A MODEL-BASED ARTICULATED HUMAN MOTION TRACKING SYSTEM

Chung-Lin Huang, Tze-Hou Tseng and Huang-Chia Shih
Electrical Engineering Department
National Tsing-Hua University
Hsin-Chu, Taiwan, ROC
e-mail: clhuang@ee.nthu.edu.tw

ABSTRACT

This paper proposes a model-based human motion analysis method using two cameras without any markers on the human actor. Using the background subtraction, we extract the silhouette of the human object from two orthogonal views. To track the human actor, we find the best match between the articulated human model and the binary human object. We propose a 3-level estimation algorithm which consists of low-level analysis, mid-level estimation and high-level verification and correction. Moreover, to relax the limitations of using the articulated human model, we predefine some auxiliary postures, and develop a projection profile analysis to recognize the postures of bending and rotating. Finally, to evaluate the performance of the tracking system, we use HMMs to recognize different human motions.

1. INTRODUCTION

Human motion analysis has received an increasing interest which is motivated by a wide spectrum of applications, such as athletic performance analysis, surveillance, man-machine interfaces, content-based image storage retrieval, and video conferencing. To analyze the human body structure through image sequences, the human bodies are represented as stick figures, 2D contours, or volumetric models. Hence, each body segment can be approximated as lines, 2D ribbons and elliptical cylinders. The human body can also be represented by the stick figure, which consists of line segments linked by joints. This concept was initially proposed by Johansson [1], who marked joints as moving light displays, and analyzed the motion of joints for motion estimation and recognition of the whole figure.

Webb *et al.* [2] use the 3D structures of Johansson-type figures in motion. Based on the fixed axis assumption, they assume that each rigid object motion is constrained and the depth of the joints can be estimated from their 2D projections. Kurakake *et al.* [3] obtain the joint locations of walking humans by establishing correspondence between extracted ribbons. They assumed small motion between two consecutive frames, and feature correspondence was conducted using various geometric constraints.

Human vision interprets moving figures based on *a priori* human shape models. Thus most of the human motion analysis methods fit models to the given images [4~6]. Having analyzed the moving human image and extracted the motion features from the image sequences, the next step is to understand the human behavior or activities. There are two typical approaches to recognize human activity or behavior: template matching approach and state-space approach. The former approaches compare the features extracted from the given image sequence to the pre-stored patterns during the recognition process. The latter defines each static posture as a state. Hidden Markov Model (HMM) has been widely used to model the human motions [7~8].

This paper introduces a model-based human motion tracking without using any markers. It uses two orthogonal cameras to analyze the human motion. We assume that background is stationary, and the human silhouette can be easily extracted by background subtraction. First, we create a 3-D articulated human model consisting of ten cylinders, representing torso, head, arms, and legs, and 10 connecting joints. Then we find the Body Definition Parameters (BDPs) and Body Animation Parameters (BAPs) which are used to describe the human motion parameters. BDPs are defined as the shapes of the human body extremities, and BAPs are the joint angles of the human body extremities.

Motion analysis process consists of three-level components: low-level analysis, mid-level BAPs estimation and high-level verification and correction. The human posture in every frame can be presented by a set of BAPs. To evaluate the performance of our system, we apply HMM for posture recognize and measure the recognition rate for each posture.

2. MODEL-BASED MOTION ANALYSIS

Model-based computer vision systems consist of two processes. The first one is the *bottom-up* process which analyzes the input images to extract the meaningful silhouette of the human object. The second process is *top-down* process which searches for BDPs and BAPs based on the matching between the silhouette and model.

2.1 Bottom-up Processing

The bottom-up processes may extract a foreground human object for top-down process to find the best-matched model. There are three main parts: camera calibration, background subtraction,

and morphology filtering shown as follows:

1. Camera Calibration. There are two cameras: one camera estimates the posture of the actor in the facade view, and the other estimates the flank view. The center of action region is at the origin of the world coordinate and the positions of the two cameras are fixed. The focal axis of camera 1 is along the *z*-axis, and the focal axis of camera 2 is along the *x*-axis, that is, the image plane of camera 1 is the *x*-*y* plane, and the image plane of camera 2 is the *z*-*y* plane. The camera calibration for our system is easy. We simply check whether two cameras are placed without tilt, the tilt angle is zero. To check the two camera, we paste horizontal line marker on the wall, and then adjust the camera viewing angle until the horizontal line markers overlap the lines $x=80$ and $x=81$.

2. Background Subtraction. The initial state of the human motion analysis problem is the extraction of targets from a video streams, we use the background subtraction[9] to extract the moving human object. The pixels of each frame of the video sequence are segmented into background and foreground pixels based on a statistical model of the background pixels. The pixels with large deviations from the background model are taken to be foreground pixels. The set of connected foreground pixels determine the foreground regions.

3. Morphology Filtering. After background subtraction, there will be spurious pixels and holes in the extracted object. To clean up anomalies in the object and spurious detection, we use two morphology operators: *Closing* and *Opening* operators. The former repairs the hole inside the foreground; the latter removes small noise. The results of morphology filtering are shown in Figure 1.



Figure 1. (a) The original images (a) the image of foreground object after morphological filtering.

2.2 Articulated 3-D Human Model

The top-down process applies the 3-D articulated human model for the tracking. The human model is represented by 10 elliptical cylinders. Each cylinder is described by three parameters: the length of the axis, the major and the minor axes of the ellipse cross section. The cylinders are connected by joints with different degree-of-freedoms (DOFs).

For each cylinder of the model, two sets of parameters are required: shape parameters and angle parameters. The shape parameters are the height, long radius, and short radius of the cylinder. The angle parameters describe the pose of the human body in terms of the angles of the joints, which connect the cylinders. The angle parameters of ten cylinders represent the ten joints of human model. These 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 flexion and spherical types. A flexion joint has only one DOF while a spherical one has three DOFs. Here, we fix the neck joint and limit the navel joint to only one DOF(θ_y). The shoulder and the hip joints are the spherical type, whereas the elbow joints, knee joints and the navel joint are the flexion type.

After tracking the posture of the human motion, we will generate two parameters: Body Definition Parameters (BDPs) and Body Animation Parameters (BAPs). The BDPs are the set of the shape parameters of the human body extremities. The BAPs are the set of the angle parameters of the joints of human body extremities. In the first frame of the human motion video sequence, the BDPs are determined, and the BAPs are also initialized. In the consequent frame, the model only adjusts the BAPs to track the human posture. We assume that the variation of BAPs between two neighboring frames is limited, so that the BAPs of the frame at time t are estimated based on the previous BAPs.

3. BDP DETERMINATION

In the beginning, we limit the human posture as shown in Figure 1, so that we may determine the BDPs from the front view and the side view.

3.1 BDPs Estimation from the Front View

First, we construct the vertical projection profile of the binary foreground image $P_v(x)$ of which the average is defined as $avg=\int P_v(x)dx \,/\, x$. Then we scan the profile from left to right to find the value x_1, which is the smallest x value where $P_v(x_1)>avg$. Similarly, we scan the histogram from right to left to find the value x_2, which is the largest x value where $P_v(x_2)>avg$. We may also define the center of body in X-axis and the width of torso.

We remove the head by applying morphological filtering operations, which consist of the morphological closing and the morphological opening as shown in Figure 2. Then we can find the location of shoulder in Y-axis and define the length of head. We assume that the ratio of length of torso and leg is 4:6, so that we have the length of leg. We also assume the length of lower legs is the same as that of upper legs.

The location of center of body is defined at the position of navel. Then we may find the BDPs including the long radius of torso, head, and leg as well as the short radius of head and leg. Average the location of the pixels on the left boundary of body in y-axis to find the terminal position in y-axis of

right arm (y_l). Average the location of the pixels on the right boundary of body in y-axis to find the terminal position in y-axis of left arm (y_r). From the terminals of arms, as shown in Figure 3(a), we may define the position of shoulder joints as shown in Figure 3(b). From the extreme position of arms and position of shoulder joints, we may calculate the length of upper arm and lower arm.

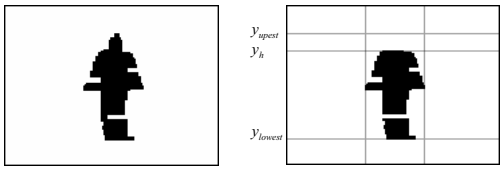


Figure 2. Head removed image.

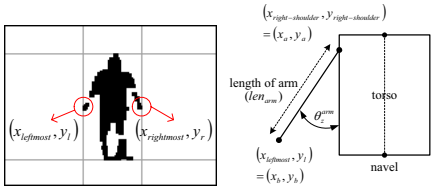


Figure 3. (a) The extreme position of arms. (b) Calculate the radius and length of arm.

3.2 BDPs Estimation from the Side View

Using the silhouette and its corresponding vertical projection profile, we may find the average of the histogram of vertical projection profile *avg*. Then, we scan the histogram from left to right to find the value x_1 , and from right to left, to find the value x_2 of which $P_v(x_1)$>avg and $P_v(x_2)$>avg. Finally, we define the short radius of torso.

3.3 BDPs Integration

The distances from two cameras to the actor are not equal, which result in different scales in the image, and the sizes of human object in two views are not equal. We simply define a *scaling factor* $r_t=t_2/t_1$, where t_1 and t_2 are the heights of human in the front view and in the side view, respectively. Since the BDPs are estimated independently from two views, if the short radius of torso from view 2, $SR_{torso,2}$, is selected as the universal BDP, then it is adjusted by $SR_{torso,u}=SR_{torso,2}/r_t$.

4. MODEL-BASED BAP ESTIMATION

To track the human motion (or find the BAPs), we propose three level motion parameter estimation processes: low-level analysis, mid-level BAPs estimation, and high-level verification and correction. The low-level analysis examines the vertical and horizontal projection profiles from two views to find the possible posture of the actor. It can be bending, rotating, or other auxiliary postures. Then the preliminary results are helpful to the mid-level BAPs estimation process, which is mainly a pose tracking operation.

Finally, the high-level estimation process will analyze the BAPs from two consecutive image frames. Since the posture of the actor will not change abruptly between two neighboring frames, we may find the consistency between the two sets of BAPs. We define some regular postures as "states". The high-level verification and correction process predicts the state at time *t* based on the previous states before time *t*. If the predicted state at time *t* is unreasonable, the false alarm will be triggered, and the estimation process will adjust the estimated motion parameters (BAPs) by another matching algorithm using pre-defined joint angles. The flow diagram of three-levels operations to search for BAPs is shown in Figure 4.

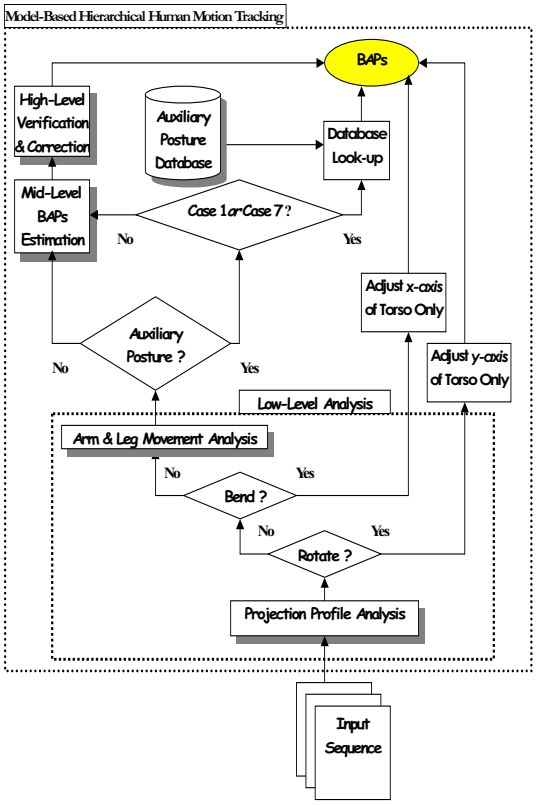


Figure 4. The tracking procedure in our system.

4.1 Low-Level Analysis

The low-level analysis consists of the following five detectors:

a) Rotation Detector (RD). Since the width of the torso in the façade view is greater than the corresponding width in the flank view, RD determines the façade or flank in two views as follows: if $W_{torso,1} \geq W_{torso,2}$ then view 1 is the façade else view 2 is the façade, where $W_{torso,1}$ and $W_{torso,2}$ are the projection profile width of the torso in view 1 and view 2, respectively.

b) Bending detector (BD). BD analyzes the characteristics of *position, height,* and *out-of-range* of the projection profile of the silhouette and then determines whether the human object is bending or not. If the human actor is bending, we assume that

he stands with two hands closed to the torso so that the mid-level and high-level processes will find all of the joints angles except the angle of *x*-axis of torso. BD analyzes the posture of the arms and the legs independently.

c) Arm movement detector (AMD). We assume that only one arm is lifted forward on z-y plane at a time. If two arms are lifted forward simultaneously, there is only one forward raising arm found in the side view. The variation of the projection profile of the forward arm-lifting binary silhouette in the façade view is not obvious.

Arm-Raising Cases	Sideward (Right)	Sideward (Left)	Forward
0	False	False	False
1 (Ambiguous)	False	False	True
2	False	True	False
3	False	True	True
4	True	False	False
5	True	False	True
6	True	True	False
7 (illegal)	True	True	True

Table 1. The eight cases determine by AMD.

AMD analyzes the projection profiles and finds that there are eight possible combinations of arm-raising postures as shown in Table 1. "Sideward (Right)" means the right arm stretches on *x-y* plane, and "Sideward (Left)" means the left arm stretches on *x-y* plane. "Forward" means at least one arm is lifted on *z-y* plane. The "True" means the right or left arm is on *x-y* or *z-y* plane, and the "False" indicates otherwise. For instance, *case 5* indicates that the right arm moves on *x-y* plane and the left arm moves on *z-y* plane as illustrated in Figure 5. The profile analysis is not accurate. There may be two or more different postures that generate the similar vertical projection profile in two views, such as Case 1 which is an ambiguous case. *Case 7* is an illegal case, which is found due to the noisy silhouette.



Figure 5. The 5th case of the raising-arm posture.

d) Leg movement detector (LMD). In our system, the legs of articulated model can perform three kinds of postures: standing, leg lifting and squatting. To decide whether the actor is squatting, LMD analyzes the horizontal projection profile of the façade view. Similar to finding the position of the torso, LMD scans the histogram from top to bottom to find the smallest y value where the corresponding projection height is greater than avg as shown in Figure 6. Since the identified position is roughly located at the shoulder, LMD averages the values of the identified location from the first ten frames as the squatting threshold. In the consequent frames, LMD may use the threshold to determine whether the actor is squatting or not.

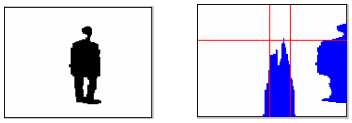


Figure 6. The posture of the squatting

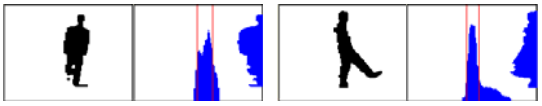


Figure 7. The posture of leg lifting in (a) façade view and (b) flank view.

To identify whether the person in the camera view is lifting leg, LMD analyzes the vertical projection profile of the lower part of the human object in flank view. Here, LMD analyzes the right section of the projection profile with *x* coordinate greater than x_2. Similar to AMD, LMD finds the lifting legs, except using different threshold for judging the height of each vertical line. As shown in Figure 7, LMD finds whether the centroid of the vertical projection profile is inside the torso to determine whether the actor is lifting his right leg. lifting his left leg, or standing with two legs

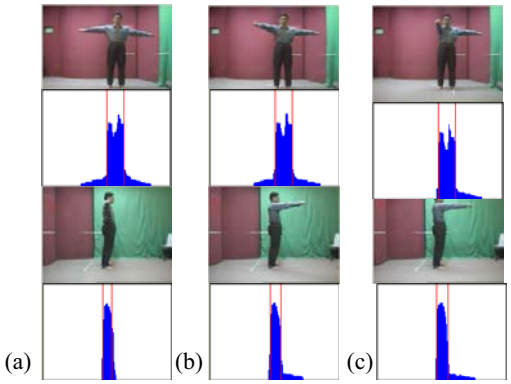


Figure 8. AP 1 consists of a transition sequence of (a) case 6, (b) case 7, and (c) case 3.

e) Auxiliary postures detector (APD). To analyze the projection profile can only identify limited postures, *i.e.,* the limbs are constrained to stretch on *x-y* and *y-z* planes. Therefore, APD needs to find the so-called auxiliary postures (APs) by analyzing the vertical projection profiles. Take the first AP for example (i.e., Figure 8), APD first finds the actor stretching two arms on the *x-y* plane (*Case 6*), then an ambiguous case (*Case 7*), and finally detects the right arm is on the *y-z* plane and the left arm is on the *x-y* plane (*Case 3)*. If APD finds the case transition as $6 \rightarrow 7 \rightarrow 3$, then it may conclude that the human actor is performing the first AP. We define 9 APs. The APD uses different case transition sequence to identify different APs as illustrated in Table 2. For each AP, the initial and last cases of the transition case sequences can not be case 1 nor case 7. If case 1 or case 7 is found in

the case transition case sequence, the following mid-level process cannot provide correct BAP information. So we need to predefine the joint angles for cases 1 and 7 of the APs as shown in Table 3.

AP	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
1st case	6	3	6	5	6	0	3	5	6
2nd case	7	7	7	7	7	1	7	7	7
3rd case	3	6	5	6	1	7	5	3	1
4th case					0	6			7
5th case									6

Table 2. The case transition sequences of 9 APs.

Case	AP	θ_z^{RUA}	θ_x^{RUA}	θ_x^{RLA}	θ_z^{LUA}	θ_x^{LUA}	θ_x^{LLA}
7	1, 2	90°	45°	0°		0°	
7	3, 4		0		90°	45°	0°
7	5~ 9	90°	45°	0°	90°	45°	0°
1	5, 6	0^0	0°~90°	0°	0°	0°~90°	0°

Table 3. The pre-determined joint angles for APs

4.2 Mid-Level BAP Estimation

We divide the estimation into three stages: (1) Arm joint angle estimation; (2) Leg joint angle estimation; (3) BAPs integration. To perform the similarity between the articulated model and the actor, we use a simple calculation method as S=*num*(I∩M)/*num*(I∪M), where S is the similarity score, I is the set of image, M is the set of model, and *num*(X) is the number of pixels in the set X. In each stage, only a few parameters are estimated while the remaining parameters are fixed.

1. Arm joint angle estimation(AJE). To search for the BAPs, we use the overlapped tri-tree algorithm [10]. Based on the probable posture, we only have to search for the possible joint angles. Because of the occlusion, to estimate the joint angles in two views are quite different. To simplify the estimation, we assume the arms only bend on certain 2-D planes so that the search space is limited. The search procedure is hierarchical: the upper arms are first calculated, then the lower arms are estimated based on the upper arms as illustrated in Figures 9(a) and 9(b). In the flank view, we only estimate θ_{X_N} as shown in Figure 9(c).

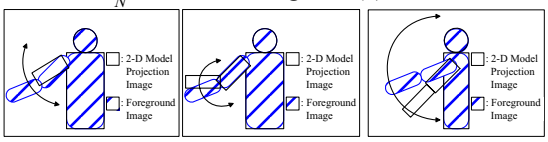


Figure 9. (a) Rotate upper arm around Z_N axis; (b) Rotate lower arm around X_{RS} axis; (c) Rotate arm around X_N axis.

2. Leg joint angle estimation (LJE). The postures of standing, squatting, and leg lifting are estimated. To estimate the leg lifting, we use the image of the flank view and estimate hierarchically the angle of the upper leg, and then the angle of the lower leg. Shown in Figure 10, we first adjust the joint angle θ_z of the hip, and calculate the joint angle θ_x of the hip with the constraint that the angle of the knee is twice as the joint angel of the hip.

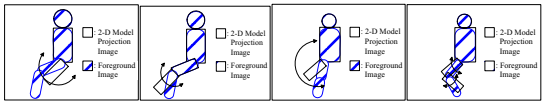


Figure 10. From left to right: (1) Determine the joint angle of leg lifting in flank view: rotate upper-leg around X_N axis then rotate lower-leg along X_N axis. (2) Determine the joint angle of the squatting in façade view: rotate upper leg around Z_N axis, then rotate upper and lower legs along X_N axis.

3 BAPs Integration. The universal integrator takes X_W^T estimated by viewer 1, Z_W^T estimated by viewer 2, and the model takes Y_W^T estimated by the facade viewer, since there are more details in the facade viewer. The AJE estimates $\theta_{X_N}^{RUA}$ and $\theta_{X_N}^{LUA}$ from the flank view and the remainder of BAPs of arms from the facade view, including the rotation angles of shoulder joints around Y_N axis and Z_N axis of the navel coordinate and the rotation angles of elbow joints around X_N axis of the shoulder coordinate.

4.3 High-level Verification and Correction

High-level process analyzes the consistency of the estimated BAPs based on the previous BAPs. Here we define states as some pre-defined angles of the joints. We partition the joints of the limbs of 3-D human model into three independent sections: *right arm section, left arm section, and legs section*. In each section, the variation of BAP is represented by a sequence of state transition. The combination of the state-transitions defines posture of the model. In the right and left arms section, we choose the joint angles of the shoulder in *x*-axis and *z*-axis as the states. Similarly, the joint angles of both two hips in *x*-axis and *z*-axis are selected as the states. Since the posture of the actor will not change abruptly in two consecutive frames, the current state at time *t* will either stay unchanged or transfer to its neighboring states.

If one of the following conditions occurs, the high level process will discover the error: (1) The Euclidean distance between the current BAPs and any of pre-defined states is greater than certain threshold. (2) The current state doesn't stay unchanged nor transfers to the neighboring states. Once the high-level verification process finds an error BAP, it will trigger a near-full search mechanism to final the correct BAPs. The search algorithm selects the angles pre-defined in a *lookup table,* which contains some joint angle vectors of shoulders and hips for verification.

5. EXPERIMENTAL RESULTS

We evaluate the performance of our system based on a sequence of BAPs. These BAPs are

treated as the feature vectors, which can be applied to recognize the human posture. The color image frame size is 160×120 and the frame rate is 15. To test the efficiency of our tracking system, we track different human postures, each one is tracked 24 times. There are 40 frames in each test video sequence.

We apply the HMM[11] to model the spatial and temporal characteristics of human motion. To train the HMMs, we need to determine some parameters: the number of symbols, the number of states, and the dimension of the joint angles. The states are not necessarily corresponding to the physical observations of the corresponding process. Here, we apply an 8-state HMM for identifying 15 different postures.



Figure 11. The 15 human motions.

However, not all of the DOFs of the joints are required for distinguishing the different postures. Thus, to recognize the human motions, we only choose some influential DOFs as the joint angle vectors representing the postures. We choose the joint angles of the θ_x and θ_z of the shoulders, the θ_x of the elbows and the θ_x and θ_z of the hips, there are totally 10 DOFs describing the postures. Each human posture is first trained by an HMM, with 8 states, 32 symbols, and 10 DOFs. The recognition rate of every posture is above 98% indicating that our tracking system is very successful.

The hierarchical matching will cause error propagations. The errors in finding the position of torso will cause more errors in estimating the joint angle. Another limitation of our model is that it can not track human actor doing any kind of 3-D motion. Since we can analyze 2-D information from two views, the motion is limited to the limb-stretching on *x-y* and *z-y* planes only. We apply the APs and APD to compensate these limitations in low-level process.

6. CONCLUSION

This paper has demonstrated a human motion analysis system based on two views: the facade view and the flank view to overcome the occlusion. It consists of a 3-level tracking algorithm: low-level analysis, mid-level estimation and high-level verification and correction to track the posture. We have demonstrated that our system can analyze 15 different human motions and the result of the recognition is satisfying.

REFERENCES

[1] G. Johansson, “Visual motion perception,” Sci. Am. 232(6), 1975, 76-88.

[2] J. A. Webb et al., “Structure from motion of rigid and jointed objects,” Artif. Intell. 19, 1982, 107-130.

[3] S. Kurakake and R. Nevatia, “Description and tracking of moving articulated objects,” in 11th ICPR, Hague, Netherlands, 1992.

[4] A. G. Bharathumar, et al. “Lower limb kinematics of human walking with the medial axis transformation,” IEEE Workshop on Motion of Non-Rigid and Articulated Objects, Austin, 1994.

[5] A. F. Bobick and J. Davis, “Real-time recognition of activity using temporal templates,” IEEE Computer Society Workshop Applications on Computer Vision, pages 39-42, Sarasota, FL, 1996.

[6] K. Rohr. “Towards model-based recognition of human movements in image sequences,” CVGIP: Image Understanding, 59(1):94-115, 1994.

[7] C. Bregler, “Learning and recognition human dynamics in video sequrence,” Proc. IEEE Conf. on CVPR, pp.568-574, 1997.

[8] M. Brand and V. Kettnaker, “Discovery and Segmentation of activities in video,” IEEE Trans. on PAMI, vol.22, no.2, pp.844-851, Aug. 2000.

[9] T. Horprasert, D. Harwood, and L. S. Davis, “A robust background subtraction and shadow detection,” The 4th ACCV, Taipei, Taiwan, 2000.

[10] C. L. Huang and C. Y. Chung, “A Real-Time Model-Based Human Motion Analysis System” ICME 2003, Baltimore, July 7-10, 2003.

[11] L.R. Rabiner, “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition,” Proc. IEEE, vol. 77, no. 2, pp. 257-2856, Feb. 1989.