EURASIP Journal on Applied Signal Processing 2004:11, 1648–1662
© 2004 Hindawi Publishing Corporation

A Real-Time Model-Based Human Motion Tracking and Analysis for Human Computer Interface Systems

Chung-Lin Huang
Department of Electrical Engineering, National Tsing-Hua University, Hsin-Chu 30055, Taiwan
Email: clhuang@ee.nthu.edu.tw

Chia-Ying Chung
Department of Electrical Engineering, National Tsing-Hua University, Hsin-Chu 30055, Taiwan
Email: cychuang@zyxel.com.tw

Received 3 June 2002; Revised 10 October 2003

This paper introduces a real-time model-based human motion tracking and analysis method for human computer interface (HCI). This method tracks and analyzes the human motion from two orthogonal views without using any markers. The motion parameters are estimated by pattern matching between the extracted human silhouette and the human model. First, the human silhouette is extracted and then the body definition parameters (BDPs) can be obtained. Second, the body animation parameters (BAPs) are estimated by a hierarchical tritree overlapping searching algorithm. To verify the performance of our method, we demonstrate different human posture sequences and use hidden Markov model (HMM) for posture recognition testing.

Keywords and phrases: human computer interface system, real-time vision system, model-based human motion analysis, body definition parameters, body animation parameters.

1. INTRODUCTION

Human motion tracking and analysis has a lot of applications, such as surveillance systems and human computer interface (HCI) systems. A vision-based HCI system need to locate and understand the user's intention or action in real time by using the CCD camera input. Human motion is a highly complex articulated motion. The inherent nonrigidity of human motion coupled with the shape variation and self-occlusions make the detection and tracking of human motion a challenging research topic. This paper presents a framework for tracking and analyzing human motion with the following aspects: (a) real-time operation, (b) no markers on the human object, (c) near-unconstrained human motion, and (d) data coordination from two views.

There are two typical approaches to human motion analysis: model based and nonmodel based, depending on whether predefined shape models are used. In both approaches, the representation of the human body has been developed from stick figures [1, 2], 2D contour [3, 4], and 3D volumes [5, 6] with increasing complexity of the model. The stick figure representation is based on the observation that human motions of body parts result from the movement of the relative bones. The 2D contour is allied with the projection of 3D human body on 2D images. The 3D volumes, such as generalized cones, elliptical cylinders [7], spheres [5], and blobs [6] describe human model more precisely.

With no predefined shape models, heuristic assumptions, which impose constraints on feature correspondence and decreasing search space, are usually used to establish the correspondence of joints between successive frames. Moeslund and Granum [8] give an extensive survey of computer vision-based human motion capture. Most of the approaches are known as analysis by synthesis, and are used in a predict-match-update fashion. They begin with a predefined model, and predict a pose of the model corresponding to the next image. The predicted model is then synthesized to a certain abstraction level for the comparison with the image data. The abstract levels for comparing image data and synthesis data can be edges, silhouettes, contours, sticks, joints, blobs, texture, motion, and so forth. Another HCI system called "video avatar" [9] has been developed, which allows a real human actor to be transferred to another site and integrated with a virtual world.

One human motion tracking method [10] applied the Kalman filter, edge segment, and a motion model tuned to the walking image object by identifying the straight edges.

It can only track the restricted movement of walking human parallel to the image plane. Another real time system, Pfinder [11], starts with an initial model, and then refines the model as more information becomes available. The multiple human tracking algorithm W^4 [12, 13] has also been demonstrated to detect and analyze individuals as well as people moving in groups.

Tracking human motion from a single view suffers from occlusions and ambiguities. Tracking from more viewpoints can help solving these problems [14]. A 3D model-based multiview method [15] uses four orthogonal views to track unconstrained human movement. The approach measures the similarity between model view and actual scene based on arbitrary edge contour. Since the search space is 22 dimensions and the synthesis part uses the standard graph rendering to generate 3D model, their system can only operate in batch mode.

For an HCI system, we need a real-time operation not only to track the moving human object, but also to analyze the articulated movement as well. Spatiotemporal information has been exploited in some methods [16, 17] for detecting periodic motion in video sequences. They compute an autocorrelation measure of image sequences for tracking human motion. However, the periodic assumption does not fit the so-called unconstrained human motion. To speed up the human tracking process, a distributed computer vision systems [18] uses a model-based template matching to track the moving people at 15 frames/second.

Real-time body animation parameters (BAP) and body definition parameters (BDP) estimation is more difficult than the tracking-only process due to the large degrees of freedom of the articulated motion. Feature point corresponding has been used to estimate the motion parameters of the posture. In [19], an interesting approach for detecting and tracking human motion has been proposed, which calculates a best global labeling of point features using a learned triangular decomposition of the human body. Another real-time human posture estimation system [20] uses trinocular images and a simple 2D operation to find the significant points of human silhouette and reconstruct the 3D positions of human object from the corresponding significant points.

Hidden Markov model (HMM) has also been widely used to model the spatiotemporal property of human motion. For instance, it can be applied for recognizing model human dynamics [21], analyzing the human running and walking motions [22], discovering and segmenting the activities in video sequences [23], or encoding the temporal dynamics of the time-varying visual pattern [24]. The HMM approaches can be used to analyze some constrained human movements, such as human posture recognition or classification.

This paper presents a model-based real time system analyzing the near-unconstrained human motion video in real-time without using any markers. For a real-time system, we have to consider the tradeoff between computation complexity and system robustness. For a model-based system, there is also a tradeoff between the accuracy of representation and the number of parameters for the model that needs to be estimated. To compromise the complexity of model with the robustness of system, we use a simple 3D human model to analyze human motion rather than the conventional ones [2, 3, 4, 5, 6, 7].

Our system analyzes the object motion by extracting its silhouette and then estimating the BAPs. The BAPs estimation is formulated as a search problem that finds the motion parameters of the 2D human model of which its synthetic appearance is the most similar to the actual appearance, or silhouette, of the human object. The HCI system requires that a single human object interacts with the computer in a constrained environment (e.g., stationary background), which allows us to apply the background subtraction algorithm [12, 13] to extract the foreground object easily. The object extraction consists of (1) background model generation, (2) background subtraction and thresholding, and (3) morphology filtering.

Figure 1 illustrates the system flow diagram, which consists of four components including two viewers, one integrator, and one animator. Each viewer estimates the partial BDPs from the extracted foreground image and sends the results to the BDP integrator. The BDP integrator creates a universal 3D model by combining the information from these two viewers. In the beginning, the system needs to generate 3D BDP for different human objects. With the complete BDPs, each viewer may locate the exact position of the human object from its own view and then forward the data to the BAP integrator. The BAP integrator combines the two positions and calculates the complete 2D locations, which can be used to determine the BDP perspective scaling factors for two viewers. Finally, each viewer estimates the BAPs individually, which are combined as the final universal BAPs.

2. HUMAN MODEL GENERATION

The human model consists of 10 cylindrical primitives, representing torso, head, arms, and legs, which are connected by joints. There are ten connecting joints with different degrees of freedom. The dimensions of the cylinders (i.e., the BDPs of the human model) have to be determined for the BAP estimation process to find the motion parameters.

2.1. 3D Human model

The 3D human model consists of six 3D cylinders with elliptic cross-section (representing human torso, head, right upper leg, right lower leg, left upper leg, and left lower leg) and four 3D cylinders with circular cross-section (representing right upper arm, right lower arm, left upper arm, and left lower arm). Each cylinder with elliptic cross-section has three shape parameters including long radius, short radius, and height. A cylinder with circular cross-section has two shape parameters including radius and height. The post of the human body can be described in terms of the angles of the joints. For each joint of cylinder, there are up to three rotating angle parameters: θ_X, θ_Y, and θ_Z.

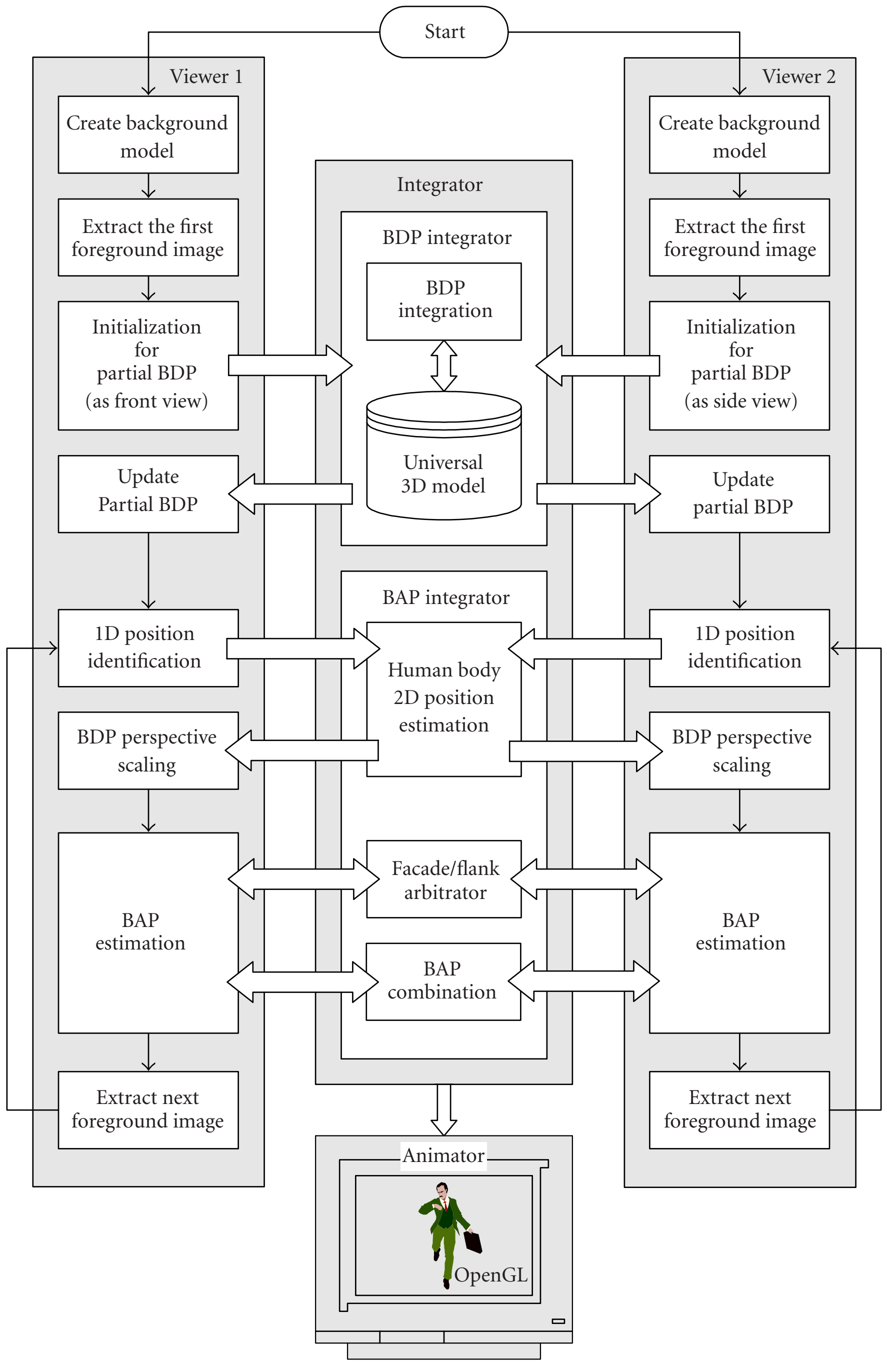

Start
Viewer 1
Create background model
Extract the first foreground image
Initialization for partial BDP (as front view)
Update Partial BDP
1D position identification
BDP perspective scaling
BAP estimation
Extract next foreground image
Viewer 2
Create background model
Extract the first foreground image
Initialization for partial BDP (as side view)
Update partial BDP
1D position identification
BDP perspective scaling
BAP estimation
Extract next foreground image
Integrator
BDP integrator
BDP integration
Universal 3D model
BAP integrator
Human body 2D position estimation
Facade/flank arbitrator
BAP combination
Animator
OpenGL

FIGURE 1: The flow diagram of our real-time system.

These 10 connecting joints are located at navel, neck, right shoulder, left shoulder, right elbow, left elbow, right hip, left hip, right knee, and left knee. The human joints are classified as either flexion or spherical. A flexion joint has only one degree of freedom (DOF) while a spherical one has three DOFs. The shoulder, hip, and navel joints are classified as spherical type, and the elbow and knee joints are classified as the flexion type. Totally, there are 22 DOFs for human model: six spherical joints and four flexion ones.

2.2. *Homogeneous coordinate transformation*

From the definition of the human model, we use a homogeneous coordinate system as shown in Figure 2. We define the basic rotation and translation operators such as $\mathbf{R}_x(\theta)$, $\mathbf{R}_y(\theta)$, and $\mathbf{R}_z(\theta)$ which denote the rotation around x-axis, y-axis, and z-axis with θ degrees, respectively, and $\mathbf{T}(l_x, l_y, l_z)$ which denotes the transition along x-, y-, and z-axis with l_x, l_y, and l_z. Using these operators, we can derive the transformation between two different coordinate systems as follows.

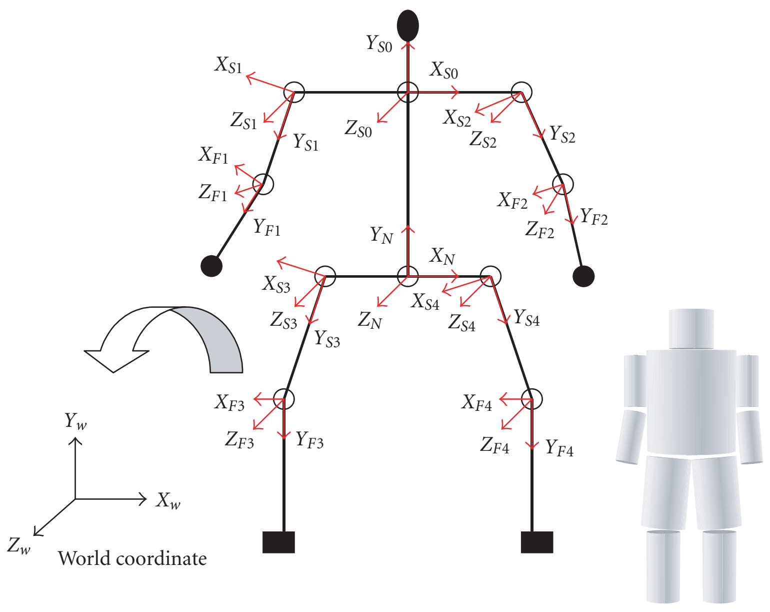


FIGURE 2: The homogeneous coordinate systems for the 3D human model.

(1) $M_W^N = R_y(\theta_y) \cdot R_x(\theta_x)$ depicts the transformation between the world coordinate (X_W, Y_W, Z_W) and the navel coordinate (X_N, Y_N, Z_N), where θ_x and θ_y represent the joint angles of the torso cylinder.

(2) $M_N^S = T(\ell_x, \ell_y, \ell_z) \cdot R_z(\theta_z) \cdot R_x(\theta_x) \cdot R_y(\theta_y)$ describes the transformation between the navel coordinate (X_N, Y_N, Z_N) and the spherical joints (such as neck, shoulder, and hip) coordinate (X_S, Y_S, Z_S), where θ_x, θ_y, and θ_z represent the joint angles of the limbs connected to torso and (l_x, l_y, l_z) represents the position of joints.

(3) $M_S^F = T(\ell_x, \ell_y, \ell_z) \cdot R_x(\theta_x)$ denotes the transformation between the spherical joint coordinate (X_S, Y_S, Z_S) and the flexion joints (such as elbow and knee) coordinate (X_F, Y_F, Z_F), where θ_x represents the joint angle of the limbs connected to the spherical joint, and (l_x, l_y, l_z) represents the position of joints.

2.3. Similarity measurement

The matching between the silhouette of human object and the synthesis image of the 3D model is to calculate the shape similarity measure. Similar to [3], we present an operator $S(I_1, I_2)$, which measures the shape similarity between two binary images I_1 and I_2 of the same dimension in interval $[0, 1]$. Our operator only considers the area difference between two shapes, that is, the ratio of positive error p (represents the ratio of the pixels in the image but not in the model to the total pixels of the image and model) and the negative error n (represents the ratio of the pixels in the model but not in the image to the total pixels of the image and model), which are calculated as

$$p = \frac{(I_1 \cap I_2^C)}{(I_1 \cup I_2)}, \qquad n = \frac{(I_2 \cap I_1^C)}{(I_1 \cup I_2)}, \tag{1}$$

where I^C denotes the complement of I. The similarity between two shapes I_1 and I_2 is the matching score defined as $S(I_1, I_2) = e^{-p-n}(1 - p)$.

2.4. BDPs determination

We assume that initially the human object stands straight up with his arms stretched as shown in Figure 3. The BDPs of the human model are illustrated in Table 1. The side viewer estimates the short radius of torso, whereas the front viewer determines the remaining parameters. The boundary of body, including x_{leftmost}, $x_{\text{rightmost}}$, y_{highest}, and y_{lowest}, is easily found, as shown in Figure 4.

The front viewer estimates all BDPs except the short radius of torso. There are three processes in the front viewer BDP determination: (a) torso-head-leg BDP determination, (b) arm BDP determination, and (c) fine tuning. Before the BDP estimation of the torso, head, and leg, we construct the vertical projection of the foreground image, that is, $P(x) = \int f(x, y)dy$, as shown in Figure 5. Then, we may find $\text{avg} = \int_{x_{\text{leftmost}}}^{x_{\text{rightmost}}} P(x)dx/(x_{\text{rightmost}} - x_{\text{leftmost}})$, where $P(x) \neq 0$ for $x_{\text{leftmost}} < x < x_{\text{rightmost}}$. To find the width of the torso, we scan $P(x)$ from left to right to find x_1, the smallest x value that makes $P(x_1) > \text{avg}$, and then scan $P(x)$ from right to left to find x_2, the largest x value that makes $P(x_2) > \text{avg}$

Table 1: The BDPs to be estimated, V indicates the existing BDP parameter.

Parameter	Limb					
	Torso	Head	Upper arm	Lower arm	Upper leg	Lower leg
Height	V	V	V	V	V	V
Radius	—	—	V	V	—	—
Long radius	V	V	—	—	V	V
Short radius	V	V	—	—	V	V

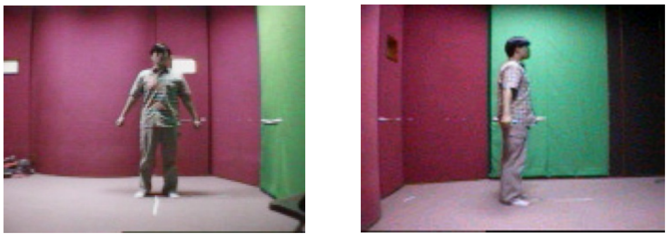


Figure 3: Initial posture of person: (a) the front viewer; (b) the side viewer.

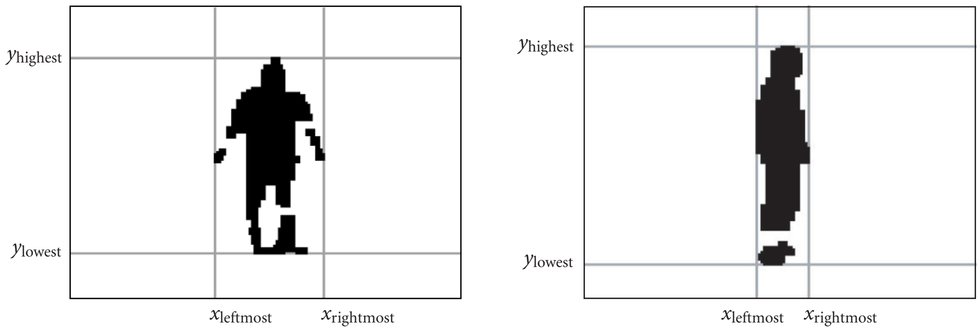


Figure 4: the BDPs estimation.

(see Figure 5). Therefore, we may define the center of body as $x_c = (x_1 + x_2)/2$, and the width of torso, $W_{\text{torso}} = x_2 - x_1$.

To find the other BDP parameters, we remove the head by applying morphological filtering operations, which consists of the morphological closing operation using a structure element (size $0.8W_{\text{torso}} \times 1$), and the morphological opening operation by the same element (as shown in Figure 6). Then we may extract the location of shoulder in y-axis (y_h) by scanning the image (i.e., Figure 6b) horizontally from top to bottom in the image without head, and define the length of head: $\text{len}_{\text{head}} = y_{\text{highest}} - y_h$. Here, we assume the ratio of length of the torso and the leg is 4 : 6, and define the length of torso as $\text{len}_{\text{torso}} = 0.4(y_h - y_{\text{lowest}})$; the length of upper leg as $\text{len}_{\text{up-leg}} = 0.5\times 0.6(y_h - y_{\text{lowest}})$, and the length of lower leg as $\text{len}_{\text{low-leg}} = \text{len}_{\text{up-leg}}$. Finally, we may estimate the center of body in y-axis as $y_c = y_h - \text{len}_{\text{torso}}$; the long radius of torso as $\text{LR}_{\text{torso}} = W_{\text{torso}}/2$; the long radius of head as $0.2W_{\text{torso}}$; the short radius of head as $0.16W_{\text{torso}}$; the long radius of leg as $0.2W_{\text{torso}}$; and the short radius of leg as $0.36W_{\text{torso}}$.

Before identifying the radius and length of arm, the system extracts the extreme position of arms, $(x_{\text{leftmost}}, y_l)$ and $(x_{\text{rightmost}}, y_r)$ (as shown in Figure 7), and then defines the position of shoulder joints, $(x_{\text{right-shoulder}}, y_{\text{right-shoulder}}) = (x_a, y_a) = (x_c - \text{LR}_{\text{torso}}, y_c - \text{len}_{\text{torso}} + 0.45\,\text{LR}_{\text{torso}})$. From the extreme position of arms and position of shoulder joints, we calculate the length of upper arm ($\text{len}_{\text{upper-arm}}$) and lower arm ($\text{len}_{\text{lower-arm}}$), and the rotating angles around z-axis of the shoulder joints (θ_z^{arm}). These three parameters are defined as follows: (a) $\text{len}_{\text{arm}} = \sqrt{(x_b - x_a)^2 + (y_b - y_a)^2}$; (b) $\theta_z^{\text{arm}} = \arctan(|x_b - x_a|/|y_b - y_a|)$; (c) $\text{len}_{\text{upper-arm}} = \text{len}_{\text{lower-arm}} = \text{len}_{\text{arm}}/2$. Finally, we fine-tune the long radius of torso, the radius of arms, the rotating angles around the z-axis of the shoulder joints, and the length of arms.

To find the short radius of torso, the side viewer constructs the vertical projection of the foreground image, that is, $P(x) = \int f(x, y)dy$, and $\text{avg} = \int_{x_{\text{leftmost}}}^{x_{\text{rightmost}}} P(x)dx/(x_{\text{rightmost}} - x_{\text{leftmost}})$, where $P(x) \neq 0$ for $x_{\text{leftmost}} < x < x_{\text{rightmost}}$. Scanning $P(x)$ from left to right, we may find x_1, the smallest x

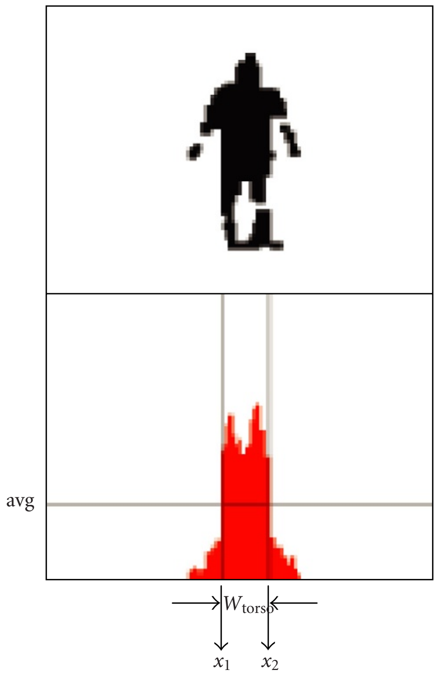


FIGURE 5: Foreground image silhouette and its vertical projection.

value, with $P(x_1) > \text{avg}$, and then scanning $P(x)$ from right to left, we may also find x_2, the largest x value, with $P(x_2) > \text{avg}$. Finally, the short radius of torso is defined as $(x_2 - x_1)/2$.

3. MOTION PARAMETERS ESTIMATION

There are 25 motion parameters (22 angular parameters and 3 position parameters) for describing human body motion. Here, we assume that three rotation angles of head and two rotation angles of torso (rotation angle around X-axis and Z-axis) are fixed. The real-time tracking and motion estimation consists of four stages: (1) facade/flank determination, (2) Human position estimation, (3) arm joint angle estimation, and (4) leg joint angle estimation. In each stage, only the specific parameters are determined based on the matching between the model and the extracted object silhouette.

3.1. Facade/flank determination

First, we find the rotation angle of torso around the y-axis of the world coordinate ($\theta_{Y_W}^T$). A y-projection of the foreground object image is constructed without the lower portion of the body, that is, $P(x) = \int_{y_{\text{hip}}}^{y_{\max}} f(x, y)dy$, as shown in Figure 8. Each viewer finds the corresponding parameters independently. Here, we define the hips' position along y-axis as $y_{\text{hip}} = (y_c + 0.2 \cdot \text{height}_{\text{torso}}) \cdot r_{t,n}$, where y_c is the center of body in y-axis, $\text{height}_{\text{torso}}$ is the height of torso, and $r_{t,n}$ is the perspective scaling factor of viewer n ($n = 1$ or 2), which will be introduced in Section 4.2. Then, each viewer scans $P(x)$ from left to right to find x_1, the least x, where $P(x_1) > \text{height}_{\text{torso}}$, and then scans $P(x)$ from right to left to find x_2, the largest x, where $P(x_2) > \text{height}_{\text{torso}}$. The width of the upper body is $W_{\text{u-body},n} = |x_2 - x_1|$, where $n = 1$ or 2 is the number of the viewer. Here, we define two thresholds for each viewer to determine whether the foreground object is a facade view or a flank view: $th_{\text{low},n}$ and $th_{\text{high},n}$, where $n = 1$ or 2 is the number of the viewer. In viewer n ($n = 1$ or 2), if $W_{\text{u-body},n}$ is smaller than $th_{\text{low},n}$, it is a flank view; if $W_{\text{u-body},n}$ is greater than $th_{\text{high},n}$, it is a facade view; otherwise, it remains unchanged.

3.2. Object tracking

The object tracking determines the position, (X_W^T, Y_W^T, Z_W^T), of human object. We may simplify the perspective projection as a combination of the perspective scaling factor and the orthographic projection. The perspective scaling factor values are calculated (in Section 4.2) by new position X_W^T and Z_W^T. Given a scaling factor and BDPs, we generate a 2D model image. With the extracted object silhouette, we shift the 2D model image along X-axis in image coordinate and search for the real X_W^T (or Z_W^T in viewer 2) that generates the best matching score, as shown in Figure 9a.

The estimated X_W^T and Z_W^T are then used to update the perspective scaling factor for the other viewer. Similarly, we shift the silhouette along Y-axis in image coordinate to find Y_W^T that generates the best matching score (see Figure 9b). In each matching process, the possible position difference between the silhouette and the model are -5, -2, -1, $+1$, $+2$, and $+5$. Finally, the positions X_W^T and Z_W^T are combined as the 2D position values and a new perspective scaling factor can be calculated for the tracking process in the next time instance.

3.3. Arm joint angle estimation

The arm joint has 2 DOFs, and it can bend on certain 2D planes. In a facade view, we assume that the rotation angles of shoulder joint around X-axis of the navel coordinate ($\theta_{X_N}^{\text{RUA}}$ and $\theta_{X_N}^{\text{LUA}}$) are fixed and then we may estimate the others including $\theta_{Z_N}^{\text{RUA}}$, $\theta_{Y_N}^{\text{RUA}}$, $\theta_{X_{RS}}^{\text{RLA}}$, $\theta_{Z_N}^{\text{LUA}}$, $\theta_{Y_N}^{LUA}$, and $\theta_{X_{LS}}^{\text{LLA}}$, where RUA depicts the right upper arm, LUA depicts the left upper arm, RLA depicts the right lower arm, LLA depicts the left lower arm, N depicts the navel coordinate system, RS depicts the right shoulder coordinate system, and LS depicts the left shoulder coordinate system.

In a facade view, the range of $\theta_{Z_N}^{\text{RUA}}$ is limited in $[0, 180^\circ]$, while $\theta_{Z_N}^{\text{LUA}}$ is limited in $[180^\circ, 360^\circ]$, and the values of $\theta_{Y_N}^{\text{RUA}}$ and $\theta_{Y_N}^{\text{LUA}}$ are either 90° or -90°. Different from [15], the range of $\theta_{X_{RS}}^{\text{RLA}}$ (or $\theta_{X_{LS}}^{\text{LLA}}$) relies on the value of $\theta_{Z_N}^{\text{RUA}}$ (or $\theta_{Z_N}^{\text{LUA}}$) to prevent the occlusion between the lower arms and the torso. In a flank view, the range of $\theta_{X_N}^{\text{RUA}}$ and $\theta_{X_N}^{\text{LUA}}$ is limited in $[-180^\circ, 180^\circ]$. Here, we develop an overlapped tritree search method, see Section 3.5, to reduce the search time and expand the search range. In a facade view, there are 3 DOFs for each arm joint, whereas in a flank view, there are 1 DOF for each arm joint. In a facade view, the right arm joint angle estimation is illustrated in the following steps.

(1) Determine the rotation angle of the right shoulder around the Z-axis of the navel coordinate ($\theta_{Z_N}^{\text{RUA}}$) by applying our overlapped tritree search method and choose the value where the corresponding matching score is the highest (see Figure 10a).

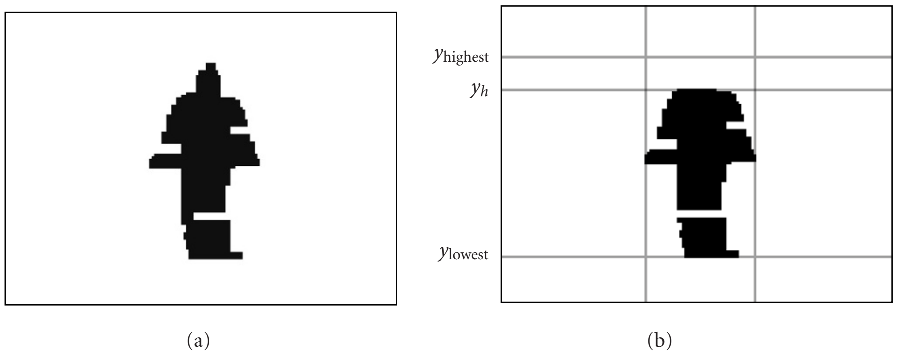


FIGURE 6: The head-removed image. (a) Result of closing. (b) Result of opening.

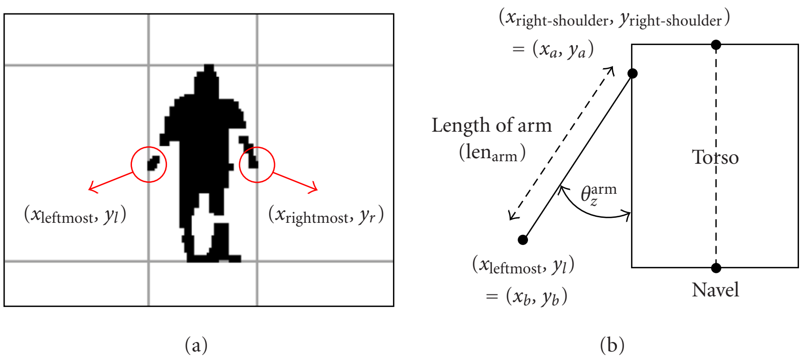

$(x_{\text{leftmost}}, y_l)$
$(x_{\text{rightmost}}, y_r)$
$(x_{\text{right-shoulder}}, y_{\text{right-shoulder}})$
$= (x_a, y_a)$
Length of arm
$(\text{len}_{\text{arm}})$
θ_z^{arm}
Torso
$(x_{\text{leftmost}}, y_l)$
$= (x_b, y_b)$
Navel
(a)
(b)

FIGURE 7: (a) The extreme position of arms. (b) The radius and length of arm.

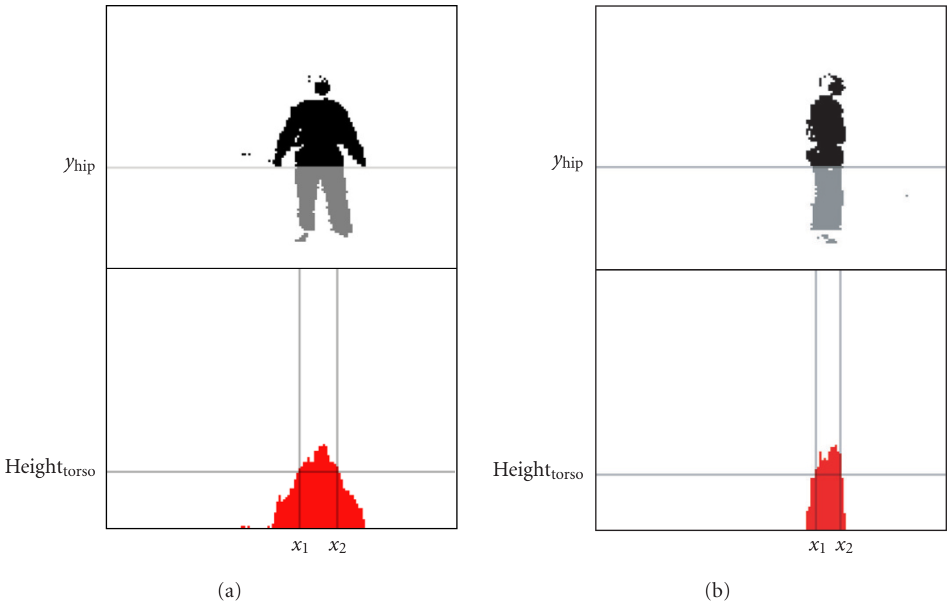


FIGURE 8: Facade/flank determination. (a) Facade. (b) Flank.

(2) Define the range of the rotation angle of the right elbow joint around x-axis in the right shoulder coordinate system ($\theta_{X_{RS}}^{\text{RLA}}$). It relies on the value of $\theta_{Z_N}^{\text{RUA}}$ to prevent the occlusion between the lower arm and the torso. First, we define a threshold th_a: if $\theta_{Z_N}^{\text{RUA}} > 110°$, then $th_a = 2 \cdot (180° - \theta_{Z_N}^{\text{RUA}})$, or else $th_a = 140°$.

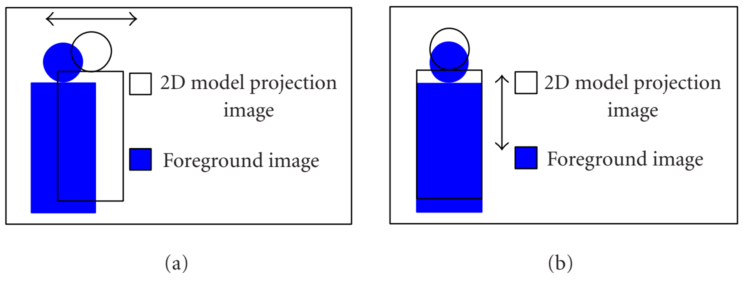


FIGURE 9: Shift the 2D model image along (a) X-axis and (b) Y-axis.

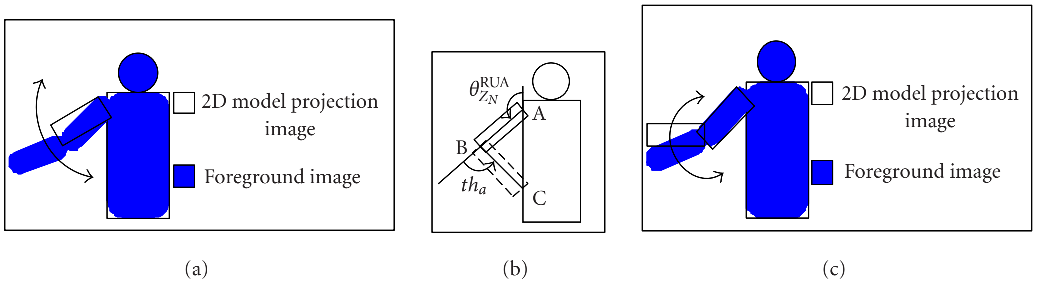

☐ 2D model projection image
■ Foreground image
(a)
$\theta^{\mathrm{RUA}}_{Z_N}$
A
B
C
th_a
(b)
☐ 2D model projection image
■ Foreground image
(c)

FIGURE 10: (a) Rotate upper arm along Z_N-axis. (b) The definition of th_a. (c) Rotate lower arm along X_{RS}-axis.

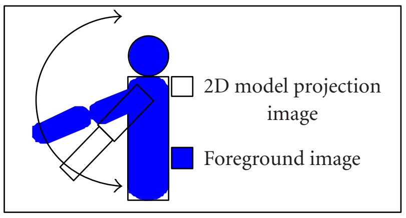


FIGURE 11: Rotate the arm along X_N-axis.

So, $\theta^{\mathrm{RLA}}_{X_{RS}} \in [-th_a, 140^\circ]$ for $\theta^{\mathrm{RUA}}_{Y_N} = 90^\circ$, and $\theta^{\mathrm{RLA}}_{X_{RS}} \in [-140^\circ, th_a]$ for $\theta^{\mathrm{RUA}}_{Y_N} = -90^\circ$. From $\triangle ABC$ shown in Figure 10b, we find $\overline{AB} = \overline{BC}$, $\angle BAC = \angle BCA = 180^\circ - \theta^{\mathrm{RUA}}_{Z_N}$, and $th_a = \angle BAC + \angle BCA = 2 \cdot (180^\circ - \theta^{\mathrm{RUA}}_{Z_N})$.

(3) Determine the rotation angle of the right elbow joint around x-axis in the right shoulder coordinate system ($\theta^{\mathrm{RLA}}_{X_{RS}}$) by applying the overlapped tritree search method and choose the value where the corresponding matching score is the highest (see Figure 10c).

Similarly, in the flank view, the arm joint angle estimation determines the rotation angle of shoulder around the X-axis of the navel coordinate ($\theta^{RUA}_{X_N}$) (see Figure 11).

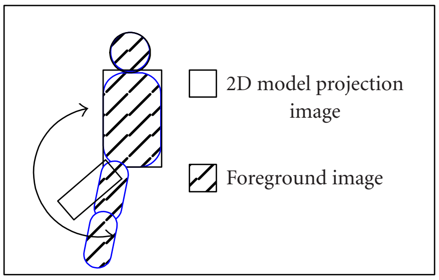
3.4. Leg joint angle estimation

The estimation processes for the joint angle of the legs in a facade view and a flank view are different. In a facade view, there are two cases depending on whether knees are bent or not. To decide which case, we check the location of navel in y-axis to see whether it is less than that of the initial posture or not. If yes, then the human is squatting down, else he is standing. For the standing case, we only estimate the rotation angles of hip joints around Z_N-axis in navel coordinate system (i.e., $\theta^{\mathrm{RUL}}_{Z_N}$ and $\theta^{\mathrm{LUL}}_{Z_N}$). As shown in Figure 12a, we estimate $\theta^{\mathrm{RUL}}_{Z_N}$ by applying the overlapped tritree search method.

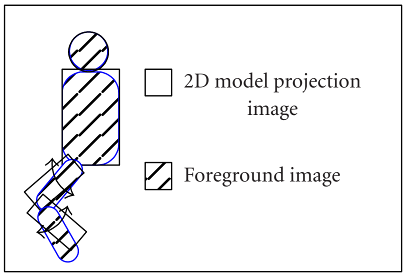
In squatting down case, we also estimate the rotation angles of hip joints around Z_N-axis in navel coordinate system ($\theta^{\mathrm{RUL}}_{Z_N}$ and $\theta^{\mathrm{LUL}}_{Z_N}$). After that, the rotation angles of the hip joints around X_N-axis in the navel coordinate system ($\theta^{\mathrm{RUL}}_{X_N}$ and $\theta^{\mathrm{LUL}}_{X_N}$) and the rotation angles of the knee joints around x_H-axis in the hip coordinate system ($\theta^{\mathrm{RLL}}_{X_{RH}}$ and $\theta^{LLL}_{X_{LH}}$) are estimated. Because the foot is right beneath the torso, $\theta^{\mathrm{RLL}}_{X_{RH}}$ (or $\theta^{\mathrm{LLL}}_{X_{LH}}$) can be defined as $\theta^{\mathrm{RLL}}_{X_{RH}} = -2\theta^{\mathrm{RUL}}_{X_N}$ (or $\theta^{\mathrm{LLL}}_{X_{LH}} = -2\theta^{\mathrm{LUL}}_{X_N}$). From $\triangle ABC$ in Figure 12c, we find $\overline{AB} = \overline{BC}$, $\angle BAC = \angle BCA = \theta^{\mathrm{RUL}}_{X_N}$, and $\theta^{\mathrm{RLL}}_{X_{RH}} = -(\angle BAC + \angle BCA)$. The range of $\theta^{\mathrm{RUL}}_{X_N}$ and $\theta^{\mathrm{LUL}}_{X_N}$ is $[0, 50^\circ]$. Take the right leg as an example, $\theta^{\mathrm{RUL}}_{X_N}$ and $\theta^{\mathrm{RLL}}_{X_{RH}}$ are estimated by applying a search method only for $\theta^{\mathrm{RUL}}_{X_N}$ with $\theta^{\mathrm{RLL}}_{X_{RH}} = -2\theta^{\mathrm{RUL}}_{X_N}$ (e.g., Figure 12b). In flank view, we estimate the rotation angles of the hip joints around x_N-axis of the navel coordinate ($\theta^{\mathrm{RUL}}_{X_N}$ and $\theta^{\mathrm{LUL}}_{X_N}$) and the rotation angles of the knee joints around X_H-axis of the hip coordinates ($\theta^{\mathrm{RLL}}_{X_{RH}}$ and $\theta^{\mathrm{LLL}}_{X_{LH}}$).

3.5. Overlapped tritree hierarchical search algorithm

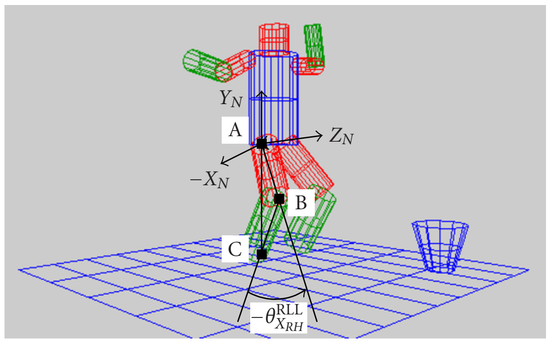
The basic concept of BAPs estimation is to find the highest matching score between the 2D model and the silhouette. However, since the search space depends on the motion activity and the frame rate of input image sequence, the faster the articulated motion is, the larger the search space



(a)



(b)



(c)

Figure 12: Leg joints angular values estimation in facade view. (a) Rotate upper leg along Z_N-axis. (b) Determine $\theta_{X_N}^{\text{RUL}}$ and $\theta_{X_{RH}}^{\text{RLL}}$. (c) The definition of $\theta_{X_{RH}}^{\text{RLL}}$.

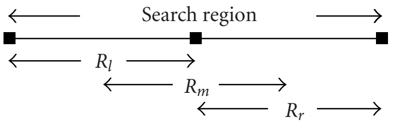


Figure 13: The search region is divided into three overlapped subregions.

will be. Instead of using the sequential search in the specific search space, we apply the hierarchical search. As shown in Figure 13, we divide the search space into three overlapped regions (left region (R_l), middle region (R_m), and right region (R_r)) and select one search angle for each region. From the three search angles, we do three different matches, and find the best match of which the corresponding region is the winner region. Then we update the next search region by the current winner region recursively until the width of the current search region is smaller than the step-to-stop criterion value. During the hierarchical search, we will update the winner angle if the current matching score is the highest. After reaching to the leaf of the tree, we assign the winner angle as the specific BAP.

We divide the initial search region R into three overlapped regions as $R = R_l + R_m + R_r$, select the step-to-stop criterion value Θ, and do the overlapped tritree searching as follows.

(1) Let n indicate the current iteration index and initialize the absolute winning score as $S_{\text{WIN}} = 0$.

(2) Set $\theta_{l,n}$ as the left extreme of the current search region $R_{l,n}$, $\theta_{m,n}$ as the center of the current search region $R_{m,n}$, and $\theta_{r,n}$ as the right extreme of the current search region $R_{r,n}$, and calculate the matching score corresponding to the right region as $S(R_{l,n}, \theta_{l,n})$, the middle region as $S(R_{m,n}, \theta_{m,n})$, and the left region as $S(R_{r,n}, \theta_{r,n})$.

(3) If $\text{Max}\{S(R_{l,n}, \theta_{l,n}), S(R_{m,n}, \theta_{m,n}), S(R_{r,n}, \theta_{r,n})\} < S_{\text{WIN}}$, go to step (5), else $S_{\text{win}} = \text{Max}\{S(R_{l,n}, \theta_{l,n}), S(R_{m,n}, \theta_{m,n}), S(R_{r,n}, \theta_{r,n})\}$, $\theta_{\text{win}} = \theta_{x,n}|_{S_{\text{win}}=S(R_{x,n},\theta_{x,n}), x\in\{r,m,l\}}$, $R_{\text{win}} = R_{x,n}|_{S_{\text{win}}=S(R_{x,n},\theta_{x,n}), x\in\{r,m,l\}}$.

(4) If $n = 1$, then $\theta_{\text{WIN}} = \theta_{\text{win}}$ and $S_{\text{WIN}} = S_{\text{win}}$, else if the current winner matching score is larger than the absolute winner matching score, $S_{\text{win}} > S_{\text{WIN}}$, then $\theta_{\text{WIN}} = \theta_{\text{win}}$ and $S_{\text{WIN}} = S_{\text{win}}$.

(5) Check the width of R_{win}, if $|R_{\text{win}}| > \Theta$, then continue, else stop.

(6) Divide R_{win} into another three overlapped subregions: $R_{\text{win}} = R_{l,n+1} + R_{m,n+1} + R_{r,n+1}$ for the next iteration $n + 1$, and go to step (2).

On each stage, we may move the center of search region according to the range of joint angular value and the previous θ_{win}, for example, when the range of arm joints is defined as $[0, 180]$ and the current search region's width is defined as$|R_{\text{arm-j}}| = 64$. If the θ_{win} in the previous stage is 172, the center of $R_{\text{arm-j}}$ will be moved to 148 ($180 - 64/2 = 148$) and $R_{\text{arm-j}} = [116, 180]$, so that the right boundary of $R_{\text{arm-j}}$ is inside the range $[0, 180]$. If θ_{win} of the previous angle is 100, the center of $R_{\text{arm-j}}$ is unchanged, $R_{\text{arm-j}} = [68, 132]$, because the search region is inside the range of angular variation of the arm joint.

In each stage, the tritree search process compares the three matches and finds the best one. However, in real implementation, it requires less matching because some matching operations in current stage had been calculated in the previous stage. When the winner region in previous stage is the right or left region, we only have to calculate the matches using the middle point of current search region, and when the winner region in previous stage is the middle region, we have to calculate the matches using the left extreme and the right extreme of the current search region.

Here we assume that the winning probabilities of the left, middle, or right region are equiprobable. The number of matching of the first stage is 3 and the average number of matching in other stages $T_{2,\text{avg}} = 2 \times (1/3) + 1 \times (2/3) = 4/3$. The average number of matching is

$$T_{\text{avg}} = 3 + T_{2,\text{avg}} \cdot \left(\log_2\left(W_{\text{init}}\right) - \log_2\left(W_{\text{sts}}\right) - 1\right), \qquad (2)$$

where W_{init} is the width of the initial search region and W_{sts} is the final width for the step to stop. The average number of matching for the arm joint is $3 + 4/3 * (6 - 2 - 1) = 7$ because $W_{\text{init}} = 64$ and $W_{\text{sts}} = 4$. The average number of matching operations for estimating the leg joint is $5.67(3 + 4/3 * (5 - 2 - 1))$ because $W_{\text{init}} = 32$ and $W_{\text{sts}} = 4$. The worst case for the arm joint estimation is $3 + 2 * (6 - 2 - 1) = 9$ matching (or $3+2*(5-2-1) = 7$ matching for the leg joint), which is better than the full search method which requires 17 matching for the arm joint estimation and 9 matching for the leg joint estimation.

4. THE INTEGRATION AND ARBITRATION OF TWO VIEWERS

The information integration consists of camera calibration, 2D position and perspective scaling determination, facade/flank arbitration, and BAP integration.

4.1. Camera calibration

The viewing directions of two cameras are orthogonal. We define the center of action region as the origin in the world coordinate and we assume that the position of these two cameras are fixed at (X_{c1}, Y_{c1}, Z_{c1}) and (X_{c2}, Y_{c2}, Z_{c2}). The viewing directions of these two cameras are parallel to z-axis and x-axis. Here we let $(X_{c1}, Y_{c1}) \approx (0, 0)$ and $(Y_{c2}, Z_{c2}) \approx (0, 0)$. The viewing direction of camera 1 points to the negative Z direction, while that of camera 2 points to the positive X direction. The camera is initially calibrated by the following steps.

(1) Fix the positions of camera 1 and camera 2 on the z-axis and x-axis.

(2) Put two sets of line markers on the scene (ML_{zg} and ML_{zw} as well as ML_{xg} and ML_{xw}, as shown in Figure 14). The first two line markers are projection of Z-axis onto the ground and the left-hand side wall. The second two line markers are the projection of X-axis onto the ground and the background wall.

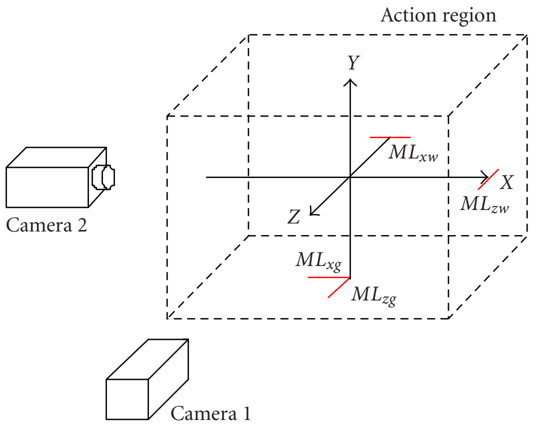


FIGURE 14: The line marker for camera calibration.

(3) Adjust the viewing direction of camera 1 until the line marker ML_{zg} overlaps the line $x = 80$ and the line $x = 81$; the line marker ML_{xw} overlaps the line $y = 60$ and the line $y = 61$.

(4) Adjust the viewing direction of camera 2 until the line mark ML_{xg} overlaps the line $x = 80$ and the line $x = 81$; the line marker ML_{zw} overlaps the line $y = 60$ and the line $y = 61$.

The camera parameters include the focal lengths and the positions of the two cameras. First we assume that there are three rigid objects located at the positions $\mathbf{A} = (0, 0, 0)$, $\mathbf{B} = (0, 0, D_Z)$, and $\mathbf{C} = (D_X, 0, 0)$ in the world coordinate, where D_X and D_Z are known. Therefore, the pinnacles of three rigid objects are located at positions $\mathbf{A}'$, $\mathbf{B}'$, and $\mathbf{C}'$, where the $\mathbf{A}' = (0, T, 0)$, $\mathbf{B}' = (0, T, D_Z)$, and $\mathbf{C}' = (D_X, T, 0)$ in the world coordinate. The pinnacles of the three rigid objects are projected at (x_{1A}, t_{1A}), (x_{1B}, t_{1B}), and (x_{1C}, t_{1C}) in the image frame of camera 1, and (z_{2A}, t_{2A}), (z_{2B}, t_{2B}), and (z_{2C}, t_{2C}) in the image frame of camera 2, respectively.

We assume λ_1 is the focal length of camera 1, and $(0, 0, Z_{c1})$ is its location. By applying the triangular geometry calculation on perspective projection images, we have $\lambda_1 = Z_{c1}(x_{1c} - x_{1A})/Dz$. Similarly, let λ_2 the focal length and $(X_{c2}, 0, 0)$ the location of camera 2, and we have $\lambda_2 = -X_{c2}(z_{2B} - z_{2A})/Dz$.

4.2. Perspective scaling factor determination

The location of the object is (X_W^T, Y_W^T, Z_W^T) in the world coordinate, of which the X_W^T and Z_W^T can be obtained from two viewers. Here, we need to find the depth information and calculate the perspective scaling factors of these two viewers. Here, we assume that the location of the object changes from $\mathbf{A} = (0, 0, 0)$ to $\mathbf{D} = (D_X', 0, D_Z')$, $X_{c1} \approx 0$, and $Z_{c2} \approx 0$. The pinnacle of the object moves from $\mathbf{A}' = (0, T, 0)$ to $\mathbf{D}' = (D_X', T', D_Z')$. The ratio T'/T is not a usable parameter because it is depth dependent and there is a great possibility that human object may be squatting down. The pinnacles of the previous and current objects are projected as (x_{1A}', t_{1A}') and (x_{1D}', t_{1D}') in camera 1, and as (z_{2A}', t_{2A}') and (z_{2D}', t_{2D}') in camera 2. The heights, t_{1D}' and t_{2D}', are unknown since they are depth dependent, however, the locations, x_{1D}' and z_{2D}' are approximated as $x_{1D}' \approx X_W{}^T$ and $z_{2D}' \approx Z_W{}^T$. The perspective scaling factors of human model in two viewers (i.e., r_{t1}' and r_{t2}') are different, where $r_{t1}' = |t_{1D}'/t_{1A}'|$ and $r_{t2}' = |t_{2D}'/t_{2A}'|$. Given x_{1A}', t_{1A}', z_{2A}', t_{2A}', x_{1D}', and z_{2D}', we may find D_X' and D_Z' as

$$\begin{aligned} D'_X &= \frac{Z_{c1}\lambda_2 + z'_{2D}x_{c2}}{\lambda_1\lambda_2/x'_{1D} + z'_{2D}}, \\ D'_Z &= \frac{x'_{1D}Z_{c1} - X_{c2}\lambda_1}{\lambda_1\lambda_2/z'_{2D} + x'_{1D}}, \end{aligned} \tag{3}$$

and then find the perspective scaling factor r_{t1}' and r_{t2}' as

$$\begin{aligned} r'_{t1} &= \left|\frac{t'_{1D}}{t'_{1A}}\right| = \frac{Z_{c1}\cdot\sqrt{\lambda_1{}^2 + x'_{1D}{}^2}}{\sqrt{\lambda_1{}^2 + x'_{1A}{}^2}\cdot\sqrt{\left(Z_{c1} - D'_Z\right)^2 + D'_X{}^2}}, \\ r'_{t2} &= \left|\frac{t'_{2D}}{t'_{2A}}\right| = \frac{-X_{c2}\cdot\sqrt{\lambda_2{}^2 + z'_{2D}{}^2}}{\sqrt{\lambda_2{}^2 + z'_{2A}{}^2}\cdot\sqrt{\left(D'_X - X_{c2}\right)^2 + D'_Z{}^2}}. \end{aligned} \tag{4}$$

The highest pixel of the silhouette is treated as the top of the object and each position of the silhouette object is approximated to be that of the human object. Using perspective scaling factor, we may scale our human model for the following BAP estimation process.

The side viewer estimates the short radius of torso, while the front viewer finds the remaining parameters. During initialization, the height of human object is t_1 in viewer 1 and t_2 in viewer 2, so the scaling factor between the viewers is $r_t = t_2/t_1$. Therefore, the BDPs of human models for viewer 1 and viewer 2 can be easily scaled. Because the universal BDPs are defined in the scaling factor of viewer 1, we define the short radius of torso in universal BDPs as $SR_{\text{torso},u} = SR_{\text{torso},2}/r_t$, where $SR_{\text{torso},2}$ is the short radius of torso in viewer 2 and the remaining parameters in universal BDPs are defined directly as those in viewer 1.

4.3. Facade/flank arbitrator

The facade/flank arbitrator combines the results of facade/flank transition processes of the two viewers. Initially, viewer 1 is the front viewer and captures the facade view of the object, whereas viewer 2 is the side viewer and captures the flank view of the object. Then, when either viewer 1 or viewer 2 changes their own facade/flank transitions, then they will ask the facade/flank arbitrator for coordination. If any one of the following transitions occurs, the facade/flank arbitrator will perform the corresponding coordination as follows.

(1) When the object in viewer 1 changes from flank to facade (i.e., $w_{\text{u-body},1} > th_{\text{high},1}$) and the same object in viewer 2 stays as facade (i.e., $w_{\text{u-body},2} \geq th_{\text{low},2}$), the arbitrator checks as follows: *if* $|w_{u\text{-}body,1} - th_{high,1}| > |w_{u\text{-}body,2} - th_{low}, 2|$, *then sets the object in viewer 2 to flank, else changes the object in viewer 1 back to flank.*

(2) When the object in viewer 1 changes from facade to flank (i.e., $w_{\text{u-body},1} < th_{\text{low},1}$) and the same object in viewer 2 stays as flank (i.e., $w_{\text{u-body},2} \leq th_{\text{high},2}$), the arbitrator checks as follows: *if* $|w_{u\text{-}body,1} - th_{low,1}| > |w_{u\text{-}body,2} - th_{high,2}|$, *then sets the object in viewer 2 to facade, else changes the object in viewer 1 back to facade.*

(3) When the object in viewer 1 remains as facade (i.e., $w_{\text{u-body},1} \geq th_{\text{low},1}$) and the same object in viewer 2 changes from flank to facade (i.e., $w_{\text{u-body},2} > th_{\text{high},2}$), the arbitrator checks as follows: *if* $|w_{u\text{-}body,1} - th_{low,1}| \geq |w_{u\text{-}body,2} - th_{high,2}|$, *then sets the object in viewer 2 back to flank, else changes the object in viewer 1 to flank.*

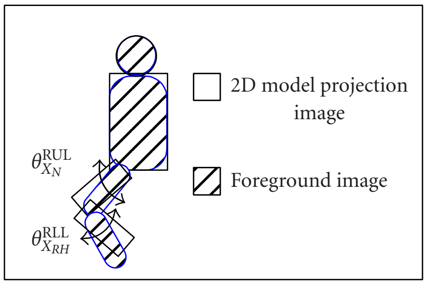
(4) When the object in viewer 1 stays as flank (i.e., $w_{\text{u-body},1} \leq th_{\text{high},1}$) and the same object in viewer 2 changes from facade to flank (i.e., $w_{\text{u-body},2} < th_{\text{low},2}$), the arbitrator checks as follows: *if* $|w_{u\text{-}body,1} - th_{high,1}| \geq |w_{u\text{-}body,2} - th_{low,2}|$, *then sets the object in viewer 2 back to facade, else changes the object in viewer 1 to facade.*

4.4. Body animation parameter integration

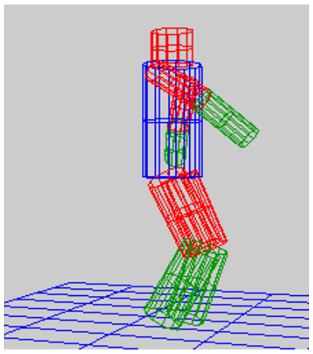
Two different sets of BAPs have been estimated by the two viewers. There are three major estimation processes for BAPs: human position estimation, arm joint angle estimation, and leg joint angle estimation. The BAP integration combines the BAPs from two different views into universal BAPs. First, in human position estimation, viewer 1 estimates X_W^T and Y_W^T, while viewer 2 estimates Z_W^T and Y_W^T. However, Y_W^T estimated by two viewers may be different. With more shape information of the object, Y_W^T estimated by the facade viewer is more robust. Second, the BAPs of the joints of arms are analyzed in two views. The flank viewer only estimates the rotation angles of shoulder joints around X_N-axis of the navel coordinate (i.e., $\theta_{X_N}^{\text{RUA}}$ and $\theta_{X_N}^{\text{LUA}}$); whereas the facade viewer estimates the other BAPs of arms including the rotation angles of shoulder joints around Y_N-axis and Z_N-axis of the navel coordinate (i.e., $\theta_{Y_N}^{\text{RUA}}$, $\theta_{Z_N}^{\text{RUA}}$, $\theta_{Y_N}^{\text{LUA}}$, and $\theta_{Z_N}^{\text{LUA}}$) and the rotation angles of elbow joints around X_N-axis of shoulder coordinates (i.e., $\theta_{X_N}^{\text{RLA}}$ and $\theta_{X_N}^{\text{LLA}}$). BAPs estimation processes of the two viewers are integrated as the universal BAPs.

Different from the integration of the arm BAPs, the estimated joint angles of leg of different viewers are related. Both viewers jointly estimate $\theta_{X_N}^{\text{RUL}}$, $\theta_{X_{RH}}^{\text{RLL}}$, $\theta_{X_N}^{\text{LUL}}$, and $\theta_{X_{RH}}^{\text{LLL}}$. For example, in **Figure 15**, the facade viewer analyzes these angles by assuming that the human is squatting down (see Figures 15a and 15b); whereas the flank viewer estimates these angles by assuming that the human is lifting his legs (see Figures 15c and 15d). Therefore, we determine whether the human is squatting down or lifting his leg from $\theta_{Z_N}^{\text{RUL}}$ and $\theta_{X_{RH}}^{\text{RLL}}$.

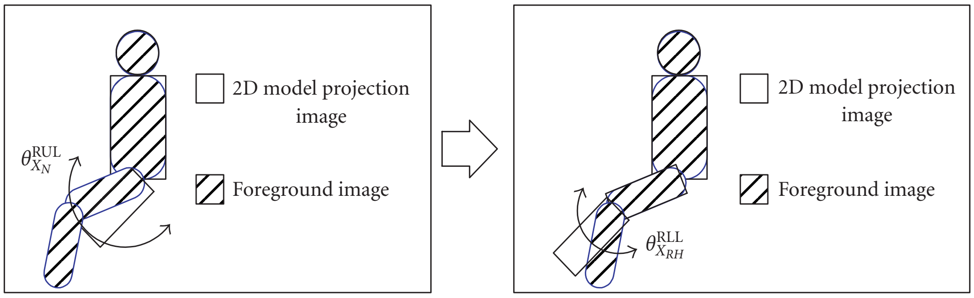
If $\theta_{Z_N}^{\text{RUL}}$ (from the facade viewer) is greater than 175° but less than 180°, the human is lifting his right leg, else he is not. Then, we may integrate $\theta_{Z_N}^{\text{RUL}}$ (from the facade viewer), $\theta_{X_N}^{\text{RUL}}$ (from the flank viewer), and $\theta_{X_{RH}}^{\text{RLL}}$ (from the flank viewer) into the universal BAPs. Similarly, we can find the similar case of the left leg movement. The universal BAPs can be extracted by integrating BAPs of two viewers as the universal BAPs.



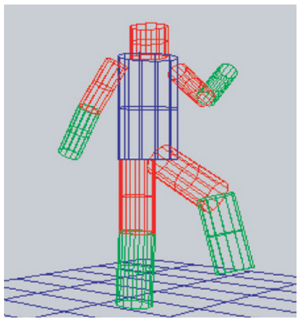
(a)



(b)



(c)



(d)

FIGURE 15: The facade viewer and the flank viewer estimate $\theta_{X_N}^{RUL}$, $\theta_{X_{RH}}^{RLL}$, $\theta_{X_N}^{LUL}$, and $\theta_{X_{RH}}^{LLL}$. (a) Squatting down (the facade view). (b) Virtual actor is squatting down. (c) Leg lifting (the facade view). (d) Virtual actor is lifting his leg.

5. EXPERIMENTAL RESULTS

The color image frame is 160×120×24 bits and the frame rate is 15 frames per second. Each test video sequence lasts more than 2 seconds, so that it may consist of about 40 frames. We use two computers equipped with video capturing equipment. Our system analyzes and estimates the BAPs of human motion in real time, based on the matching between the articulated human model and the 2D binary human object. In the experiments, we illustrate 15 human postures composed of the following five basic movements: (1) walking; (2) arm raising; (3) arm swing; (4) squatting; (5) kicking. To evaluate the performance of our tracking process, we test the system by using 15 different human motion postures. Each one is performed by 12 different individuals. People with casual wear and no markers are instructed to perform 15 different actions as shown in Figure 16.

We cannot measure the real BAPs from the human actor for comparing the real BAPs with the estimated BAPs. To evaluate the system performance, we use the HMM model to verify whether the estimate BAPs are correct or not. HMM is a probabilistic state machine widely used in human gesture and action recognition [21, 22, 23]. The HMM-based human posture recognition consists of two phases: training phase and recognition phase.

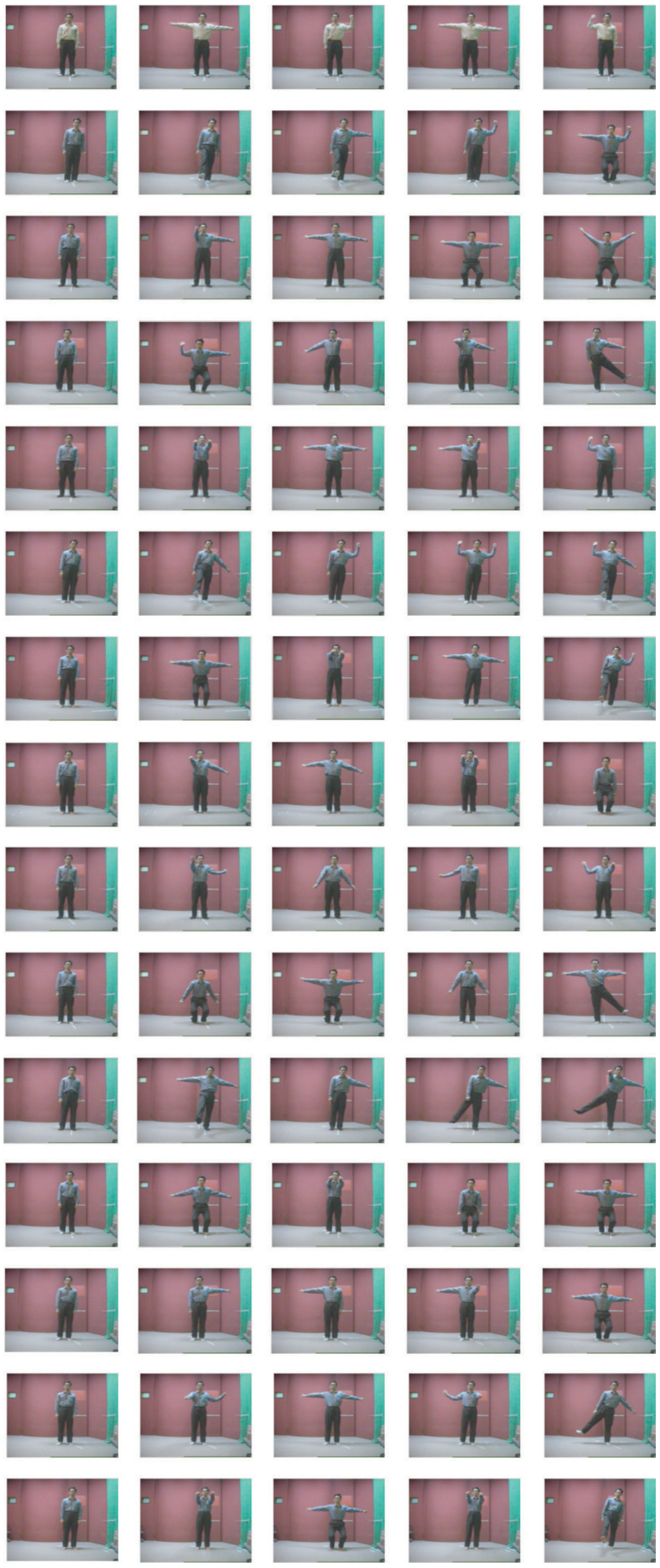


Figure 16: The 15 human postures in our experiment.

Table 2: The number of correct recognitions for each posture.

Posture	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Correct Recognition	22	21	23	21	24	20	23	24	22	22	23	24	21	22	20

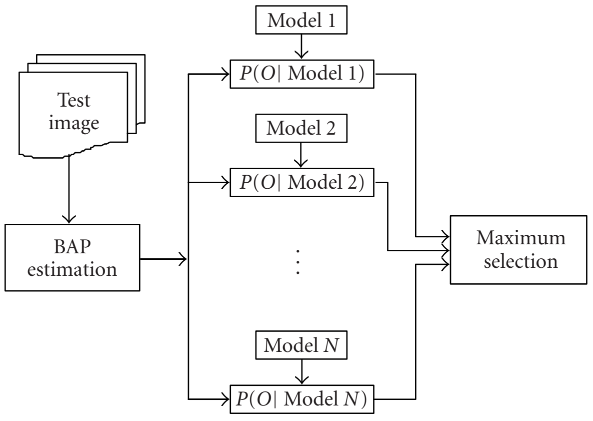


Figure 17: The evaluation system.

5.1. *Training phase*

A set of the joint angles (i.e., BAPs) have been extracted from each video frame which are combined as a so-called feature vector. A feature vector will be assigned to an observation or to a symbol. To train the HMMs, we need to determine some parameters: the observation number, the state number, and the dimension of the feature vector. There is a tradeoff between selecting a large observation number and a faster HMM computation. A larger one means more accurate observations and more computation. From the experiments, we choose 64 symbols. The issue of the number of states also needs to be determined. The states are not necessarily corresponding to the physical observations of the corresponding process. The number of states and the number of the different postures in human motion sequences are related. Here, we develop the 5-state HMM, which is most suitable for our experiments.

The tracking process has estimated the joint angles of the human actor, and there are 17 joint angles for the human model. Actually, not all of the joint angles are required for describing different postures. Hence, we only choose some influential joint angles representing the postures, such as the joint angles θ_x and θ_z of the shoulders, θ_x of the elbows, and θ_x and θ_z of the hips. Totally, 10 joint angles are selected as one feature vector. Here, we need to train 15 HMMs corresponding to 15 different postures. The training process will generate the model parameter λ_i for the ith HMM.

5.2. *Recognition phase*

In our experiments, there are 360 testing sequences for performance evaluation. There are 15 different human postures, and each one is performed twice by 12 different individuals. As shown in Figure 17, every testing sequence, O, is evaluated by 15 HMMs. The likelihood of the observation sequences can be computed for each HMM as $P_i = \log(P(O|\lambda_i))$, where λ_i is the model parameter of the ith HMM. The HMM with maximum likelihood is selected to represent the recognized posture which is currently performed by the human actor in the test video sequence.

The experimental results are shown in Table 2. Each posture is tested 24 times by 12 different individuals. The recognition errors are caused mainly by the incorrect BAPs. The BAP estimation algorithm may fail if the extracted foreground object is noisy or ambiguous, which is caused by the occlusion between the limbs and the torso. The limitation of our algorithm can be summarized as follows.

(1) Since the BAP estimation is based on the preceding BAP in the previous time instance, the error propagation cannot be avoided. Once the error of the previous BAP is above certain level, the search range for the following BAP no longer covers the correct BAP, and the system may crash.

(2) The occlusion of human body is the major challenge for our algorithm. By using two views, some occlusion in one view should be clear in the other view. However, if the arm is swing beside the torso, it makes occlusion in both the facade and flank views. The occlusion among the limbs and the torso will make BAP estimation fail, since the matching process cannot differentiate the limb from the torso in the silhouette image.

(3) Arm swing is another difficult issue. The side viewer cannot differentiate whether one arm or two arms is being raised. The silhouette of the arm swing viewed from the front view is not very reliable for accurate angle estimation.

(4) It cannot tell if a facade is a front view or just a back view. We may add the face-finding algorithm to identify whether the human actor is facing toward the camera or not.

6. CONCLUSION AND FUTURE WORKS

We have demonstrated real-time human motion analysis method for HCI system by using a new overlapped hierarchical tritree search algorithm with less searching time and wider search range. The wider search range enables us to track some fast human motions under lower frame rate. In the experiments, we have shown some successful examples. In the near future, we may extend to multiple person tracking and analysis, which may be used in HCI systems such as human identification, surveillance, and gesture recognition.

REFERENCES

[1] G. Johansson, "Visual motion perception," *Scientific American*, vol. 232, no. 6, pp. 76–89, 1975.

[2] A. G. Bharatkumar, K. E. Daigle, M. G. Pandy, Q. Cai, and J. K. Aggarwal, "Lower limb kinematics of human walking with the medial axis transformation," in *Proc. IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pp. 70–76, Austin, Tex, USA, November 1994.

[3] Y. Li, S. Ma, and H. Lu, "A multiscale morphological method for human posture recognition," in *Proc. IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 56–61, Nara, Japan, April 1998.

[4] M. K. Leung and Y.-H. Yang, "First sight: a human body outline labeling system," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 17, no. 4, pp. 359–377, 1995.

[5] J. O'Rourke and N. I. Badler, "Model-based image analysis of human motion using constraint propagation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 2, no. 6, pp. 522–536, 1980.

[6] K. Sato, T. Maeda, H. Kato, and S. Inokuchi, "CAD-based object tracking with distributed monocular camera for security monitoring," in *Proc. 2nd CAD-Based Vision Workshop*, pp. 291–297, Champion, Pa, USA, February 1994.

[7] D. Marr and H. K. Nishihara, "Representation and recognition of the spatial organization of three-dimensional shapes," *Proc. Roy. Soc. London. Ser. B.*, vol. 200, no. 1140, pp. 269–294, 1978.

[8] T. B. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *Computer Vision and Image Understanding*, vol. 81, no. 3, pp. 231–268, 2001.

[9] K. Tamagawa, T. Yamada, T. Ogi, and M. Hirose, "Developing a 2.5-D video avatar," *IEEE Signal Processing Magazine*, vol. 18, no. 3, pp. 35–42, 2001.

[10] K. Rohr, "Human movement analysis based on explicit motion models," in *Motion-Based Recognition*, M. Shah and R. Jain, Eds., vol. 9 of *Computational Imaging and Vision*, chapter 8, pp. 171–198, Kluwer Academic Publishers, Boston, Mass, USA, 1997.

[11] C. R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: real-time tracking of the human body," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780–785, 1997.

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

[13] I. Haritaoglu, D. Harwood, and L. S. Davis, "A fast background scene modeling and maintenance for outdoor surveillance," in *Proc. IEEE 5th International Conference on Pattern Recognition (ICPR '00)*, vol. 4, pp. 179–183, Barcelona, Spain, September 2000.

[14] Q. Cai and J. K. Aggarwal, "Automatic tracking of human motion in indoor scenes across multiple synchronized video streams," in *Proc. IEEE Sixth International Conf. on Computer Vision (ICCV '98)*, pp. 356–362, Bombay, India, January 1998.

[15] D. M. Gavrila and L. S. Davis, "3-D model-based tracking of humans in action: a multi-view approach," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '96)*, pp. 73–80, San Francisco, Calif, USA, June 1996.

[16] R. Cutler and L. Davis, "Robust real-time periodic motion detection, analysis, and applications," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 781–796, 2000.

[17] Y. Ricquebourg and P. Bouthemy, "Real-time tracking of moving persons by exploiting spatio-temporal image slices," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 797–808, 2000.

[18] A. Nkazawa, H. Kato, and S. Inokuchi, "Human tracking using distributed vision system," in *Proc. IEEE 14th International Conference on Pattern Recognition (ICPR '98)*, vol. 1, pp. 593–596, Brisbane, Australia, August 1998.

[19] A. Utsumi, H. Yang, and J. Ohya, "Adaptive human motion tracking using non-synchronous multiple viewpoint observations," in *Proc. IEEE 15th International Conference on Pattern Recognition (ICPR '00)*, vol. 4, pp. 607–610, Barcelona, Spain, September 2000.

[20] S. Iwasawa, J. Takahashi, K. Ohya, K. Sakaguchi, T. Ebihara, and S. Morishima, "Human body postures from trinocular camera images," in *Proc. 4th IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 326–331, Grenoble, France, March 2000.

[21] C. Bregler, "Learning and recognition human dynamics in video sequrence," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '97)*, pp. 568–574, Puerto Rico, June 1997.

[22] I.-C. Chang and C.-L. Huang, "The model-based human body motion analysis system," *Image and Vision Computing*, vol. 18, no. 14, pp. 1067–1083, 2000.

[23] M. Brand and V. Kettnaker, "Discovery and segmentation of activities in video," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 844–851, 2000.

[24] N. Krahnstover, M. Yeasin, and R. Sharma, "Towards a unified framework for tracking and analysis of human motion," in *Proc. IEEE Workshop on Detection and Recognition Events in Video*, pp. 47–54, Vancouver, Canada, July 2001.

Chung-Lin Huang was born in Tai-Chung, Taiwan, in 1955. He received his B.S. degree in nuclear engineering from the National Tsing-Hua University, Hsin-Chu, Taiwan, in 1977, and his M.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1979, respectively. He obtained his Ph.D. degree in electrical engineering from the University of Florida, Gainesville, Fla, USA, in 1987. From 1981 to 1983, he was an Associate Engineer in ERSO, ITRI, Hsin-Chu, Taiwan. From 1987 to 1988, he worked for the Unisys Co., Orange County, Calif, USA as a project engineer. Since August 1988, he has been with the Department of Electrical Engineering, National Tsing-Hua University, Hsin-Chu, Taiwan. Currently, he is a Professor in the same department. His research interests are in the area of image processing, computer vision, and visual communication. Dr. Huang is a Member of IEEE and SPIE.

Chia-Ying Chung was born in Tainan, Taiwan, in 1977. He received his B.S. degree in 1999 and M.S. degree in 2001, both from Department of Electrical Engineering, National Tsing-Hua University, Hsin-Chu, Taiwan. Since 2001, he has been working for Zyxel Communication Co., Hsin-Chu, Taiwan. His research interests are in video communication and wireless networking.