which are converted to gray-scale for display). The movie **SubjectG-track.mpg** contains results on all three sequences. In the movie, the upper left corner represents one of the input camera images and the upper right corner illustrates the tracked body parts with joint skeleton overlaid on a gray-scale version of the input images. The lower left corner illustrates the results of applying the estimated motion data to a 3D articulated voxel model (obtained from the articulated CSP model as discussed at the end of Section 2.4) of the person while the lower right corner shows the results

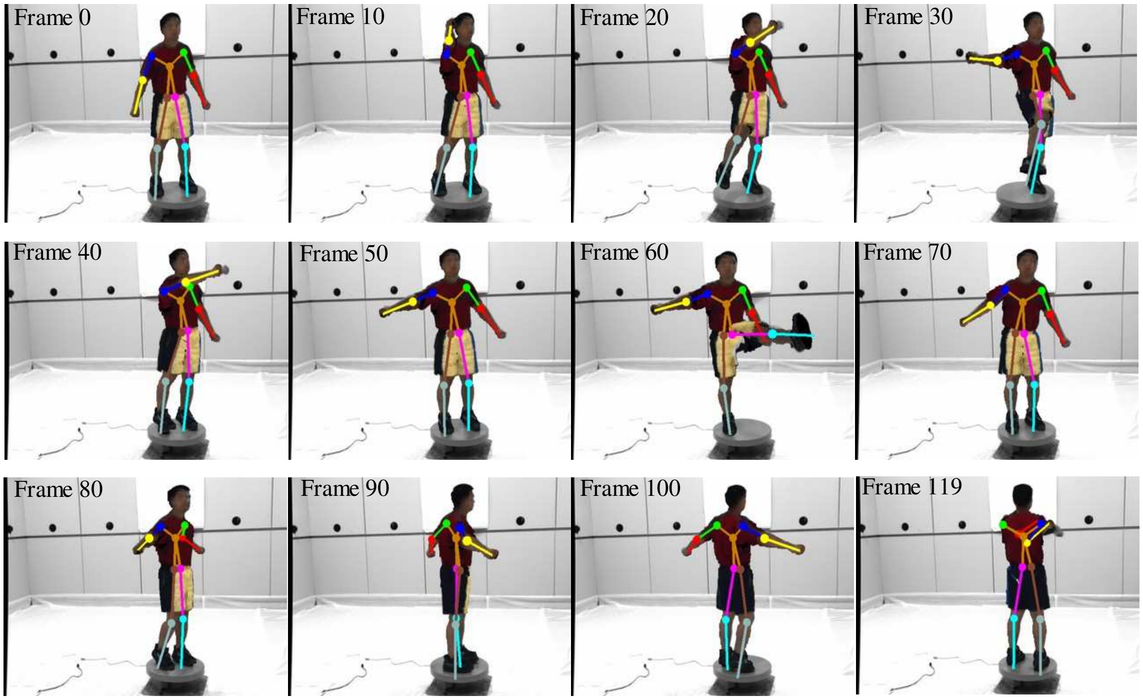


Figure 12: Twelve selected frames of the tracking results for the AEROBICS sequence. The tracked body parts and joint skeleton (rendered color) are overlaid on one of the input camera images (which are converted from color to gray-scale for clarity). The whole sequence can be seen in the movie **SubjectG-track.mpg**.

of applying the estimated motion data to an avatar. The video demonstrates that our tracking algorithm tracks well on both simple motions (STILLMARCH, AEROBICS) and complicated motions (KUNGFU). Note that in the above three sequences, the remedy discussed in Section 3.3.3 is not used for dealing with the problem of local minimum. Since the motions in the STILLMARCH and AEROBICS are simple, no local minimum problems are encountered in these two sequences. However, for the KUNGFU sequence, the tracking of the right arm is lost in frame 91 for 10 frames due to local minimum but recovers automatically at frame 101.

A motion sequence THROW (155 frames) of SubjectS is captured. The sequence is first tracked by our algorithm without using the local minimum remedy. Since body parts are not checked for collision, when the left arm is very close to the body at frame 70, local minimum pulls the left arm inside the body (see Figure 10(a)). Moreover, the tracking of both legs is also lost around frame 43 (which is shown in Figure 10(b)) when the legs start to cross each other. To resolve these problems, the sequence is re-tracked with the local minimum remedy turned on. The results are shown in Figures 14 which shows 24 selected frames of the sequence with the (color) tracked body parts and the joint skeleton overlaid on one of the eight camera input images (which are converted to gray-scale for display). The local minima problems of the legs and the left arm are successfully resolved by checking for body part collision and reinitialization. The whole THROW sequence can be seen in the movie **SubjectS-track.mpg**.

Two sequences: STEP-FLEX (90 frames) and SLOWDANCE (270 frames) of SubjectE were also captured and tracked. Some of the tracked frames are shown in Figure 15 for the SLOW-DANCE sequence and Figure 16 for the STEP-FLEX sequence (the tracking results of both sequences are included in the movie clip **SubjectE-track.mpg**). The shoulder joint ambiguity problem (Figure 10(c)) happens in the SLOWDANCE sequence on the left arm around frame 28 and on the right arm around frame 85 though the tracking recovers in later frames of the sequence. Because we do not include the waist joint in our kinematic model, generally motions involve the bending of the body around the waist cannot be tracked accurately. However for the bending motion in the STEP-FLEX sequence, the geometric constraints from the silhouette drove our tracking algorithm to approximate the bending of the body using the hip joints which are the only degrees of freedom that can explain the silhouette images. Note that the tracked motion will be more accurate and natural if the waist joint is modeled.

Note that in the tracking results of the STEP-FLEX sequence, there are frames in which a tracked foot slips/slides or floats in the air when it has to be kept touching the ground, causing very unnatural looking motions. In cases where the type of motion is known to have contacts between the body parts and the surrounding environment (such as contact between the feet and the ground), these contact constraints can be incorporated into the optimization formulation to increase the tracking accuracy. However because our system is designed to capture general motion, currently we do not impose any contact constraints during tracking.

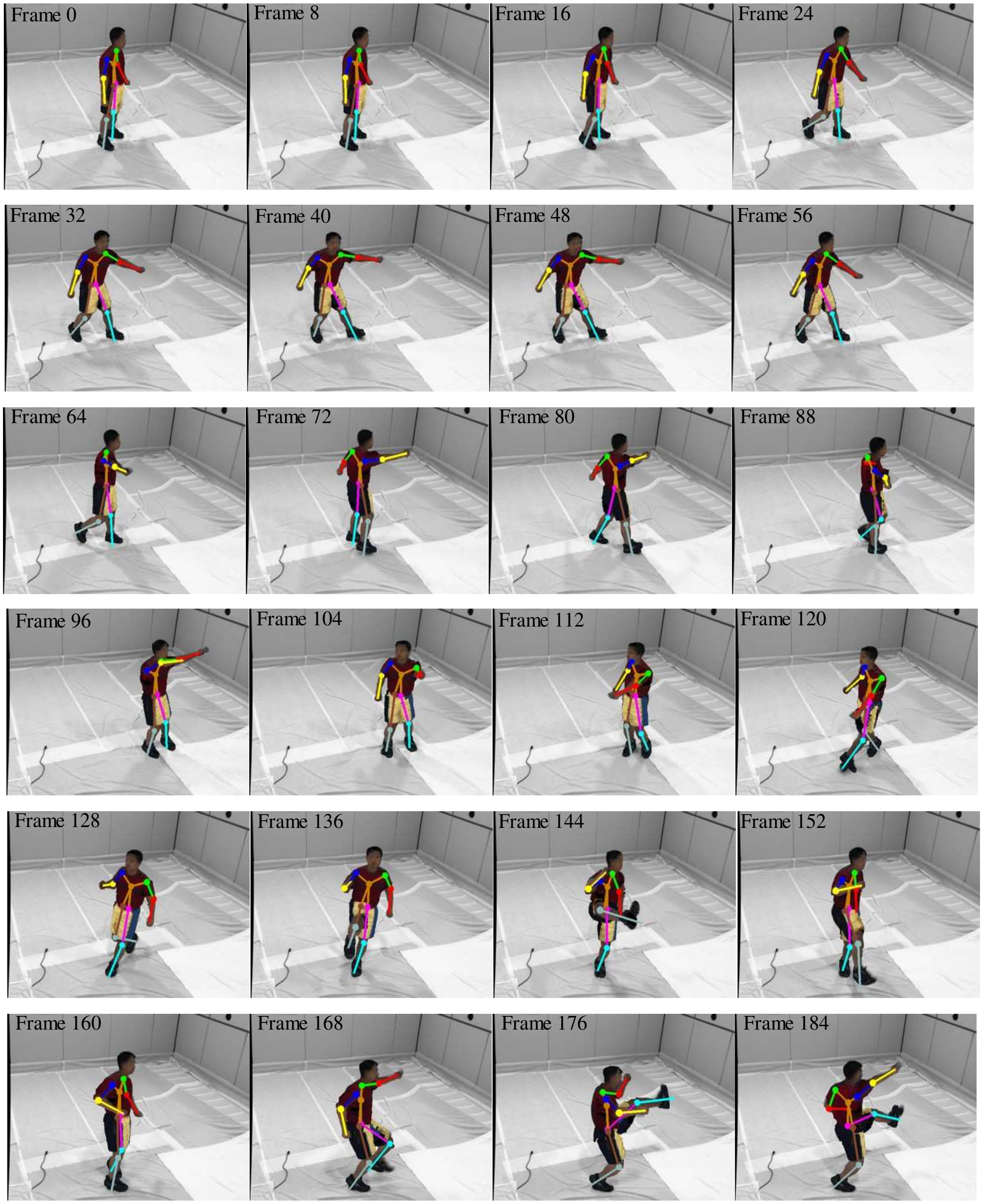


Figure 13: Twenty-four selected frames of the tracking results for the KUNGFU sequence. The whole sequence can be seen in the movie **SubjectG-track.mpg**.

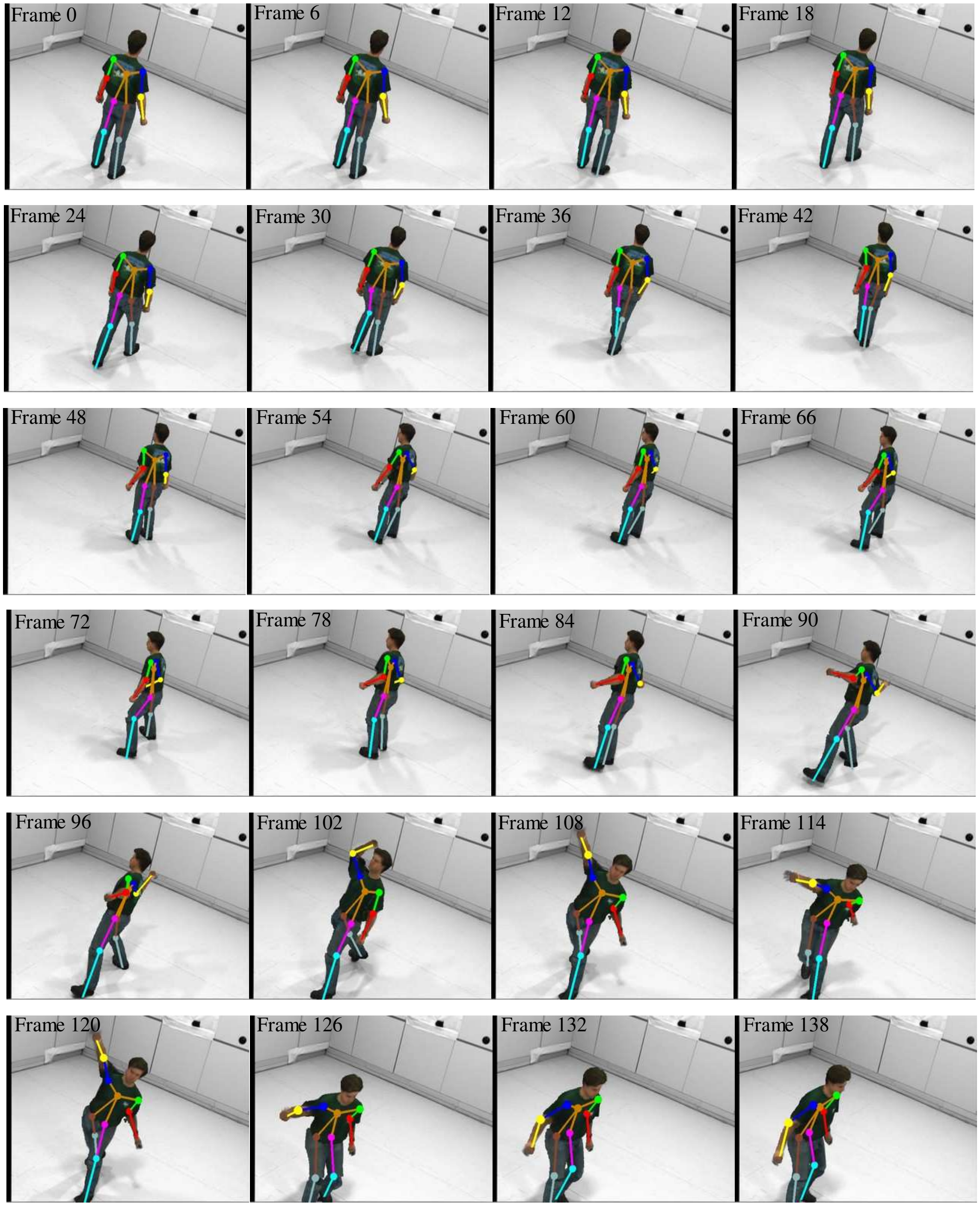


Figure 14: Twenty-four selected frames of the tracking results for the THROW sequence. The whole sequence can be seen in the movie **SubjectS-track.mpg**.

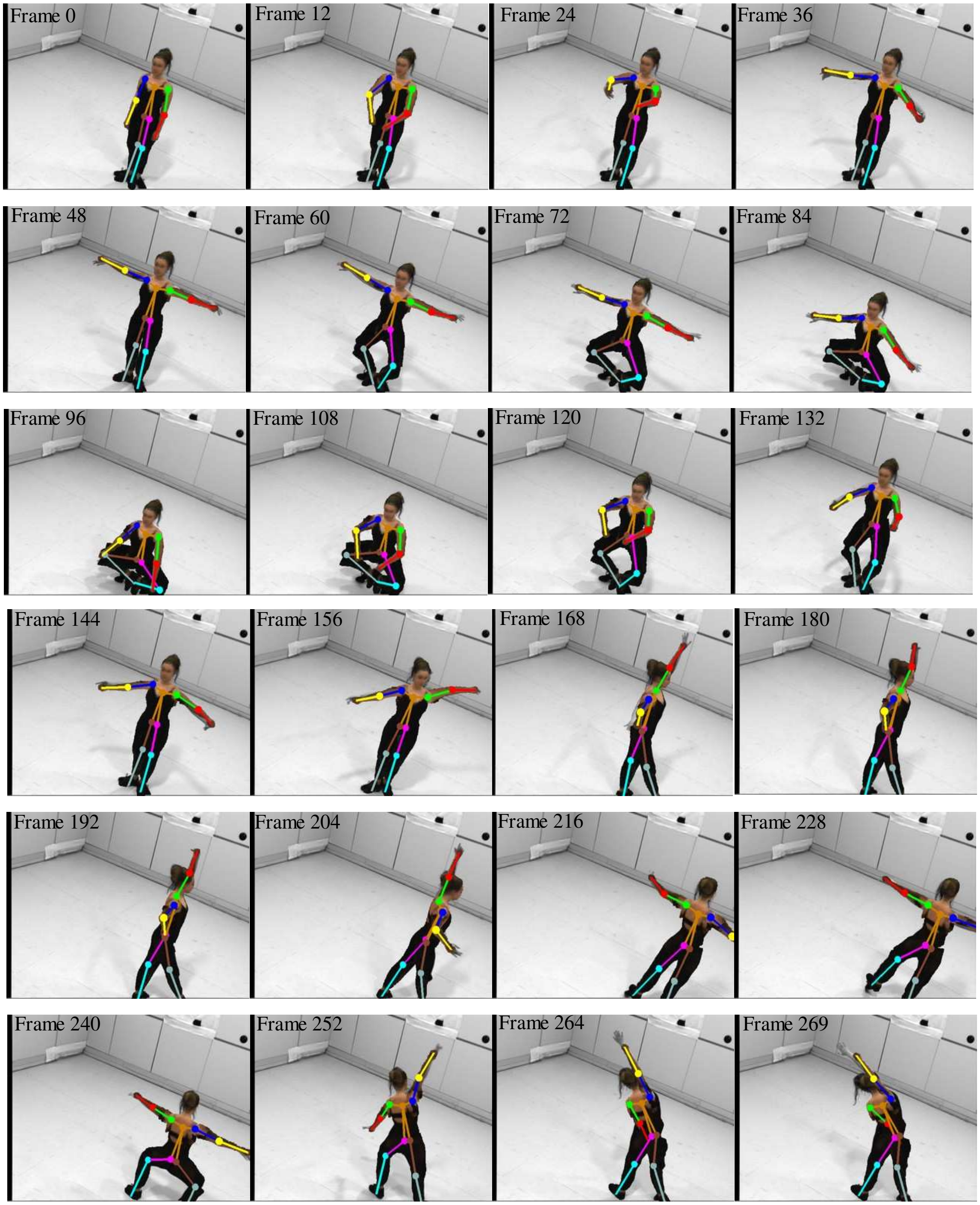


Figure 15: Twenty-four selected frames of the tracking results for the SLOWDANCE sequence. The whole sequence can be seen in the movie **SubjectE-track.mpg**.

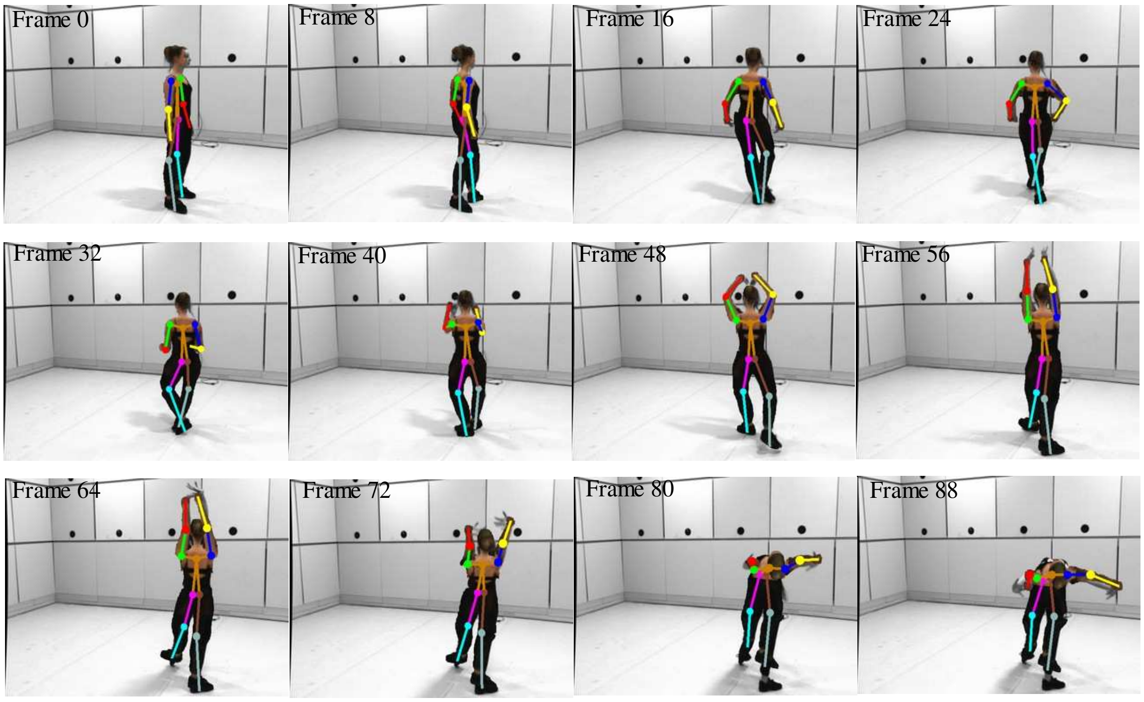


Figure 16: Twelve selected frames of the tracking results for the STEP-FLEX sequence. The whole sequence can be seen in the movie **SubjectE-track.mpg**.

4 Conclusion

4.1 Summary

Compared to other human modeling approaches which fit generic human models composed of simple shape primitive to the input image data [LY95, KM98, PFD99, CKBH00], our vision-based kinematic modeling system constructs a body model from scratch using simple joint connection knowledge of the body without using any a priori shape model. We acquire and register the skeletal structure using video sequences of the person moving their limbs and extract shape information (in terms of CSPs) of the body parts directly from the silhouette and color images. The joint and shape information is then merged to form a complete kinematic model consisting of voxels segmented into body parts and joint locations. Compared to laser scanning technology which usually only captures shape information, our system is cheaper, non-invasive and more importantly, provides the joint locations. However, since our system uses the motion of the body parts to recover the joint locations, it does not perform as well with joints which have a restricted range of movement, such as the neck, wrist and ankle joints.

Due to the high number of degrees of freedom of the human body, motion tracking is a difficult problem. The problem is particularly challenging for vision-based (marker-less) approaches because of self occlusion, unknown kinematic information, perspective distortion and cluttered environments. In this paper, we have shown how to use Visual Hull for human motion tracking. Our tracking algorithm has two major advantages compared to other model-based methods. First, our person specific models closely approximate the actual shape of the body parts, with joint information estimated directly from the motion of the person. The accurate kinematic model gives better shape and joint constraints than methods which uses simple approximating geometric primitives. Secondly the (color) appearance model provided by the CSPs combines the geometric constraints and color consistency in one optimization formulation. Most other vision-based motion tracking methods lack the ability to use both color and shape information simultaneously.

For relatively simple motions, such as the STILLMARCH and AEROBICS sequences, our tracking algorithm works very well. However, for complex motions such as those in the KUNGFU and THROW sequences, our algorithm suffers from the problem of local minima. This problem is unavoidable because of the error minimization formulation of the algorithm. Although the remedy we suggested in Section 3.3.3 is able to resolve some of these local minima problems, there are unresolvable situations such as the one in Figure 10(c). Another way to alleviate the local minima problem is to apply joint angles limits (or reachable workspace constraints as defined in [MLS94]) to the tracking error measure. See Section 4.2 for more details of how this might be done.

4.2 Future Work

Our work in this paper can be considered as a step toward building a completely vision-based and totally autonomous 3D human modeling and motion capture system, however, there are still several difficulties to overcome before such systems are widely used in industry. We briefly discuss three possibilities for future work to further improve our systems.

Because we model each separate body part as rigid, our system is not able to capture subtle surface deformation caused by muscles and clothing. The ability to capture such deformation is essential for realistic animation of the acquired model and captured motion [CBHK04]. One possible future direction is to incorporate deformable models into our system to capture non-rigid movements of the skin, muscle and clothing.

Although our tracking algorithm works well, it suffers from the problem of local minima, a problem common to all methods that use an error optimization formulation. In Section 3 we suggested including joint angles and velocity limits to reduce the problem of local minimum. Prior to tracking, the allowable range of motions (of all the joint angles) and angular velocity of the person are estimated. The space of all joint parameters is then divided into valid and invalid workspaces. This a priori workspace information can then be incorporated into the tracking optimization equations by adding very high errors to the error criterion when the body joint angles are in the invalid workspace or the angular velocities are out of the pre-estimated ranges, while no extra error is added when the joint angles are in the valid zones.

Last but not the least, the current version of our modeling and tracking systems are not realtime Being able to model people and track their motions is critical in applications such as human computer interface, security and surveillance. Another possible area of future work is to explore the possibility of applying efficient image alignment algorithms such as [BM04] to reduce the processing time for both modeling and tracking.

References

[ACP03] B. Allen, B. Curless, and Z. Popovic. The space of human body shapes: Reconstruction and parameterization from range scans. In *Computer Graphics Annual Conference Series (SIGGRAPH'03)*, pages 587–594, San Diego, CA, July 2003.

[BK99] D. Beymer and K. Konolige. Real-time tracking of multiple people using stereo. In *Proceedings of International Conference on Computer Vision (ICCV'99)*, Corfu, Greece, September 1999.

[BK00] C. Barron and I. Kakadiaris. Estimating anthropometry and pose from a single image. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'00)*, Hilton Head Island SC, June 2000.

[Bli82] J. Blinn. A generalization of algebraic surface drawing. *ACM Transactions on Graphics*, 1(3):235–256, 1982.

[BM97] C. Bregler and J. Malik. Video motion capture. Technical Report CSD-97-973, University of California Berkeley, 1997.

[BM98] C. Bregler and J. Malik. Tracking people with twists and exponential map. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'98)*, volume 1, pages 8–15, Santa Barbara, CA, June 1998.

[BM04] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221 – 255, March 2004.

[CA96] Q. Cai and J. Aggarwal. Tracking human motion using multiple cameras. In *Proceedings of International Conference on Pattern Recognition (ICPR'96)*, volume 3, pages 68–72, August 1996.

[CA98] Q. Cai and J. Aggarwal. Automatic tracking of human motion in indoor scenes across multiple synchronized video streams. In *Proceedings of the Sixth International Conference on Computer Vision (ICCV'98)*, Bombay, India, January 1998.

[CBHK04] K. Cheung, S. Baker, J. Hodgins, and T. Kanade. Markerless human motion transfer. In *Proceedings of the Second International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT'04)*, Thessaloniki, Greece, September 2004.

[CBK03a] G. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette for articulated objects and its use for human body kinematics estimation and motion capture. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'03)*, Madison, MI, June 2003.

[CBK03b] G. Cheung, S. Baker, and T. Kanade. Visual hull alignment and refinement across time:a 3D reconstruction algorithm combining shape-frame-silhouette with stereo. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'03)*, Madison, MI, June 2003.

[CBK04] K. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette across time: Theory and algorithms. *Accepted by International Journal on Computer Vision*, 2004.

[Che03] G. Cheung. *Visual Hull Construction, Alignment and Refinement for Human Kinematic Modeling, Motion Tracking and Rendering*. PhD thesis, Carnegie Mellon University, 2003.

[CKBH00] G. Cheung, T. Kanade, J. Bouquet, and M. Holler. A real time system for robust 3D voxel reconstruction of human motions. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'00)*, Hilton Head Island, SC, June 2000.

[Coe98] M. Coen. Design principals for intelligent environments. In *Proceedings of AAAI Spring Symposium on Intelligent Environments*, Stanford, CA, 1998.

[CR99a] T. Cham and J. Rehg. A multiple hypothesis approach to figure tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'99)*, Ft. Collins, CO, June 1999.

[CR99b] T. Cham and J. Rehg. Dynamic feature ordering for efficient registration. In *Proceedings of International Conference on Computer Vision (ICCV'99)*, Corfu, Greece, September 1999.

[CTMS03] J. Carranza, C. Theobalt, M. Magnor, and H. Seidel. Free-viewpoint video of human actors. In *Computer Graphics Annual Conference Series (SIGGRAPH'03)*, pages 569–577, San Diego, CA, July 2003.

[CYB] Cybearware. http://www.cyberware.com.

[DBR00] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'00)*, Hilton Head Island, SC, June 2000.

[DC01] T. Drummond and R. Cipolla. Real-time tracking of highly articulated structures in the presence of noisy measurements. In *Proceedings of International Conference on Computer Vision (ICCV'01)*, pages 315–320, Vancouver, Canada, June 2001.

[DCR99] D. DiFranco, T. Cham, and J. Rehg. Recovering of 3D articulated motion from 2d correspondences. Technical Report CRL 99/7, Compaq Cambridge Research Laboratory, 1999.

[DCR01] D. Difranco, T. Cham, and J. Rehg. Reconstruction of 3D figure motion from 2D correspondences. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'01)*, Kauai, HI, December 2001.

[DF99] Q. Delamarre and O. Faugeras. 3D articulated models and multi-view tracking with silhouettes. In *Proceedings of International Conference on Computer Vision (ICCV'99)*, Corfu, Greece, September 1999.

[FGDP02] P. Fua, A. Gruen, N. D'Apuzzo, and R. Plänkers. Markerless full body shape and motion capture from video sequences. *International Archives of Photogrammetry and Remote Sensing*, 34(5):256–261, 2002.

[FHPB00] P. Fua, L. Herda, R. Plänkers, and R. Boulic. Human shape and motion recovery using animation models. In *XIX ISPRS Congress*, July 2000.

[GD96] G. Gavrila and L. Davis. Tracking of humans in action : 3D model-based approach. In *ARPA Image Understanding Workshop 1996*, February 1996.

[HHD98] I. Haritaoglu, D. Harwood, and L. S. Davis. W4 : Who? when? where? what? a real time system for detecting and tracking people. In *Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition (ICAFGR'98)*, Japan, 1998.

[JBY96] S. Ju, M. Black, and Y. Yacoob. Cardboard people: A parameterized model of articulated image motion. In *Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition (ICAFGR'96)*, Vermont, USA, October 1996.

[JTH99] N. Jojic, M. Turk, and T. Huang. Tracking self-occluding articulated objects in dense disparity maps. In *Proceedings of International Conference on Computer Vision (ICCV'99)*, Corfu, Greece, September 1999.

[KM95] I. Kakadiaris and D. Metaxas. 3D human body model acquisition from multiple views. In *Proceedings of International Conference on Computer Vision (ICCV'95)*, pages 618–623, Cambridge MA, June 1995.

[KM98] I. Kakadiaris and D. Metaxas. 3D human body model acquisition from multiple views. *International Journal on Computer Vision*, 30(3):191–218, 1998.

[KMB94] I. Kakadiaris, D. Metaxas, and R. Bajcsy. Active part-decomposition, shape and motion estimation of articulated objects: A physics-based approach. Technical Report IRCS Report 94-18, University of Pennsylvania, 1994.

[KYS01] N. Krahnstoever, M. Yeasin, and R. Sharma. Automatic acquisition and initialization of kinematic models. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'01), Technical Sketches*, Kauai, HI, December 2001.

[KYS03] N. Krahnstoever, M. Yeasin, and R. Sharma. Automatic acquisition and initialization of articulated models. In *To appear in Machine Vision and Applications*, 2003.

[LC01] D. Liebowitz and S. Carlsson. Uncalibrated motion capture exploiting articulated structure constraints. In *Proceedings of International Conference on Computer Vision (ICCV'01)*, Vancouver, Canada, June 2001.

[LY95] M. Leung and Y. Yang. First sight : A human body outline labeling system. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 17(4):359–377, April 1995.

[LZG98] M. Lucente, G. Zwart, and A. George. Visualization space: A testbed for deviceless multimodal user interface. In *Proceedings of AAAI Spring Symposium on Intelligent Environments*, Stanford, CA, 1998.

[Mat01] W. Matusik. Image-based visual hulls. Master's thesis, Massachusetts Institute of Technology, 2001.

[MET] Meta motion. http://www.metamotion.com.

[MG01] T. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding: CVIU*, 81(3):231–268, 2001.

[MHTC01] I. Mikic, E. Hunter, M. Trivedi, and P. Cosman. Articulated body posture estimation from multi-camera voxel data. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'01)*, Kauai, HI, December 2001.

[MLS94] R. Murray, Z. Li, and S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[MTG97] S. Moezzi, L. Tai, and P. Gerard. Virtual view generation for 3D digital video. *IEEE Computer Society Multimedia*, 4(1), January-March 1997.

[MTHC03] I. Mikic, M. Trivedi, E. Hunter, and P. Cosman. Human body model acquisition and tracking using voxel data. *International Journal on Computer Vision*, 53(3):199–223, July 2003.

[OBBH00] J. O'Brien, R. Bodenheimer, G. Brostow, and J. Hodgins. Automatic joint parameter estimation from magnetic motion capture data. In *Proceedings of Graphics Interface'00*, pages 53–60, May 2000.

[PF01] R. Plänkers and P. Fua. Articulated soft objects for video-based body modeling. In *Proceedings of International Conference on Computer Vision (ICCV'01)*, pages 394–401, Vancouver, Canada, June 2001.

[PFD99] R. Plänkers, P. Fua, and N. D'Apuzzo. Automated body modeling from video sequences. In *Proceedings of the 1999 International Workshop on Modeling People (MPEOPLE'99)*, Corfu, Greece, September 1999.

[PRCM99] V. Pavlovic, J. Rehg, T. Cham, and K. Murphy. A dynamic bayesian network approach to figure tracking using learned dynamic models. In *Proceedings of International Conference on Computer Vision (ICCV'99)*, Corfu, Greece, September 1999.

[RK95] J. Rehg and T. Kanade. Model-based tracking of self-occluding articulated objects. In *Proceedings of International Conference on Computer Vision (ICCV'95)*, pages 612–617, Cambridge MA, June 1995.

[SBF00] H. Sidenbladh, M. Black, and D. Fleet. Stochastic tracking of 3D human figures using 2D image motion. In *Proceedings of European Conference on Computer Vision (ECCV'00)*, Dublin, Ireland, June 2000.

[SC02] J. Sullivan and S. Carlsson. Recognizing and tracking human action. In *Proceedings of European Conference on Computer Vision (ECCV'02)*, Denmark, 2002.

[SDB00] H. Sidenbladh, F. DeLaTorre, and M. Black. A framework for modeling the appearance of 3D articulated figures. In *Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition (ICAFGR'00)*, March 2000.

[SKB$^+$98] S. Shafer, J. Krumm, B. Brumitt, B. Meyers, M. Czerwinski, and D. Robbins. The new easyliving project at microsoft research. In *Proceedings of Joint DARPA/NIST Smart Spaces Workshop*, Gaithersburgh, MD, July 1998.

[SMP03] P. Sand, L. McMillan, and J. Popovic. Continuous capture of skin deformation. In *Computer Graphics Annual Conference Series (SIGGRAPH'03)*, pages 578–586, San Diego, CA, July 2003.

[TTI] Thirdtech inc. http://www.3rdtech.com.

[VIC] Vicon motion systems. http://www.vicon.com.

[WADP97] C. 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, July 1997.

[YSK$^+$98] M. Yamamoto, A. Sato, S. Kawada, T. Kondo, and Y. Osaki. Incremental tracking of human actions from multiple views. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'98)*, volume 1, pages 2–7, CA, June 1998.