Shape-From-Silhouette of Articulated Objects and its Use for Human Body Kinematics Estimation and Motion Capture

German K.M. Cheung Simon Baker Takeo Kanade
Robotics Institute, Carnegie Mellon University,
Pittsburgh PA 15213
german+@cs.cmu.edu simonb+@cs.cmu.edu tk+@cs.cmu.edu

Abstract

Shape-From-Silhouette (SFS), also known as Visual Hull (VH) construction, is a popular 3D reconstruction method which estimates the shape of an object from multiple silhouette images. The original SFS formulation assumes that all of the silhouette images are captured either at the same time or while the object is static. This assumption is violated when the object moves or changes shape. Hence the use of SFS with moving objects has been restricted to treating each time instant sequentially and independently. Recently we have successfully extended the traditional SFS formulation to refine the shape of a rigidly *moving object over time. Here we further extend SFS to apply to dynamic* articulated *objects. Given silhouettes of a moving articulated object, the process of recovering the shape and motion requires two steps: (1) correctly segmenting (points on the boundary of) the silhouettes to each articulated part of the object, (2) estimating the motion of each individual part using the segmented silhouette. In this paper, we propose an iterative algorithm to solve this simultaneous assignment and alignment problem. Once we have estimated the shape and motion of each part of the object, the articulation points between each pair of rigid parts are obtained by solving a simple motion constraint between the connected parts. To validate our algorithm, we first apply it to segment the different body parts and estimate the joint positions of a person. The acquired kinematic (shape and joint) information is then used to track the motion of the person in new video sequences.*

1. Introduction

Traditional Shape-From-Silhouette (SFS) assumes either that all of the silhouette images are captured at the same time or that the object is static [15, 18, 14]. Although systems have been proposed to apply SFS to video [5, 2], these systems apply SFS to each frame sequentially and independently. Recently there has been some work on using SFS on rigidly moving objects to recover shape and motion [21, 22], or to refine the shape over time [4]. These methods involve the estimation of the 6 DOF rigid motion of the object between successive frames. In [22] the motion is assumed to be circular. Frontier points are extracted from the silhouette boundary and used to estimate the axis of rotation. In [21], Ponce et al. define a local parabolic structure on the surface of a *smooth curved* object and use epipolar geometry to locate corresponding frontier points on three silhouette images. The motion between the images is then estimated by a two-step minimization.

In [4] the 6 DOF motion is estimated by combining both the silhouette and the color information. At each time instant, 3D line segments called Bounding Edges are constructed from rays through the camera centers and points on the silhouette boundary. Using the fact that each Bounding Edge touches the object at at least one point, a multi-view stereo algorithm is proposed to extract the colors and positions of these touching points (subsequently referred to as Colored Surface Points). The motion between consecutive frames is then computed by minimizing the errors of projecting the Colored Surface Points into the images. Once the 6 DOF rigid motion is recovered and compensated for, all the silhouette images are treated as taken at the same time and traditional SFS is applied to get a refined shape of the object.

In this paper we extend [4] to handle articulated objects. An articulated object consists of a set of rigidly moving parts which are connected to each other at certain articulation points. A good example of an articulated object is the human body (if we approximate the body parts as rigid). Here we propose an algorithm to automatically recover the joint positions, and the shape and motion of each part of an articulated object. We begin with silhouette images, although color information is used to break the alignment ambiguity as in [4].

Given silhouettes of a moving articulated object, recovering the shape and motion requires two inter-related steps: (1) correctly segment (points on the boundary of) the silhouettes to each part of the object and (2) estimate the shape and motion of the individual parts. We propose an iterative algorithm to solve this simultaneous assignment and 6 DOF motion estimation problem. Once the motions of the rigid parts are known, their articulation points are estimated by imposing motion constraints between adjoining parts. To test our algorithm, we apply it to acquire the shape and joint locations of articulated human models. Once this kinematic information of the person has been acquired, we show how the 6 DOF motion estimation algorithm can be used to track the articulated motion of that person in new video sequences. Results on both synthetic and real data are presented to show the validity of our algorithms.

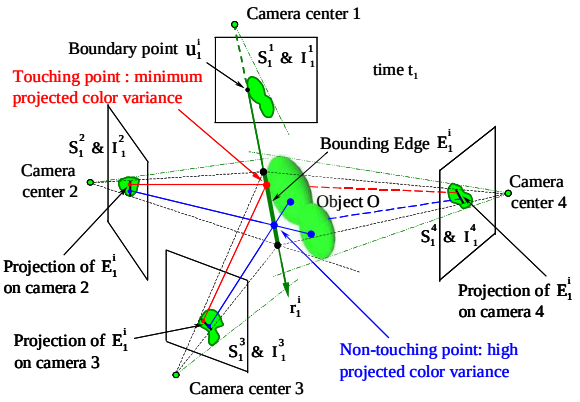


Figure 1. The Bounding Edge E_1^i is obtained by first projecting the ray r_1^i onto S_1^2, S_1^3, S_1^4 and then re-projecting the segments overlapping the silhouettes back into 3D space. E_1^i is the intersection of the reprojected segments. The point where the object touches E_1^i is located by searching along E_1^i for the point with the minimum projected color variance. Note that the image from camera 4 is not used because it is occluded. See [4] for details.

2. Background

In [3] and [4] we extended the traditional SFS formulation to rigidly moving objects. Combining the silhouette and color images, we first extract 3D points on the surface of the object at each time instant. These surface points are then used to estimate the 6 DOF motion between successive frames. Once the rigid motion across time is known, all of the silhouette images are treated as being captured at the same time and SFS is performed to estimate the shape of the object. Below we give a brief review of this temporal SFS algorithm.

2.1. Visual Hulls and Their Bounding Edges

The term Visual Hull (VH) was first coined by Laurentini in [13] to denote the 3D shape obtained by intersecting the visual cones formed by the silhouette images and the camera centers [13, 14, 2]. One useful property of a VH is that it provides an upper bound on the shape of the object. In [4], we introduced a new representation of a VH called the Bounding Edge representation. Assume there are K color-balanced and calibrated cameras positioned around a Lambertian object $\boldsymbol{O}$. Let $\{I_j^k; \quad S_j^k; \quad k = 1, \ldots, K\}$ be the set of color and corresponding silhouette images of the object $\boldsymbol{O}$ obtained from the K cameras at time t_j. Let u_j^i be a point on the boundary of the silhouette image S_j^k. Through the center of camera k, u_j^i defines a ray r_j^i in 3D space. A Bounding Edge E_j^i is defined as the portion of r_j^i such that the projection of E_j^i on the image planes of all the other cameras lies completely inside the silhouettes. An example is shown in Figure 1. E_j^i can be constructed by successively projecting the ray r_j^i onto each silhouette image, and retaining the portion whose projection overlaps all the silhouettes.

2.2. Colored Surface Points (CSP)

The most important property of a Bounding Edge is the Second Fundamental Property of Visual Hulls (2nd FPVH) which states that each Bounding Edge touches the object (which forms the silhouette images) at at least one point [4]. Using this property, we are able to locate points on the surface of the object using a multi-stereo color matching approach. Consider a Bounding Edge E_j^i. Since we assume the object is Lambertian and the cameras are color balanced, there exists at least one point on E_j^i (the point where it touches the object) such that the projected colors of this point in all the *visible* color images I_j^k are the same. In other words, this point has zero projected color variance among the visible color images. In practice, due to noise and inaccuracies in color balancing, instead of searching for the point that has zero projected color variance, we assign the touching point on E_j^i to be the point with the minimum color variance, as shown in Figure 1. We refer to this point as a Colored Surface Point (CSP) of the object and represent its position and color (which is obtained by averaging its projected color across all visible cameras) by W_j^i and μ_j^i respectively. By sampling the boundaries of all the silhouette images, a set of L_j Colored Surface Points can be constructed. Note that there is *no* point-to-point correspondence relationship between two different sets of CSPs obtained at different time instant. The only property common to the CSPs is that they all lie on the surface of the object.

2.3. SFS Across Time for Rigid Objects

We now describe our algorithm for recovering the 6 DOF motion of a rigid object using the CSPs. Without loss of generality, we assume that the orientation and position of the object at time t_1 is $(\boldsymbol{I}, \mathbb{0})$ and that at time t_j it is $(\boldsymbol{R}_j, \boldsymbol{t}_j); \quad j = 2, \ldots, J$. The rigid object alignment problem is then equivalent to recovering $(\boldsymbol{R}_j, \boldsymbol{t}_j) \ \forall j$. Consider the motion between t_1 and t_2 as an example and assume we have two sets of data $\{I_j^k, S_j^k, W_j^i, \mu_j^i; \ i = 1, \ldots, L_j; \ k = 1, \cdots\cdots, K; \quad j = 1, 2\}$ obtained at t_1 and t_2 respectively. To find $(\boldsymbol{R}_2, \boldsymbol{t}_2)$, we align the CSPs with the 2D silhouette and color images. The idea is very similar to that in [19] for 2D image alignment.

Suppose we have an estimate of $(\boldsymbol{R}_2, \boldsymbol{t}_2)$. For a CSP W_1^i (with color μ_1^i) at time t_1, its 3D position at time t_2 would be $\boldsymbol{R}_2 W_1^i + \boldsymbol{t}_2$. Consider two different cases of the projection of $\boldsymbol{R}_2 W_1^i + \boldsymbol{t}_2$ into the k^{th} camera:

1. The projection lies inside the silhouette S_2^k. In this case, we use the color difference as an error measure:

$$[\boldsymbol{c}_2^k(\boldsymbol{R}_2 W_1^i + \boldsymbol{t}_2) - \mu_1^i]^2 \,, \tag{1}$$

where $\boldsymbol{c}_2^k(P)$ is the projected color of a 3D point P into the color image I_2^k. Here we assume this color error is zero if the projection of P lies outside S_2^k.

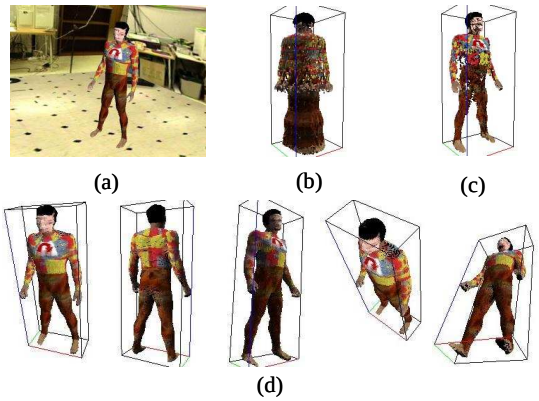


Figure 2. Results of our temporal SFS algorithm [4] applied to synthetic data: (a) one of the input images, (b) unaligned CSPs, (c) aligned CSPs, (d) refi ned visual hull.

2. The projection lies outside S_2^k. In this case, we use the distance of the projection from S_2^k, represented by $d_2^k(\boldsymbol{R}_2 W_1^i + \boldsymbol{t}_2)$ as an error measure. The distance is zero if the projection lies inside S_2^k.

Summing over all cameras in which W_1^i is visible, the error measure of W_1^i with respect to $(\boldsymbol{R}_2, \boldsymbol{t}_2)$ is given by

$$e_{2,1}^i = \sum_k \{\tau * d_2^k(\boldsymbol{R}_2 W_1^i + \boldsymbol{t}_2) + [\boldsymbol{c}_2^k(\boldsymbol{R}_2 W_1^i + \boldsymbol{t}_2) - \mu_1^i]^2\} \quad , \qquad (2)$$

where τ is a weighing constant. Similarly, the error measure of a CSP W_2^i at time t_2 is written as

$$e_{1,2}^i = \sum_k \{\tau * d_1^k(\boldsymbol{R}_2^T(W_2^i - \boldsymbol{t}_2)) + [\boldsymbol{c}_1^k(\boldsymbol{R}_2^T(W_2^i - \boldsymbol{t}_2)) - \mu_2^i]^2\} \quad . \qquad (3)$$

Now the problem of estimating the motion $(\boldsymbol{R}_2, \boldsymbol{t}_2)$ is posed as minimizing the total error

$$\min_{\boldsymbol{R}_2, \boldsymbol{t}_2} e = \min_{\boldsymbol{R}_2, \boldsymbol{t}_2} \sum_{i=1}^{L_2} e_{1,2}^i + \sum_{i=1}^{L_1} e_{2,1}^i \, , \qquad (4)$$

which can be solved using a gradient descent or Iterative Levenberg-Marquardt algorithm [20]. Hereafter we refer to this motion estimation process as either "temporal SFS" or the visual hull alignment algorithm.

To show the validity of our visual hull alignment algorithm, we apply it to both synthetic and real sequences of a rigidly moving person. In the synthetic sequence, a computer graphics model of a person is made to rotate about the z-axis. Twenty five sets of color and silhouette images of the model from eight virtual cameras are rendered using OpenGL. One example of the rendered color images is shown in Figure 2(a). CSPs are then extracted and aligned. Figures 2(b) and (c) illustrate respectively the unaligned and aligned CSPs for all 25 frames. Figure 2(d) shows the visual hull constructed by applying SFS to all the silhouette images (compensating for the alignment). A real sequence of a person standing on a turntable (with unknown speed and rotation axis) was also captured by eight cameras with

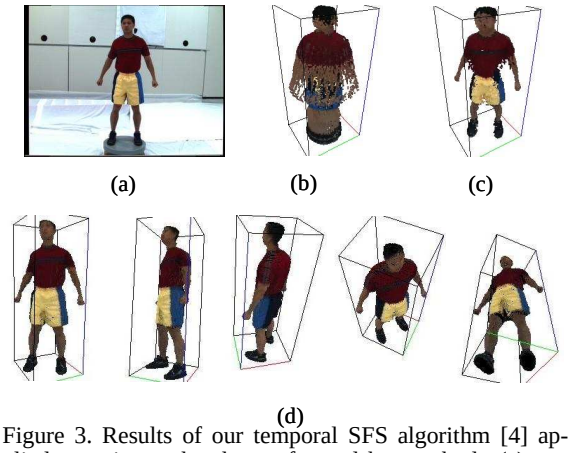


Figure 3. Results of our temporal SFS algorithm [4] applied to estimate the shape of a real human body (a) one of the input images, (b) unaligned CSPs, (c) aligned CSPs, (d) refi ned visual hull displayed from several different view points.

thirty frames per camera. The person was asked to remain still throughout the capture process to satisfy the rigidity assumption. The results are presented in Figure 3. It can be seen that excellent shape estimates (the visual hulls shown in Figure 2(d) and Figure 3(d)) of the human bodies can be obtained using our temporal SFS algorithm [4].

Although the 3D shape of a person can be obtained in detail using the VH alignment algorithm described above, the acquired shape does not contain any kinematic information. Kinematic information is essential for applications such as motion tracking, capture, recognition and rendering. We now show how this information can be obtained automatically and accurately using temporal SFS algorithm for articulated objects.

3. SFS for Articulated Objects

To extend the temporal SFS algorithm to articulated objects we employ an idea similar to that used for multiple layered motion estimation in [16]. The rigid parts of the articulated object are first modeled as separate and independent of each other. With this assumption, we iteratively (1) assign the extracted CSPs to different parts of the object and (2) apply the temporal SFS algorithm to align each part across time. Once the motions of all the parts are recovered, an articulation constraint is applied to estimate the joint positions. Note that this iterative approach can be categorized as belonging to the Expectation Maximization framework [7]. The whole algorithm is explained below in detail using a two-part, one-joint articulated object.

3.1. Segmentation/Alignment Algorithm

Consider an one-joint object $\boldsymbol{O}$ which consists of two rigid parts A and B as shown in Figure 4 at two different time instants t_1 and t_2. Assume CSPs of the object are extracted from the color and silhouette images of K calibrated and color-balanced cameras, denoted by $\{I_j^k, S_j^k, W_j^i, \mu_j^i; \quad j = 1, 2\}$. Furthermore, treating A

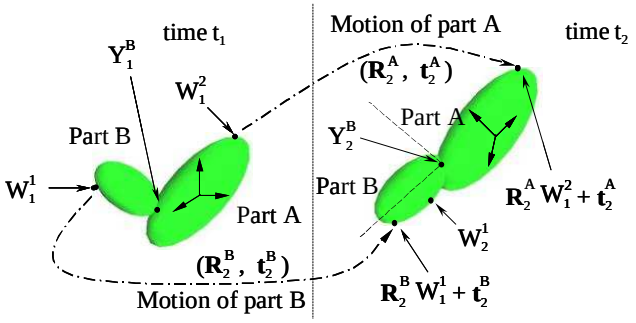


Figure 4. A two-part articulated object at two different time instants t_1 and t_2.

and B as two independently moving rigid objects allows us to represent the relative motion of A between t_1 and t_2 as $(\boldsymbol{R}_2^A, \boldsymbol{t}_2^A)$ and that of B as $(\boldsymbol{R}_2^B, \boldsymbol{t}_2^B)$. Now consider the following two complementary cases.

3.1.1. Alignment with known segmentation

Suppose we have segmented the CSPs at t_j into groups belonging to part A and part B, represented by G_j^A and G_j^B respectively for both $j = 1, 2$. By applying the temporal SFS algorithm described in Section 2 (Eq. (4)) to A and B separately, estimates of the relative motions $(\boldsymbol{R}_2^A, \boldsymbol{t}_2^A), (\boldsymbol{R}_2^B, \boldsymbol{t}_2^B)$ are obtained.

3.1.2. Segmentation with known alignment

Assume we are given the relative motion $(\boldsymbol{R}_2^A, \boldsymbol{t}_2^A), (\boldsymbol{R}_2^B, \boldsymbol{t}_2^B)$ of A and B from t_1 to t_2. For a CSP W_1^i at time t_1, consider the following two error measures

$$e_{2,1}^{i,A} = \sum_k \{\tau * d_2^k (\boldsymbol{R}_2^A W_1^i + \boldsymbol{t}_2^A) + [\boldsymbol{c}_2^k (\boldsymbol{R}_2^A W_1^i + \boldsymbol{t}_2^A) - \mu_1^i]^2\} \quad , \qquad (5)$$

$$e_{2,1}^{i,B} = \sum_k \{\tau * d_2^k (\boldsymbol{R}_2^B W_1^i + \boldsymbol{t}_2^B) + [\boldsymbol{c}_2^k (\boldsymbol{R}_2^B W_1^i + \boldsymbol{t}_2^B) - \mu_1^i]^2\} \quad . \qquad (6)$$

Here $e_{2,1}^{i,A}$ is the error of W_1^i with respect to the color/silhouette images at t_2 if it belongs to part A (thus following the motion model $(\boldsymbol{R}_2^A, \boldsymbol{t}_2^A)$). Similarly $e_{2,1}^{i,B}$ is the error if W_1^i lies on the surface of B. In these expressions the summations are over all visible cameras k. By comparing these two errors, a simple strategy to classify the point W_1^i is devised as follows:

$$W_1^i \in \begin{cases} G_1^A & \text{if} \quad e_{2,1}^{i,A} < \kappa * e_{2,1}^{i,B} \\ G_1^B & \text{if} \quad e_{2,1}^{i,B} < \kappa * e_{2,1}^{i,A} \\ G_1^\emptyset & \text{otherwise} \end{cases} \quad , \qquad (7)$$

where $0 \leq \kappa \leq 1$ is a thresholding constant and $G_1^\emptyset$ contains all the CSPs which are classified as neither belonging to part A nor part B. Similarly, the CSPs at time t_2 can be classified using the errors $e_{1,2}^{i,A}$ and $e_{1,2}^{i,B}$.

In practice, the above decision rule does not work very well because of image/silhouette noise and camera calibration errors. Here we suggest using spatial coherency and temporal consistency to improve the segmentation. To use spatial coherency, the notion of a spatial neighborhood has to be defined. Since it is difficult to define a spatial neighborhood for the scattered CSPs in 3D space (see for example Figure 3(b)), an alternate way is used. Recall that each CSP W_1^i lies on a Bounding Edge which in turn corresponds to a boundary point u_1^i of the silhouette image S_1^k. We define two CSPs W_1^i and W_1^{i+1} as "neighbors" if their corresponding 2D boundary points u_1^i and u_1^{i+1} are neighboring pixels (in 8-connectivity sense) in the same silhouette image. This neighborhood definition allows us to easily apply spatial coherency to the CSPs. From Figure 5(a) it can be seen that different parts of an articulated object *usually* project onto the silhouette image as continuous outlines. Inspired by this property, the following spatial coherency rule (SCR) is proposed:

Spatial Coherency Rule (SCR):
If W_1^i is classified as belonging to part A by Eq. (7), it stays as belonging to part A if all of its m left and right immediate "neighbors" are also classified as belonging to part A by Eq. (7), otherwise it is reclassified as belonging to $G_1^\emptyset$. The same procedure applies to part B.

Figure 5(a) shows how the spatial coherency rule can be used to remove spurious partition error. The second constraint we utilize to improve the segmentation results is temporal consistency as illustrated in Figure 5(b). Consider three successive frames captured at t_{j-1}, t_j and t_{j+1}. For a CSP W_j^i, it has two classifications due to motion from t_{j-1} to t_j and motion from t_j to t_{j+1}. Since W_j^i either belongs to part A or B, the temporal consistency rule (TCR) simply requires that the two classifications have to agree with each other:

Temporal Consistency Rule (TCR):
If W_j^i has the same classification by SCR from t_{j-1} to t_j and from t_j to t_{j+1}, the classification is maintained, otherwise, it is reclassified as belonging to $G_j^\emptyset$.

Note that SCR and TCR not only remove wrongly segmented points, but they also remove some of the correctly classified CSPs. Overall though they are effective because few but more accurate data is always preferred over abundant but less accurate data, especially in our case where the segmentation has a great effect on motion estimation.

3.1.3. Iterative algorithm

Summarizing the above discussion, we propose an iterative segmentation/alignment process to estimate the shape and motion of parts A and B over J frames as follows :

1. Given segmentations $\{G_j^A, G_j^B\}$ of CSPs, recover the relative motions $\{(\boldsymbol{R}_j^A, \boldsymbol{t}_j^A)\ ; (\boldsymbol{R}_j^B, \boldsymbol{t}_j^B)\}$ of A and B over all frames $j = 2, \ldots J$ using the temporal SFS algorithm.

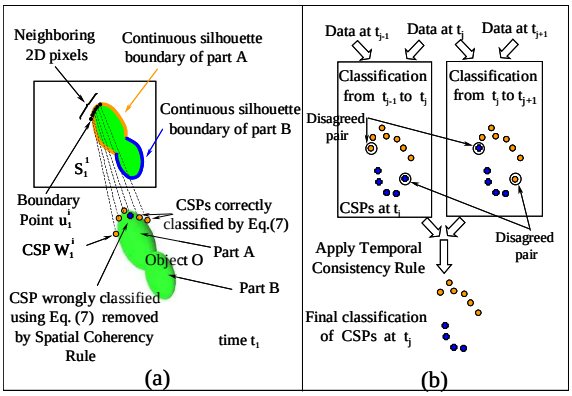


Figure 5. (a) Spatial coherency removes spurious segmentation errors, (b) Temporal consistency ensures segmentation agrees between successive frames.

2. Repartition the CSPs according to the estimated motions by applying Eq. (7), followed by the intra-frame SCR and inter-frame TCR.

3. Repeat Steps 1 and 2 until the segmentation/alignment converges or for a fixed maximum number of times.

Although for the sake of explanation we have described this algorithm for an articulated object with two rigid parts, it can easily be generalized to apply to objects with N parts.

3.2. Initialization

As common to all iterative EM algorithms, initialization is always a problem [16]. Here we suggest two different approaches to start our algorithm. Both approaches are commonly used in the layer estimation literature [16]. The first approach uses the fact that the 6 DOF motion of each part of the articulated object represents a single point in a six dimensional space. In other words, if we have a large set of estimated motions of all the parts of the object, we can apply clustering algorithms on these estimates in the 6D space to separate the motion of each individual part. To get a set of estimated motions for all the parts, the following method is used. The CSPs at each time instant are first divided into subgroups by cutting the corresponding silhouette boundaries into arbitrary segments. These subgroups of CSPs are then used to generate the motion estimates using the VH alignment algorithm, each time with a randomly chosen subgroup from each time instant. Since this approach requires the clustering of points in a 6D space, it performs best when the motions between different parts of the articulated object are relatively large so that the motion clusters are distinct from each other.

The second approach is applicable in situations where one part of the object is much larger than the other. Assume, say, part A is the dominant part. Since this assumption means that most of the CSPs of the object belong to A, the dominant motion $(\boldsymbol{R}^A, \boldsymbol{t}^A)$ of A can be approximated using all the CSPs. Once an approximation of $(\boldsymbol{R}^A, \boldsymbol{t}^A)$ is available, the CSPs are sorted in terms of their errors with respect to this dominant motion. An initial segmentation is then obtained by thresholding the sorted CSPs errors.

3.3. Articulation Point Estimation

After recovering the motions of parts A and B separately, the point of articulation between them is estimated. Suppose we represent the joint position at time t_1 as Y_1^B. Since Y_1^B lies on both A and B, it must satisfy the motion equation from t_1 to t_2 as follows

$$\boldsymbol{R}_2^A Y_1^B + \boldsymbol{t}_2^A = \boldsymbol{R}_2^B Y_1^B + \boldsymbol{t}_2^B \ . \tag{8}$$

Putting together similar equations for Y_1^B over J frames, we get

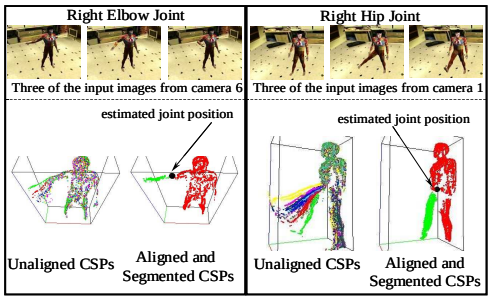
$$\begin{bmatrix} \boldsymbol{R}_2^A - \boldsymbol{R}_2^B \\ \vdots \\ \boldsymbol{R}_J^A - \boldsymbol{R}_J^B \end{bmatrix} Y_1^B = \begin{bmatrix} \boldsymbol{t}_2^B - \boldsymbol{t}_2^A \\ \vdots \\ \boldsymbol{t}_J^B - \boldsymbol{t}_J^A \end{bmatrix} . \tag{9}$$

The least squares solution of Eq. (9) can be computed using Singular Value Decomposition.

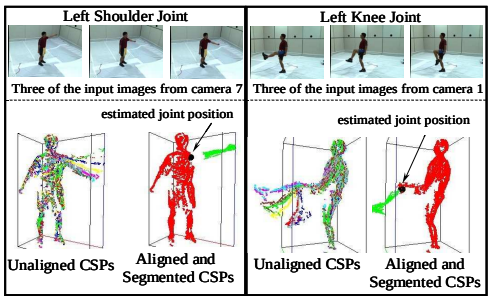
3.4. Human Body Kinematics Acquisition

Here we apply our SFS algorithm for articulated objects to segment the body parts and to estimate the joint positions of a person. Instead of estimating all the joints at the same time, we take a sequential approach and model the joints one by one. To find the position of, say the left shoulder joint, the person is asked to move his whole left arm around the shoulder while keeping the rest of the body still. This makes the human body a one-articulation point object. Since the size of the whole body is much larger than a single body part, the dominant motion initialization method is used. Figure 6(a) shows some of the input images and the results for the right elbow and the right hip joints of the computer graphics model used in the synthetic sequence (Figure 2) at the end of Section 2. Figure 6(b) presents some of the input images and the results for the left shoulder and the left knee joints of the person in Figure 3. The input images, CSPs and results for the left hip/knee joints of the synthetic data set can be seen in the movie syn-kinematics-leftleg.mpg and those for the right shoulder/elbow and right hip/knee joints of the real person in the movie real-kinematics-rightarm.mpg and real-kinematics-rightleg.mpg respectively. All the movie sequences mentioned in this paper can be found at http://www.cs.cmu.edu/~german/research/CVPR2003/HumanMT.

To create a complete articulated human model (after each body part is segmented and its joint position is located using our SFS algorithm for articulated objects) the various body parts are aligned to the whole body voxel model acquired at the end of Section 2 (Figure 2(d) for the synthetic data and Figure 3(d) for the real person). The alignment is done between the 3D CSPs of the body part and the reference image of the sequences that are used to obtain the whole body voxel model. Figure 7(a) displays the complete articulated model of the synthetic data set with the joint locations and



(a). Results of right elbow and right hip joints for the synthetic data set



(b). Results of left shoulder and left knee joints for the real human

Figure 6. (a). Estimated right elbow and right hip joints of a synthetic data set. (b). Estimated left shoulder and left knee joints of a real data set. For each joint, the unaligned CSPs from different frames are drawn with different colors. The aligned and segmented CSPs are shown with two different colors to show the segmentation. The estimated articulation point (joint location) is indicated by the black sphere.

segmented body parts (shown in terms of the 3D points derived from the voxel centers of the model). We have also added a skeleton by joining the joint locations together. The articulated model of the real person is shown in Figure 7(b).

The work most similar to our vision-based human body kinematic information acquisition is by Kakadiaris et al. in [12]. They first use deformable templates to segment 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. Here we address the acquisition of motion, shape and articulation information at the same time, while [12] focuses mainly on shape estimation.

4. Application: Motion Capture

Due to increased applications in entertainment, security/surveillance and human-computer interaction, the problem of vision-based motion capture has gained much attention in recent years. Several researchers have proposed systems to track body parts from video sequences [10, 1, 11, 6, 5, 17, 8, 9]. In most of these systems, generic shapes (e.g. rectangles/ellipses in 2D, cylinders/ellipsoids in 3D) are used to model the body parts of the person. Although generic models/shapes are simple to use and can be generalized to different persons easily, they suffer from two disadvantages. Firstly they only coarsely approximate the actual body shape of the person. Secondly generic shapes/models also lack accurate joint information of the

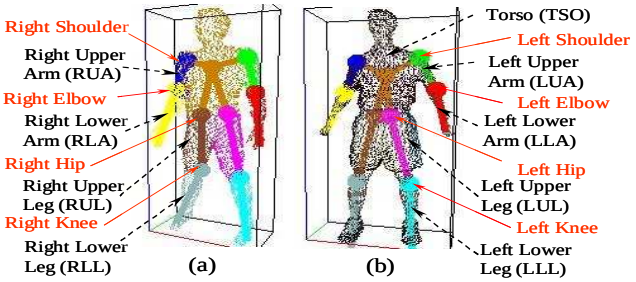


Figure 7. Complete articulated human model of (a) the synthetic data set (different body parts shown with different colors), (b) the real person. These are the models used for motion tracking in the experimental results in Section 4.3.

person. In vision-based motion capture systems, precise kinematic information (shape and joint) is essential to obtain accurate motion data. Here we show how to use the acquired human kinematic model of the person in the previous section to perform motion capture from color and silhouette image sequences. As compared to other systems which use either only color images [1, 17] or only silhouette images [6, 5], our algorithm combines both silhouette and color information to fit the articulated model.

4.1. Human Body Model

The articulated human model used in our tracking algorithm is the same as those depicted in Figure 7. It consists of nine body parts (torso, right/left lower/upper arms, right/left lower/upper legs) connected by eight joints (right/left shoulder/elbow joints, right/left hip/knee joints). Each body part is assumed to be rigid with the torso being the base. The shoulder and hip joints have 3 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.

4.2. Tracking with An Articulated Model

Assume we have estimated the kinematic information of all nine body parts of the person at a reference time t_1 with color and silhouette images I_1^k, S_1^k. Represent the shape of body part Z in terms of a set of CSPs as $(W_1^{i,Z}, \mu_1^{i,Z}; \quad i = 1, \cdots\cdots, L_1^Z)$, its joint as Y_1^Z and call this the model data set. Now suppose we are given the run-time data set at t_j, which consists of K color/silhouette images and the corresponding CSPs $\{I_j^k, S_j^k, W_j^i\}$ of the person. Let $\boldsymbol{Q}_j^Z$ be the rotation matrix of Z at its joint Y_j^Z and $\boldsymbol{s}_j^{TSO}$ be the translation of the torso base at t_j. Without loss of generality, assume $\boldsymbol{s}_1^{TSO}$ is zero and $\boldsymbol{Q}_1^Z$ is the identity matrix for all body parts at t_1. The motion capture problem can be posed as estimating $\boldsymbol{s}_j^{TSO}$ and $\boldsymbol{Q}_j^Z$ for all the body parts Z from the color and the silhouette images $\{I_j^k, S_j^k\}$.

The most straightforward way to solve the above motion capture problem is to align *all* the body parts (with a total of 22 DOF) of the human model directly to the silhouette

and color images *all at once*. Although this all-at-once approach can be done by generalizing the temporal SFS algorithm to perform a non-linear optimization over all 22 DOF, in practice it is prone to the problem of falling into local minima because of the high dimensionality. To avoid this local minimum problem, we instead use a two-step hierarchical approach: first fit the torso base and then 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, under most of the cases, largely independent of each other.

The first step of our hierarchical approach involves recovering the global translation and orientation $(\boldsymbol{Q}_j^{TSO}, \boldsymbol{s}_j^{TSO})$ of the torso base. This can be done by using the 6 DOF temporal SFS algorithm described in Section 2. Once the global motion of the body is estimated, the four joint positions: left/right shoulders and left/right hips are calculated. The four limbs of the body are then aligned separately around these fixed joint positions in the second step. For each limb, the two joint rotations (shoulder and elbow for arms, hip and knee for legs) are estimated simultaneously. We briefly explain the second step below using the left arm and time t_2 as an example. Here only the errors of projecting the model CSPs onto the run-time color/silhouette images are considered. This can be extended to include the projection errors of the run-time CSPs by segmenting them to individual part of the body.

Assume we have recovered the torso translation and orientation $(\boldsymbol{Q}_2^{TSO}, \boldsymbol{t}_2^{TSO})$, then the joint location Y_2^{LUA} and the transformed position $\bar{W}_1^{i,LUA}$ of a model CSP $W_1^{i,LUA}$ on the left upper arm (LUA) at time t_2 are expressed as

$$Y_2^{LUA} = \boldsymbol{Q}_2^{TSO} Y_1^{LUA} + \boldsymbol{s}_2^{TSO} \ ,$$

$$\bar{W}_1^{i,LUA} = \boldsymbol{Q}_2^{TSO} \boldsymbol{Q}_2^{LUA} (W_1^{i,LUA} - Y_1^{LUA}) + Y_2^{LUA} \ .$$

Using these and Eq. (2), we can express the sum of projected color/silhouette error $e_{2,1}^{i,LUA}$ of $\bar{W}_1^{i,LUA}$ across visible cameras at t_2 as a function of the unknown $\boldsymbol{Q}_2^{LUA}$. Similarly, the error $e_{2,1}^{i,LLA}$ for each CSP on the Left Lower Arm (LLA) can be written as function of $\boldsymbol{Q}_2^{LUA}$ and $\boldsymbol{Q}_2^{LLA}$. By optimizing the combined errors of the whole left arm as

$$\min_{\boldsymbol{Q}_2^{LUA}, \boldsymbol{Q}_2^{LLA}} [\sum_{i=1}^{L_1^{LLA}} e_{2,1}^{i,LLA} + \sum_{i=1}^{L_1^{LUA}} e_{2,1}^{i,LUA}] \ , \qquad (10)$$

the joint rotation matrices are estimated. This simultaneous estimation approach, as compared to estimating the joint rotations (e.g. first shoulder and then elbow) of the limb individually and sequentially, is better because both joint constraints are incorporated implicitly into the equations at the same time.

4.3. Experimental Results

4.3.1. Synthetic sequences

Two synthetic motion video sequences: KICK (60 frames) and PUNCH (72 frames) were generated using the synthetic human model in Figure 2(a). A total of eight cameras are used. The complete articulated model shown in Figure 7(a) is used to track the motion in these sequences. Figure 8 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 file syn-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 color) overlaid on one of the input images (which are converted from color to gray-scale for clarity). The lower left corner depicts the ground-truth motion rendered through an avatar and the lower right corner represents the tracked motions rendered through the same avatar. The avatar renderings show that the ground-truth and tracked motions are almost indistinguishable from each order.

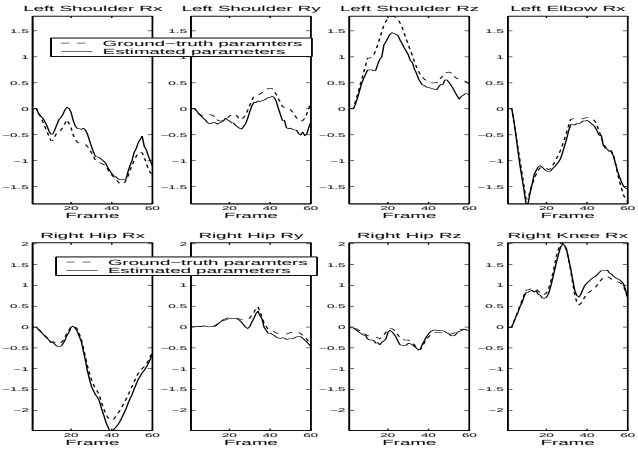
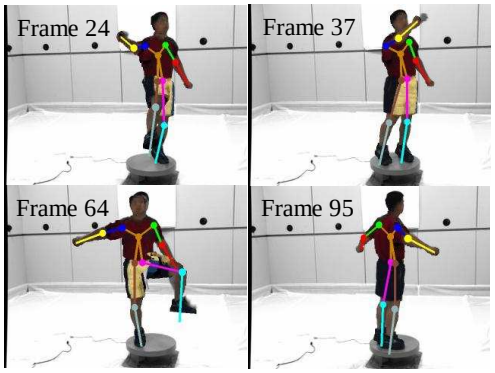


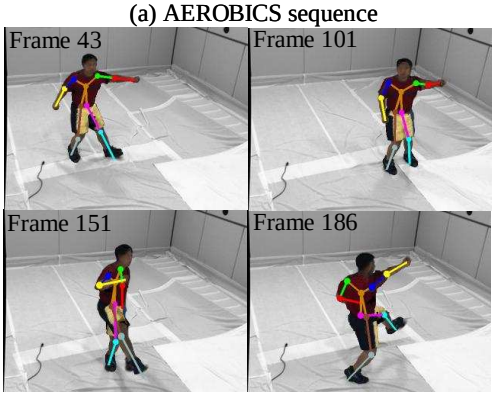
Figure 8. Graphs comparing ground-truth and estimated joint angles of the left arm and right leg of the synthetic sequence KICK. The estimated joint angles closely follow the ground-truth values throughout the whole sequence.

4.3.2. Real sequences

Three video sequences: STILLMARCH (158 frames) , AEROBICS (110 frames) and KUNGFU (200 frames) of the real person in Figure 3(a) were captured to test the tracking algorithm. Eight cameras are used in each sequence and the articulated model in Figure 7(b) acquired in Section 3.4 is used. Figures 9(a)(b) show the tracking results on the AEROBICS and KUNGFU sequences respectively. Each figure shows four selected frames of the sequence with the (color) tracked body parts and the joint skeleton overlaid on one of the eight camera (turned gray-scale) input images. The movie real-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 joint angles to a 3D articulated visual hull (voxel) model (obtained by combining the results in Figure 3(d) and the kine-



(a) AEROBICS sequence



(b) KUNGFU sequence

Figure 9. Tracking results for the two sequences (a) AEROBICS and (b) KUNGFU. Each set of images contains selected frames of the tracked body parts and joint skeleton (rendered color) overlaid on one of the input camera images (which are converted from color to gray-scale for clarity). All the frames of both sequences can be seen in the movie real-track.mpg.

matic information) of the person while the lower right corner shows the results of applying the estimated motion data to an avatar. This video demonstrates that our algorithm is able to track the body parts and joint angles correctly in difficult real sequences, although in 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.

5. Summary

We have proposed a SFS algorithm for articulated objects to recover the motion, shape and joints of an articulated object from silhouette and color images. The algorithm iteratively segments points on the silhouettes to each articulated part of the object and estimates the motion of each individual part using the segmented silhouette. Once the motion/shape of each part is recovered, the joints are estimated by articulation constraints. We applied our articulated SFS algorithm to acquire the kinematic information (shape of body parts and joint positions) of a person and then used the model to track the person in new video sequences.

References

[1] C. Bregler and J. Malik. Video motion capture. Technical Report CSD-97-973, UCB, 1997.

[2] C. Buehler, W. Matusik, L. McMillan, and S. Gortler. Creating and rendering image-based visual hulls. Technical Report MIT-LCS-TR-780, MIT, 1999.

[3] G. K. M. Cheung. Visual hull construction, alignment and refi nement across time. Technical Report CMU-RI-TR-02-05, Carnegie Mellon University, January 2002.

[4] G. K. M. Cheung, S. Baker, and T. Kanade. Visual hull alignment and refi nement across time:a 3D reconstruction algorithm combining shape-frame-silhouette with stereo. In *Proc. of CVPR'03*, June 2003.

[5] G. K. M. Cheung, T. Kanade, J. Bouquet, and M. Holler. A real time system for robust 3D voxel reconstruction of human motions. In *Proceedings CVPR'00*, Hilton Head Island, SC, June 2000.

[6] Q. Delamarre and O. Faugeras. 3D articulated models and multi-view tracking with silhouettes. In *Proc. of ICCV'99*, Sept. 1999.

[7] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *J.R. Statistic. Soc.*, B 39:1–38, 1977.

[8] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle fi ltering. In *Proceedings CVPR'00*, Hilton Head Island, SC, June 2000.

[9] D. E. Difranco, T. J. Cham, and J. M. Rehg. Reconstruction of 3D fi gure motion from 2D correspondences. In *Proceedings CVPR'01*, Kauai, HI, December 2001.

[10] G. Gavrila and L. Davis. Tracking of humans in action : 1 3D model-based approach. In *ARPA Image Understanding Workshop '96*, Palm Springs, CA, Feb. 1996.

[11] I. Haritaoglu, D. Harwood, and L. S. Davis. W4 : Who? when? where? what? a real time system for detecting and tracking people. In *Proc. of IEEE International Conference on Automatic Face and Gesture Recognition*, 1998.

[12] I. A. Kakadiaris and D. Metaxas. 3D human body model acquisition from multiple views. In *Proceedings of ICCV'95*, pages 618–623, Boston, MA, June 1995.

[13] A. Laurentini. The visual hull : A new tool for contour-based image understanding. In *Proc. 7th Scandinavian Conference on Image Analysis*, pages 993–1002, 1991.

[14] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Trans. PAMI*, 16(2):150–162, February 1994.

[15] 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.

[16] H. S. Sawhney and S. Ayer. Compact representations of videos through dominant and multiple motion estimation. *IEEE Trans. PAMI*, 18(8):814–830, 1996.

[17] H. Sidenbladh, M. J. Black, and D. Fleet. Stochastic tracking of 3D human fi gures using 2D image motion. In *Proceedings ECCV'00*, Dublin, Ireland, June 2000.

[18] R. Szeliski. Rapid octree construction from image sequences. *CVGIP : Image Understanding*, 58(1):23–32, July 1993.

[19] R. Szeliski. Image mosaicing for tele-reality applications. Technical Report CRL 94/2, Compaq Cambridge Research Laboratory, 1994.

[20] W. T. Vetterling and et. al. *Numerical Recipes in C*. Cambridge University Press, 1993.

[21] B. Vijayakumar, D. Kriegman, and J. Ponce. Structure and motion of curved 3D objects from monocular silhouettes. In *Proc. of CVPR'96*, pages 327–334, June 1996.

[22] K. Y. K. Wong and R. Cipolla. Structure and motion from silhouettes. In *Proc. of ICCV'01*, June 2001.