Real-time Human Motion Capturing and Recognition

Bau-Cheng Shen, Huang-Chia Shih and Chung-Lin Huang,
Departmenrt of Electrical Engineering
National Tsing-Hua University, Hsin-Chu, Taiwan.
E-mail: clhuang@ee.nthu.edu.tw, royal_shih@hotmail.com

Extended Abstract

In this paper, we represent a real time vision-based human motion capturing and recognition system using two calibrated CCD cameras. We propose a simple but effective method to estimate the motion parameters (BAPs) of the human object by analyzing the vertical projection profile and the horizontal projection profile in each view to identify different arm and leg postures. With the identified postures, we can apply the Kalman filtering to capture the motion parameters (joint angels). Our method is divided into *macro motion analysis* and *micro motion analysis*. The former identifies certain well-defined postures and the latter traces the variation of joint angle or BAPs. In the experiments, we test 22 different arm and leg postures and show the errors of the estimated BAPs. Finally, based on the estimated BAPs, we apply Hidden Markov Model (HMM) to recognize 18 different human motion actions which consists of different posture sequence. The extracted object motion and the corresponding avatar are shown in Figure 1.

The preprocessing module is developed to extract the human object followed by the human motion capturing module. Here, we apply a 3-D articulated human model which consists of ten elliptical cylinders (representing torso, head, arms, and legs) and 10 connecting joints. First, we assume that the focal axes of two cameras are along the *z*-axis and the *x*-axis respectively, *i.e*, 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 is simple, two cameras are placed with zero tilt angle. To verify, we paste a horizontal line marker on the wall, and adjust the camera tilt angle until the horizontal line markers overlap the lines $x=80$ and $x=81$. To extract the silhouette of human object, we apply the well-known background subtraction method which extracts the foreground object by evaluating the difference between the background model and the current image.

Based on the vertical projection profiles of the human silhouette from two views, we may construct 3-D human model. The variation of the joint angles is the prominent features for analyzing human motion. The 3-D articulated model has been used as a top-down process for the human motion tracking. 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 in the model are connected by joints with different degree-of-freedoms (DOFs).

The BAPs (motion parameters) 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 either flexion or spherical. The former has 1 DOF, whereas the latter has 3 DOFs. Totally, there are four spherical and five flexion joints, and the human model has 17 DOFs.

First, we assume the initial posture to be standing up with stretched-arms. To evaluate Body Definition Parameters (BDPs), we construct the vertical projection profile of the binary human silhouette in façade as $P_{v,l}(x_i)$ and $x_r \leqq x_i \leqq x_l$, where x_l and x_r are the left and right boundary of the profile. Second, we remove the head by applying morphological filtering operations, which consist of the morphological closing and then opening operations by a structuring element (a horizontal rectangle with size $0.8W_{torso}\times 1$). Third, we define the center of body in y-axis: $y_c = y_h - len_{torso}$. Let (x_c, y_c) be the location of center of body which happens to be the position of navel. Define the long radius of torso, the long radius of head, the short radius of head, the long radius of leg, and the short radius of leg as follows: $LR_{torso}=0.5W_{torso,1}$, $LR_{head}=0.2W_{torso,1}$, $SR_{head}=0.16W_{torso,1}$, $LR_{leg}=0.2W_{torso,1}$, and $SR_{leg}=0.36W_{torso,1}$. Finally, we average the location of the pixels on the left (or right) boundary of body to find the extreme position of right (or left) arm as y_l (or y_r). Having extracted the extreme positions, $(x_{leftmost}, y_l)$ and $(x_{rightmost}, y_r)$, we may define the position of shoulder joints as $(x_{right\text{-}shoulder}, y_{right\text{-}shoulder}) = (x_c - LR_{torso}, y_c - len_{torso} + 0.45LR_{torso})$.

There are four posture categories: bending, arm postures, over-head and leg postures. Once our system locates the human object, the macro motion analysis may identify the possible postures by analyzing the two projection profiles. The *macro motion analysis* recognizes the possible postures, whereas the *micro motion analysis* finds the BAPs of the human object. The macro motion analysis differentiates (1) bending posture, (2) arm postures, (3) over-head arm-raising postures, and (4) leg postures. We find that some transitions between two arm postures are not allowed. We may define all the possible transitions between two arm postures. We assume the variation of joint angles on the *x*-*y* plane or *y*-*z* plane is linear so that we can apply the micro motion analysis to track the movement of the arm in terms of the variations of joint angles. If we know the initial and terminating postures, we can decide the motion plane and find out

how the arm moves. We consider the arm-raising postures and other postures to construct the posture transition map (Figure 2), in which we show all allowable transitions between every two postures. Based on the map, the micro motion analysis may tracked arm motion effectively by using Kalman filter.

To track the movement between the transition of two identified postures we apply the Kalman filtering which consists of two groups: *time update equations* and *measure update equations*. The time update equations are responsible for predicting the current state and error covariance estimates to obtain a priori estimates for the next time step, *i.e.,* $\boldsymbol{x}_{k+1} = \mathbf{F}_{k+1,k}\boldsymbol{x}_k + \boldsymbol{w}_k$, where $\mathbf{F}_{k+1,k}$ is the transition matrix taking the state $\boldsymbol{x}_k$ from time k to time $k+1$. The measurement update equations are responsible for the feedback, *i.e.*, for incorporating a new measurement into a prior estimate to obtain an improved posteriori estimate such as $\boldsymbol{y}_k = \mathbf{H}_k\boldsymbol{x}_k + \boldsymbol{v}_k$, where $\mathbf{y}_k$ is the observable at time k and $\mathbf{H}_k$ is the measurement matrix.

In the absence of any observed data at time $k=0$, we may choose the initial estimate of the state. The first task during the measurement update is to compute the Kalman gain, $\mathbf{G}_k$. The next step is to actually measure the process to obtain $\boldsymbol{y}_k$, and then to generate a posteriori state estimate by incorporating the measurement. The final step is to obtain a posteriori error covariance estimate. After each time and measurement update, the process is repeated with the previous a posteriori estimates used to predict the new a priori estimates.

We assume the state vector which represents the joint angles of the human is defined as $\boldsymbol{x}_i[k] = [\theta_i(x, k), \theta_i(y, k), \theta_i(z, k), \Delta\theta_i(x, k), \Delta\theta_i(y, k), \Delta\theta_i(z, k)]^T$, where, $\theta_i(x, k)$, $\theta_i(y, k)$, $\theta_i(z, k)$ represent the angles of the i-th joint in the k-th frame. $\Delta\theta_i(x, k)$, $\Delta\theta_i(y, k)$, $\Delta\theta_i(z, k)$ represent the small joint angle difference between two frames. In our system, we assume the initial value as $\Delta\theta_i(x,0)=\Delta\theta_i(y,0)=\Delta\theta_i(z,0)=5$. We use Kalman filter as a linear prediction-correction filter, of which our purpose is not to solve occlusion, but to predict and correct the joint angles in motion-plane.

To develop the most accurate prediction for the current frame, we construct $\mathbf{F}_{k,k-1}$ based on a simple motion model $\theta(i, k)=\theta(i, k-1)+a\Delta\theta(i, k-1)$ with $\Delta\theta(i, k)=b\Delta\theta(i, k-1)$ where $i = x, y, z$. We assume that if we know the current joint angle, then we can predict the next joint angle by adding a small angle. Meanwhile, we also correct the joint angle difference between two estimates. Using the above motion model, we can construct the transition matrix. After the training phase, a and b are determined and can be used in tracking phase. To observe every element in state vector, we let measurement matrix $\mathbf{H}$ be an identity matrix.

To estimate the state effectively, we need an accurate observation for state update. Because we apply frame-based estimation, we update the state frame by frame. In our system, we analyze the projection profile to find the true joint angle.

To evaluate the performance of our tracking process, we test the system by using 18 different human motion actions. Each one is performed by 12 different individuals. People with casual wear and no markers are instructed to perform 18 different actions. 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.

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. Every testing sequence, O, are evaluated by 18 HMMs. The likelihood of the observation sequences can be computed for each HMM as $P_i=log(P(O \mid \lambda_i))$, 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 performed by the human object in the test video sequence. In our experiments, the recognition rate of our system is above 92%. Our system fails in estimating the BAP of the human object making self-occlusion. So it can only track the human object doing certain pre-defined actions.

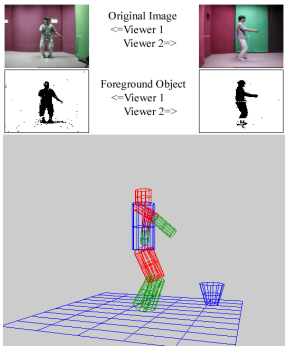


Figure 1. The motion capturing from two views.

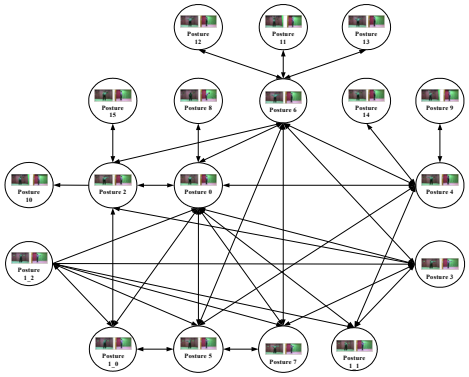


Figure 2. The transition map between two postures.