Vision-Based Tracking and Interpretation of Human Leg Movement for Virtual Reality Applications

Chin-Chun Chang and Wen-Hsiang Tsai, *Senior Member, IEEE,*

***Abstract*—A vision-based system for tracking and interpreting leg motion in image sequences using a single camera is developed for a user to control his movement in the virtual world by his legs. Twelve control commands are defined. The trajectories of the color marks placed on the shoes of the user are used to determine the types of leg movement by a first-order Markov process. Then, the types of leg movement are encoded symbolically as input to Mealy machines to recognize the control command associated with a sequence of leg movements. The proposed system is implemented on a commercial PC without any special hardware. Because the transition functions of Mealy machines are deterministic, the implementation of the proposed system is simple and the response time of the system is short. Experimental results with a 14-Hz frame rate on 320 $\times$ 240 image resolution are included to prove the feasibility of the proposed approach.**

***Index Terms*—Human–computer interaction, leg motion, Markov process, Mealy machine, virtual reality, vision-based tracking.**

I. Introduction

With the advance of computer vision technology, human–computer interaction devices become more and more friendly and intuitive for operation. For example, we can communicate with the computer using hand gestures [1], or play games by hand gestures or body postures [2]. For virtual reality applications, we can use some devices such as keyboards, wands, mice, treadmills, etc., to explore the virtual world [3]. However, these devices are either nonnatural for us, or expensive and heavy. To make the user immerse himself in the virtual world, a system providing a more intuitive way for a user to control his movement and action in the virtual world is needed. Since it is natural for us to express movement and action like walking, running, and jumping by using our legs, a vision-based system which tracks and interprets human leg motion in image sequences is developed to achieve this goal in this study. In the proposed system, vision is used because vision provides a noninvasive method for tracking, and thus the user and the system do not connect by cables. Since the proposed system provides another method for human–computer interaction, the system can be applied to exploring the virtual world, telerobotics, computer assisted instruction, video games, etc.

Research areas of vision-based human–computer interaction devices include facial expression recognition, head orientation detection, hand gesture recognition, gaze tracking, human body tracking, body posture recognition, etc. [4]. The proposed system focuses on how to recognize the type of leg motion and is related to the study of gait analysis and body posture recognition. To recognize the posture or the type of motion of a human, a popular approach is to use moving light displays (MLDs) [5] which are sequences of images of discrete points. In order to produce an MLD, the user is required to attach small glass bead reflectors to his joints while subject to intense light, or he must wear small light bulbs and stay in a dark room. To obtain the 3-D coordinates of MLDs, stereo vision must be used. Alternatively, by properly arranging the reflectors or bulbs, the 3-D coordinates of MLDs can be also obtained by using monocular vision [6], [7]. Although MLDs provide pure aspects of motion vision, the setup of such systems is not easy.

On the other hand, the posture of a human body can be recognized by analyzing the image of a human body. In [8], Rohr proposed a model-based approach to recognizing human movements in image sequences. A human body is represented by fourteen cylinders which are connected by joints. The image region of a walking person is obtained by a change detection algorithm and the model parameters are estimated by a discrete linear Kalman filter. Gavrila and Davis [9] employed a model with 22 degrees of freedom for a person, and estimated the parameters of the model by a generate-and-test strategy. In Ju *et al.* [10], human limbs are represented by a set of planar patches, and a parameterized model of optical flow of a planar patch is used to estimate articulated motion of human limbs. The above systems all need to estimate many parameters and require massive computational resources. In Wren *et al.* [11], a real-time human body tracker called Pfinder was developed and a blob representation was used to model a person. The system had been applied to gesture control and recognition of a subset of American Sign Language, etc. Freeman *et al.* [2] developed a vision-based system to recognize the posture of a human body for computer games. By extracting moment features and orientation histograms from the input image of the body, and then matching the extracted features to those of the prototypes of body postures, the closest match is regarded as the type of the body posture. However, due to only 2-D information being concerned, it is difficult for the systems to handle the variations of the leg motion (will be explained later).

In the proposed system, the user works within the field of view of a single camera and moves his legs according to some predefined rules. The system can continuously recognize the type of leg motion and execute the command associated with the type. For example, in this system, stepping on a spot means walking in the virtual world. The control commands provided in this system as well as some snapshots of the leg movement

Manuscript received November 5, 1997; revised May 10, 2000. This work was supported by National Science Council under Grant NSC-86-2213-E-009-114. This paper was recommended by Associate Editor E. Petajan.

The authors are with the Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan 300, R.O.C.

Publisher Item Identifier S 1051-8215(01)00666-8.

TABLE I
CONTROL COMMANDS

Control Command	Abbrev.	Description
Start system	<start>	Initially, the system does not recognize the leg motion of a user until he is ready and stands still with two feet close to each other for about three seconds. If the system does not start, the system (or the user) is in the *ready mode*; otherwise, the system (or the user) is in the *running mode*.
Stop system	<stop>	The user can cross his legs and keep this pose for about three seconds to stop the system and the system gets it into the *ready mode*.
Walk	<walk>	When the user walks on a spot, the system interprets this type of leg motion as walking.
Jump	<jump>	The user jumps up.
Change moving direction to forward	<forward>	When the user steps on a spot twice by his right leg, it means to change the moving direction to forward.
Change moving direction to backward	<backward>	When the user steps on a spot twice by his left leg, it means to change the moving direction to backward.
Move left	<move left>	The user moves his left leg sideways.
Move right	<move right>	The user moves his right leg sideways.
Turn left	<turn left>	The user moves his left leg foward.
Turn right	<turn right>	The user moves his right leg foward.
Left leg kick	<left kick>	The user kicks with his left leg.
Right leg kick	<right kick>	The user kicks with his right leg.

are listed in Table I and Fig. 1. Before introducing the proposed system, some characteristics of the leg motion in this system are identified as follows.

1) First, it is difficult to judge whether a leg is on the ground or not from a single image of the legs. Moreover, two trajectories of the leg motion of an identical command projected on the image plane may be different because the user may change his position and orientation. Therefore, directly analyzing a 2-D image to recognize the type of leg motion (like Freeman's system) is not feasible and the system must be able to handle such kinds of variation.
2) Second, since the leg motion is fast and the user has to keep himself balanced, it does not have significant "pause" between the leg motion associated with consecutive commands. Thus, it is not easy to indicate the starting and ending positions of the leg motion associated with a command and a method for continuous leg motion recognition is needed.
3) Third, different durations of the same type of leg motion may convey different meanings. For example, in the proposed system, to stop this system, the user can cross his legs and keeps this posture at least 3 s. However, the user crossing his legs does not always mean to stop this system because the user may cross his legs momentarily just because of moving sideways.

Because of using monocular vision, the trajectory of the leg on the image plane when the user moves his leg backward and behind his body is very similar to the trajectory of the leg when the user raises up the leg. Therefore, we do not design the type of leg motion which moves the leg backward and behind the

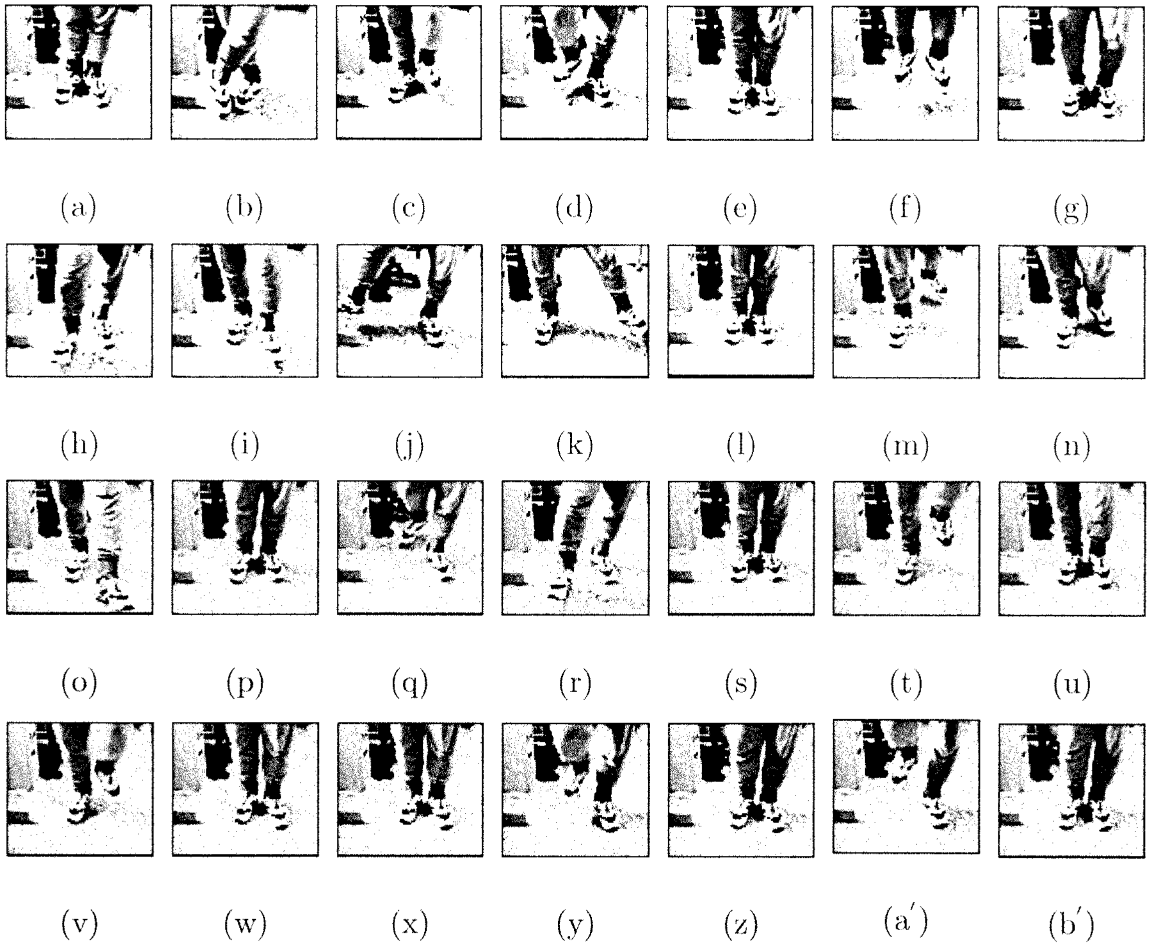


Fig. 1. Some snapshots of the leg motion corresponding to the control commands. (a) <start>. (b) <stop>. (c) and (d) <walk>. (e)–(g) <jump>. (h) <turn right>. (i) <turn left>. (j) <move right>. (k) <move left>. (l)–(r) <left kick>, and then <right kick>. (s)–(w) <backward>. (x)–(b′) <forward>.

user's body in this system. To obtain the region of the user's legs, some existing methods can be applied such as the active shape model [12], and the statistical shape model [13]. However, they are time-consuming to detect and track the legs because the motion of human legs in the NTSC video (30 frames/s) can be very large, and the change of the shapes of legs can be big due to self-occlusion, shadow effects, deformation of trousers, and the flexion of the user's legs. In order to ease the installation of this system in a general indoor environment, and the identification of the two legs of a user even if they are occluded or crossed, the user is required to wear two different color bands on his legs for use as the reference points.

In the past, the hidden Markov model (HMM) [1], dynamic time warping [14], and the finite state machine [15] have been used to recognize the hand gesture. Alternatively, in this study, the leg motion is regarded as a sequence of state transitions and modeled by a first-order Markov process. In addition, in order to continuously recognize the meaning of a sequence of leg states, two Mealy machines [16] are implemented. In the following is the organization of this paper. In Section II, an overview of the system is described. In Section III, the definitions of leg states are described and the method to determine the current leg state using a first-order Markov process is introduced. In Section IV, two Mealy machines proposed to understand the meaning associated with a sequence of leg motion are defined. In Section V, image processing techniques used to identify the color band worn on the user's legs is presented. Some experimental results are shown in Section VI followed by concluding remarks.

II. System Overview

To use the system, the user stands in front of a camera with two color bands on his feet. Shown in Fig. 2 is a configuration of the system. Initially, the system is in the *ready mode*. In this mode, the system is waiting for the user to enter into the field of view of the camera and issue the command <start> (defined in Table I) to start the system to interpret the user's leg motion. When the system is started, the system is in the *running mode* and the user can issue the commands listed in Table I. The current state of the user's legs is determined by a first-order Markov process. In addition, a Mealy machine $\mathbf{M}_1$ is developed to output the command associated with the sequence of the leg states. To handle the problem that different durations of the same type of leg motion may convey different meanings, the other Mealy machine $\mathbf{M}_2$ is designed to translate the leg states to the input symbols of $\mathbf{M}_1$. Specifically, the relationship among the Markov process, $\mathbf{M}_1$, and $\mathbf{M}_2$ is that the current state of the user's legs is determined by the Markov process, and served as an input symbol of $\mathbf{M}_2$, then $\mathbf{M}_2$ translates the current state

TABLE II
DEFINITIONS OF LEG STATES IN THE RUNNING MODE

State no.	Abbrev.	Description
1	<s,s>	Both feet are stationary and on the ground. Fig. 1 (a) shows an example.
2	<v,v>	Jump, i. e., the user raises or lowers the left leg and the right leg simutaneously. In this study, we do not allow the user to jump sideways, forward, or backward because the user may easily jump out of the camera view. Fig. 1 (f) is an example.
3	<v,s>	The left leg is raised or lowered and the right foot is stationary and on the ground. Fig. 1 (c) is an example.
4	<cv,s>	The left leg is raised or lowered and the right foot is stationary and on the ground; in addition, the two legs are crossed.
5	<h,s>	The left leg is moved sideways and the right foot is stationary and on the ground. Fig. 1 (k) is an example.
6	<ch,s>	The left leg is moved sideways and the right foot is stationary and on the ground; in addition, the two legs are crossed. Fig. 1 (b) is an example.
7	<f,s>	The left leg is moved forward and the right foot is stationary and on the ground. Fig. 1 (i) shows an example.
8	<cf,s>	The left leg is moved forward and the right foot is stationary and on the ground; in addition, the two legs are crossed.
9-14	State 9: <s,v>, State 10: <s,cv>, State 11: <s,h>, State 12: <s,ch>, State 13: <s,f>, and State 14: <s,cf> are defined similarly for the states when the left foot is stationary and on the ground, and the right leg is in motion.	

to the input symbol of $\mathbf{M}_1$, and $\mathbf{M}_1$ outputs a symbol which represents the command issued. When the system is in the *running mode*, the above process is repeated until the user issues the command <stop> to stop the system, and then the system goes to the *ready mode* to wait for the user to start this system again. In the following sections, the Markov process $\mathbf{M}_1$, and $\mathbf{M}_2$ will be described in detail.

III. DETERMINATION OF LEG STATES

Leg motion can be regarded as being composed by a sequence of leg states. Thus, to understand the meaning of leg motion, it is necessary to know the current state of the legs. When the system is in the *ready mode*, we only need to know whether the user stands still on the ground or not; however, it becomes more complicated when the system is in the *running mode*. In this section, 16 leg states are defined; two are defined for use in the *ready mode*, and the rest for use in the *running mode*. We first introduce the method to determine the leg state using a first-order Markov process when the system is in the *running mode*, and then the technique to determine the leg state when the system is in the *ready mode*.

A. Definitions of Leg States in the Running Mode

In the *running mode*, fourteen leg states are defined according to the snapshots of the configurations of the legs. For example, two legs may be still or above the ground (i.e., the user jumps up); or one leg may be still, and the other leg above the ground, or moved sideways. The detailed definitions of the fourteen leg states are listed in Table II. Other types of leg states can also be defined, but the 14 states are found to be enough in this study.

As described in Table II, the configurations of a leg may be above the ground, moved left or right, or moved forward, and these configurations are identified as in a "vertical" position, in a "sideways" position, and in a "forward" position, respectively. If the position of a leg is exactly "vertical," "sideways," or "forward," we can connect the current position of the mark on the leg to the last position of the mark which is just on the ground, and form three directional vectors which are useful for determining the configurations of the leg. For example, if the direction from the last position of the mark which is on the ground to the current position of the mark is similar to the "vertical" direction, then it will be more possible to say that the current configuration of the leg is in a "vertical" position. Specifically, the directional vectors can be defined by the linear combination of three orthonormal vectors in the camera coordinate system: the unit normal vector $\mathbf{e}_g$ of the ground plane, the unit normal vector $\mathbf{e}_f$ of the frontal plane which is orthogonal to the ground plane and passing through the marks on the legs, and the cross product $\mathbf{e}_s = \mathbf{e}_g \times \mathbf{e}_f$ of the two vectors. The direction of $\mathbf{e}_g$ is away from the ground plane and the direction of $\mathbf{e}_f$ is toward the camera. Furthermore, $\mathbf{e}_f$ can be changed if the user turns his body. In this system, the directional vector $\mathbf{e}_f$ is regarded to be the walking direction of the user with respect to the camera coordinate system. After observing the leg motion of a human being stepping on a spot, we define the three directional unit vectors for the left leg as follows:

$$\begin{aligned} \text{vertical:}\quad & \mathbf{e}_g; \\ \text{sideways:}\quad & \mathbf{e}_g \sin 15^\circ + \mathbf{e}_s \cos 15^\circ \\ \text{forward:}\quad & \mathbf{e}_g \cos 69.2952^\circ + \mathbf{e}_s \cos 69.2952^\circ + \mathbf{e}_f \cos 30^\circ \end{aligned}$$

and three directional unit vectors for the right leg as follows:

$$\begin{aligned} \text{vertical:}\quad & \mathbf{e}_g \\ \text{sideways:}\quad & \mathbf{e}_g \sin 15^\circ - \mathbf{e}_s \cos 15^\circ \\ \text{forward:}\quad & \mathbf{e}_g \cos 69.2952^\circ - \mathbf{e}_s \cos 69.2952^\circ + \mathbf{e}_f \cos 30^\circ. \end{aligned}$$

Fig. 2 is an illustration of the geometric configuration of the ground plane, the frontal plane, and the six directional vectors. How to estimate $\mathbf{e}_g$ and $\mathbf{e}_f$ will be described later.

From a single camera, it has some difficulty to know the 3-D positions of the marks when the user's legs are not on the ground; in addition, when the user stops moving one of his legs, we cannot know whether this foot arrives at the ground or not until the other leg moves. On the other hand, the 3-D positions of the marks can be easily computed if we know that the user's feet are on the ground. In addition, when the user moves one of his legs, the other leg must be stationary and on the ground except when he jumps. Thus, in this study, once we are sure that a leg is stationary and on the ground, said to be at an *initial position*, the 2-D trajectories which are the perspective projections of the mark moving in the three directions can be computed immediately to determine the leg state.

Let the ground plane equation with respect to the camera coordinate system be $ax+by+cz+d=0$ with $a^2+b^2+c^2=1$ and $c \le 0$, which can be estimated via a flat calibration mark on the ground [17]. Obviously, $\mathbf{e}_g = [a\ b\ c]^t$. Let the last initial positions of the left mark and the right mark projected on the image plane be $\mathbf{l} = [x_l\ y_l]^t$ and $\mathbf{r} = [x_r\ y_r]^t$, respectively.

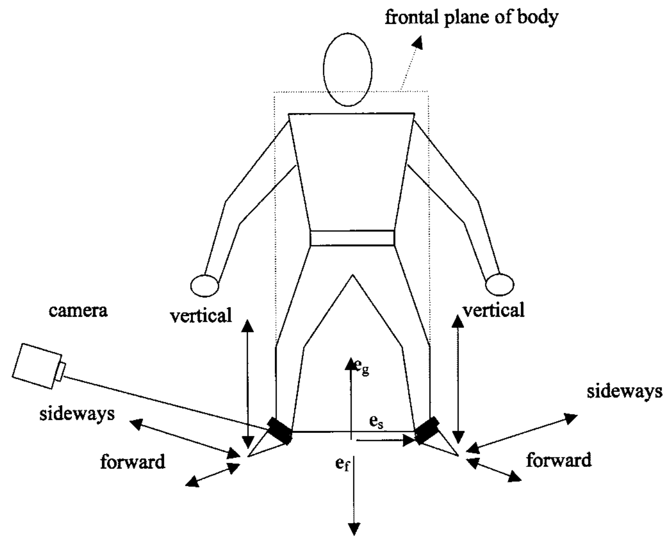


Fig. 2. An illustration of the geometric configuration of the ground plane, the frontal plane, and the six directional vectors.

Then, in the camera coordinate system, the 3-D position of the mark at the initial position can be easily computed by

$$\mathbf{p} = \frac{-d}{ax+by+cf}\,[x \quad y \quad f]^t$$

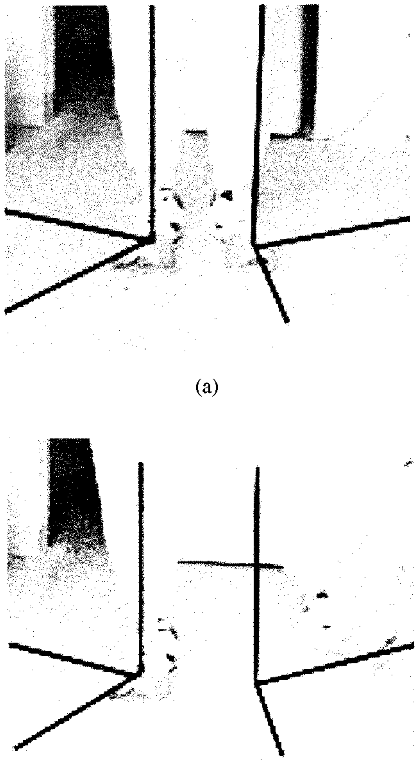
where $\mathbf{p}$ and $\mathbf{x} = [x\ y]^t$ are the 3-D coordinate and the perspective projection of either the left mark or the right mark at the initial position and f is the focal length. Since $\mathbf{e}_f$ is orthonormal to the normal vector of the ground plane and the initial positions of the marks are on the frontal plane, the unit normal vector of the frontal plane $\mathbf{e}_f$ can be obtained by normalizing the cross product of $\mathbf{e}_g$ and the vector passing through the two initial positions

$$\mathbf{e}_f = \frac{\mathbf{e}_g \times (\mathbf{p}_r - \mathbf{p}_l)}{|\mathbf{p}_r - \mathbf{p}_l|}$$

where $|\cdot|$ represents the 2-norm of a vector, $\mathbf{p}_l$ and $\mathbf{p}_r$ represent the 3-D coordinates of the left mark and the right marks at the initial positions, respectively. In addition, the trajectory, which is the perspective projection of a mark moving from an initial position in the direction $\mathbf{d} = [d_x\ d_y\ d_z]^t$, is a uni-directional line starting from the perspective projection $\mathbf{x}$ of the mark at the initial position, and can be expressed as follows:

$$\mathbf{ld} = \frac{\mu}{\sqrt{(fd_x - xd_z)^2 + (fd_y - yd_z)^2}} \begin{bmatrix} fd_x - xd_z \\ fd_y - yd_z \end{bmatrix} + \mathbf{x} \tag{1}$$

where $\mathbf{ld}$ represents the direction of the trajectory and $\mu \ge 0$. Using (1), the directions of the directional vectors projected on the image plane can be computed. Let $\mathbf{ld}_v$, $\mathbf{ld}_s$, and $\mathbf{ld}_f$ denote the directions of the trajectories of the left mark moving up or down, sideways, and forward, respectively, and let $\mathbf{rd}_v$, $\mathbf{rd}_s$, and $\mathbf{rd}_f$ be defined similarly for the right mark. In this study, $\mathbf{e}_s, \mathbf{e}_f$, the six vectors, and the last two initial positions of the left mark and the right mark are called the *state parameters*. Fig. 3(a) shows a real image and a configuration of the six directional vectors; Fig. 3(b) shows that the left mark moves along $\mathbf{ld}_s$ when the user moves his left leg sideways.



(a)

(b)

Fig. 3. An illustration of the trajectories of the six directional vectors. (a) Configuration of the six directional vectors. (b) The user moves his left leg sideways and the trajectory of the left mark is along $\mathbf{ld}_s$.

In practice, the movement of a leg is not always vivid and the moving direction of a leg may not align well with the three directional vectors. Thus, to correctly determine the current leg state, we must take the past states of the legs into account because consecutive leg states are highly correlated.

B. Determination of Leg State in Running Mode by First-Order Markov Process

According to our experience, some of the leg states are highly correlated to their previous states. For example, when walking on a spot, one may first raise his right leg up (going into the state <s,v>), then step down with his right leg (remaining in the state <s,v>) and raise his left leg up (going into the state <v,s>) immediately, and then step down with his left leg (remaining in the state <v,s>), and repeat this process again and again. In this study, a first-order Markov process is used to describe the process of transitions among the 14 leg states.

Let s_t denote the leg state at time t, $P(j|i)$ denote the transition probability from the state i to the state j, and $\mathbf{l}_t = [lx_t \; ly_t]$ and $\mathbf{r}_t = [rx_t \; ry_t]$ denote, respectively, the positions of the marks of the left and right legs at time t in the image plane. Our everyday experience tells us that if we know the initial positions of the marks and the current leg state, we can easily figure out the possible positions of the marks. That is, the positions of the marks $\mathbf{l}_t$ and $\mathbf{r}_t$ at time t are highly dependent on their last initial positions $\mathbf{l}_{t-m}$ and $\mathbf{r}_{t-n}$ and the current leg state. Thus, the conditional probability distribution

$$P(\mathbf{l}_t, \mathbf{r}_t | s_t = j, \mathbf{l}_{t-1}, \mathbf{l}_{t-2}, \ldots, \mathbf{l}_0, \mathbf{r}_{t-1}, \mathbf{r}_{t-2}, \ldots, \mathbf{r}_0)$$

can be reduced to

$$\begin{aligned} P\left(\mathbf{l}_t, \mathbf{r}_t \,\middle|\, \begin{array}{c} s_t = j, \mathbf{l}_{t-1}, \mathbf{l}_{t-2}, \ldots, \mathbf{l}_0, \\ \mathbf{r}_{t-1}, \mathbf{r}_{t-2}, \ldots, \mathbf{r}_0 \end{array}\right) \\ = P(\mathbf{l}_t, \mathbf{r}_t | s_t = j, \mathbf{l}_{t-m}, \mathbf{r}_{t-n}). \end{aligned}$$

Accordingly, the *a posteriori* probability that the leg state at time t is i can be expressed as in (2), shown at the bottom of the page [18], , where the *a priori* probability for the leg state at time t is given by

$$\begin{aligned} &P(s_t = i | \mathbf{l}_{t-1}, \mathbf{l}_{t-2}, \ldots, \mathbf{l}_0, \mathbf{r}_{t-1}, \mathbf{r}_{t-2}, \ldots, \mathbf{r}_0) \\ &\quad = \sum_{j=1}^{14} P(i|j) P(s_{t-1} = j | \mathbf{l}_{t-1,} \mathbf{l}_{t-2}, \ldots, \mathbf{l}_0, \mathbf{r}_{t-1} \\ &\qquad \mathbf{r}_{t-2}, \ldots, \mathbf{r}_0). \end{aligned} \tag{3}$$

When the system changes the mode from ready to running, the user must stand still and the leg state must be in the state <s,s>. Therefore, t is initialized to be one and the *a posteriori* probabilities of the leg states at time zero are $P(s_0 = 1 | \mathbf{l}_0, \mathbf{r}_0) = 1$, and $P(s_0 = i | \mathbf{l}_0, \mathbf{r}_0) = 0$, for $i = 2, 3, \ldots, 14$. In addition, the 14 leg states can be classified into four disjoint sets by examining which leg is in motion. The four sets are described as follows: 1) set 1: {state 1}—both feet are stationary and on the ground; 2) set 2: {state 2}—both legs are moving up or down; 3) set 3: {state 3, state 4, ..., state 8}—the left leg is in motion and the right foot is stationary and on the ground; 4) set 4: {state 9, state 10, ..., state 14}—the right leg is in motion and the left foot is stationary and on the ground. And the *a posteriori* probabilities $P_1(t)$, $P_2(t)$, $P_3(t)$, and $P_4(t)$ of the four sets can be defined, respectively, as follows:

$$\begin{aligned} P_1(t) &= P(s_t = 1 | \mathbf{l}_t, \mathbf{l}_{t-m}, \mathbf{r}_t, \mathbf{r}_{t-n}) \\ P_2(t) &= P(s_t = 2 | \mathbf{l}_t, \mathbf{l}_{t-m}, \mathbf{r}_t, \mathbf{r}_{t-n}) \\ P_3(t) &= \sum_{i=3}^{8} P(s_t = i | \mathbf{l}_t, \mathbf{l}_{t-m}, \mathbf{r}_t, \mathbf{r}_{t-n}) \\ P_4(t) &= \sum_{i=9}^{14} P(s_t = i | \mathbf{l}_t, \mathbf{l}_{t-m}, \mathbf{r}_t, \mathbf{r}_{t-n}). \end{aligned}$$

$$P(s_t = i | \mathbf{l}_t, \mathbf{l}_{t-1}, \ldots, \mathbf{l}_0, \mathbf{r}_t, \mathbf{r}_{t-1}, \ldots, \mathbf{r}_0) = \frac{P(s_t = i | \mathbf{l}_{t-1}, \ldots, \mathbf{l}_0, \mathbf{r}_{t-1}, \ldots, \mathbf{r}_0) P(\mathbf{l}_t, \mathbf{r}_t | s_t = i, \mathbf{l}_{t-m}, \mathbf{r}_{t-n})}{\sum_{j=1}^{14} P(s_t = j | \mathbf{l}_{t-1}, \ldots, \mathbf{l}_0, \mathbf{r}_{t-1}, \ldots, \mathbf{r}_0) P(\mathbf{l}_t, \mathbf{r}_t | s_t = j, \mathbf{l}_{t-m}, \mathbf{r}_{t-n})} \tag{2}$$

Thus, when the system is in the *running mode*, the leg state s_t at time t can be determined by (4) which selects the leg state with the largest *a posteriori* probability from the set whose *a posteriori* probability is the largest among $P_1(t)$, $P_2(t)$, $P_3(t)$, and $P_4(t)$

$$s_t = \begin{cases} 1 & \text{if } P_1(t) = \max_{j=1,2,3,4}\{P_j(t)\}; \\ 2 & \text{if } P_2(t) = \max_{j=1,2,3,4}\{P_j(t)\}; \\ i & \text{if } P_3(t) = \max_{j=1,2,3,4}\{P_j(t)\} \text{ and} \\ & P\left(s_t = i \,\middle|\, \begin{matrix}\mathbf{l}_t, \mathbf{l}_{t-m}, \\ \mathbf{r}_t, \mathbf{r}_{t-n}\end{matrix}\right) \\ & \quad = \max_{j=3,4,\ldots,8}\left\{P\left(s_t = j \,\middle|\, \begin{matrix}\mathbf{l}_t, \mathbf{l}_{t-m}, \\ \mathbf{r}_t, \mathbf{r}_{t-n}\end{matrix}\right)\right\}, \text{ or} \\ & \text{if } P_4(t) = \max_{j=1,2,3,4}\{P_j(t)\} \text{ and} \\ & P\left(s_t = i \,\middle|\, \begin{matrix}\mathbf{l}_t, \mathbf{l}_{t-m}, \\ \mathbf{r}_t, \mathbf{r}_{t-n}\end{matrix}\right) \\ & \quad = \max_{j=9,10,\ldots,14}\left\{P\left(s_t = j \,\middle|\, \begin{matrix}\mathbf{l}_t, \mathbf{l}_{t-m}, \\ \mathbf{r}_t, \mathbf{r}_{t-n}\end{matrix}\right)\right\}. \end{cases} \tag{4}$$

However, determining the leg state s_t just by (4) may lead to serious mistake because the *a posteriori* probability of the selected leg state may be not large enough to dominate the *a posteriori* probabilities of the other leg states. Thus, when the *a posteriori* probability of the selected leg state is not large enough (smaller than 0.5, in this study), an additional leg state—"undetermined" is introduced to postpone making decision at time t.

C. Definitions of Conditional Probability Distributions

To compute the *a posteriori* probability of the leg state at time t, we must know the conditional probability distributions $P(\mathbf{l}_t, \mathbf{r}_t | s_t = i, \mathbf{l}_{t-m}, \mathbf{r}_{t-n})$, $i = 1, 2, \ldots, 14$, and the transition probabilities $P(j|i)$, $i, j = 1, 2, \ldots, 14$. In this section, the conditional probability distributions will be defined, and a method to obtain the transition probability table will be introduced in the next section. First, some auxiliary functions are defined as follows.

1) The probability distribution function $P_n(\text{dist}(\mathbf{x}, \mathbf{x}'))$ of two points $\mathbf{x}$ and $\mathbf{x}'$ being nonstationary and its complement $P_s(\text{dist}(\mathbf{x}, \mathbf{x}'))$ are defined as follows:

$$\begin{aligned} P_n(\text{dist}(\mathbf{x}, \mathbf{x}')) &= \frac{1}{\Gamma(\alpha)} \int_0^{\text{dist}(\mathbf{x}-\mathbf{x}')} \exp(-t) t^{a-1}\, dt \\ P_s(\text{dist}(\mathbf{x}, \mathbf{x}')) &= 1 - P_s(\text{dist}(\mathbf{x}, \mathbf{x}')) \end{aligned} \tag{5}$$

where $\text{dist}(\mathbf{x}, \mathbf{x}')$ represents the Euclidean distance between $\mathbf{x}$ and $\mathbf{x}'$, $\Gamma(a)$ is the gamma function

$$\Gamma(\alpha) = \int_0^\infty \exp(-t) t^{\alpha-1}\, dt$$

$P_n(0) = 0$, $P_n(\infty) = 1$, and α is used to control the sensitivity for detecting point movement. Thus, at time t, the probability that the legs are both stationary (but not necessary on the ground) is

$$P_{\text{stationary}}(t) = P_s(\text{dist}(\mathbf{l}_t, \mathbf{l}_{t-1})) P_s(\text{dist}(\mathbf{r}_t, \mathbf{r}_{t-1})).$$

2) $P_d(\mathbf{v}, \mathbf{x}, \mathbf{x}')$ defined in the following is the probability distribution function of the direction of two points $\mathbf{x}$ and $\mathbf{x}'$ consistent with the direction of the vector $\mathbf{v}$

$$P_d(\mathbf{v}, \mathbf{x}, \mathbf{x}') = \begin{cases} \dfrac{\mathbf{v} \cdot (\mathbf{x} - \mathbf{x}')}{|\mathbf{v}|\,|\mathbf{x} - \mathbf{x}'|}, & \text{if } \mathbf{v} \cdot (\mathbf{x} - \mathbf{x}') \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

where $\cdot$ represents the inner product of two vectors. In addition, $P_{\text{bid}}(\mathbf{v}, \mathbf{x}, \mathbf{x}')$ is similarly defined but takes additionally the reverse direction of $\mathbf{v}$ into account

$$P_{\text{bid}}(\mathbf{v}, \mathbf{x}, \mathbf{x}') = P_d(\mathbf{v}, \mathbf{x}, \mathbf{x}') + P_d(-\mathbf{v}, \mathbf{x}, \mathbf{x}').$$

3) A function τ whose output is zero or one is defined as follows:

$$\tau(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{otherwise.} \end{cases}$$

4) If the legs are crossed, the left mark must have been moved to the right side of the right mark, or the right mark must have been moved to the left side of the left mark. These relations can be expressed as follows:

$$\text{leg relation} = \begin{cases} \text{crossed}, & \text{if } 1 - \tau(P_d(\mathbf{ld}_s, \mathbf{l}_t, \mathbf{r}_t)) = 1 \text{ or} \\ & \quad 1 - \tau(P_d(\mathbf{rd}_s, \mathbf{r}_t, \mathbf{l}_t)) = 1 \\ \text{not crossed}, & \text{if } \tau(P_d(\mathbf{ld}_s, \mathbf{l}_t, \mathbf{r}_t)) = 1 \text{ or} \\ & \quad \tau(P_d(\mathbf{rd}_s, \mathbf{r}_t, \mathbf{l}_t)) = 1. \end{cases}$$

As mentioned previously, when the user stops moving one of his legs, we cannot know whether the foot arrives at the ground or not. Thus, in this study, it is taken as a principle that when a leg stops moving, the leg state is not changed until the leg moves in another type of motion or another leg moves. With this principle and the above auxiliary functions, the 14 conditional probability distributions can be defined. Here, the conditional probability $P(\mathbf{l}_t, \mathbf{r}_t | s_t = 3, \mathbf{l}_{t-m}, \mathbf{r}_{t-n})$ is used to illustrate how to define the conditional probabilities. State 3 means that the left leg is in a "vertical" position, the right leg is stationary and on the ground, and the two legs are not crossed. Thus, $P(\mathbf{l}_t, \mathbf{r}_t | s_t = 3, \mathbf{l}_{t-m}, \mathbf{r}_{t-n})$ can be expressed by the product of the four functions: $P_n(\text{dist}(\mathbf{l}_t, \mathbf{l}_{t-m}))$, $P_d(\mathbf{ld}_v, \mathbf{l}_t, \mathbf{l}_{t-m})$, $P_s(\text{dist}(\mathbf{r}_t, \mathbf{r}_{t-m}))$, and $\tau(P_d(\mathbf{ld}_s, \mathbf{l}_t, \mathbf{r}_t))$, where the product of the first two functions represents the probability of the left leg in a "vertical" position, the third function denotes the probability that the right leg is stationary and on the ground, and the fourth function states the probability of the two legs not being crossed. The definitions of the other conditional probabilities are given in Table III and their derivations are omitted.

TABLE III
DEFINITIONS OF CONDITIONAL PROBABILITIES

State no. s_t	Conditional probability $P(\mathbf{l}_t, \mathbf{r}_t \mid s_t = i, \mathbf{l}_{t-m}, \mathbf{r}_{t-n})$
1	$P_s(\mathrm{dist}(\mathbf{l}_t, \mathbf{l}_{t-m}))P_s(\mathrm{dist}(\mathbf{r}_t, \mathbf{r}_{t-n}))$
2	$P_n(\mathrm{dist}(\mathbf{l}_t, \mathbf{l}_{t-m}))P_d(\mathbf{ld}_v, \mathbf{l}_t, \mathbf{l}_{t-m})P_n(\mathrm{dist}(\mathbf{r}_t, \mathbf{r}_{t-n}))P_d(\mathbf{rd}_v, \mathbf{r}_t, \mathbf{r}_{t-n})$
3	$P_n(\mathrm{dist}(\mathbf{l}_t, \mathbf{l}_{t-m}))P_d(\mathbf{ld}_v, \mathbf{l}_t, \mathbf{l}_{t-m})\tau(P_d(\mathbf{ld}_s, \mathbf{l}_t, \mathbf{r}_t))P_s(\mathrm{dist}(\mathbf{r}_t, \mathbf{r}_{t-n}))$
4	$P_n(\mathrm{dist}(\mathbf{l}_t, \mathbf{l}_{t-m}))P_d(\mathbf{rd}_v, \mathbf{l}_t, \mathbf{r}_{t-m})(1 - \tau(P_d(\mathbf{ld}_s, \mathbf{l}_t, \mathbf{r}_t)))P_s(\mathrm{dist}(\mathbf{r}_t, \mathbf{r}_{t-n}))$
5	$P_n(\mathrm{dist}(\mathbf{l}_t, \mathbf{l}_{t-m}))P_d(\mathbf{ld}_s, \mathbf{l}_t, \mathbf{l}_{t-m})\tau(P_d(\mathbf{ld}_s, \mathbf{l}_t, \mathbf{r}_t))P_s(\mathrm{dist}(\mathbf{r}_t, \mathbf{r}_{t-n}))$
6	$P_n(\mathrm{dist}(\mathbf{l}_t, \mathbf{l}_{t-m}))P_{bid}(\mathbf{rd}_s, \mathbf{l}_t, \mathbf{r}_{t-m})(1 - \tau(P_d(\mathbf{ld}_s, \mathbf{l}_t, \mathbf{r}_t)))P_s(\mathrm{dist}(\mathbf{r}_t, \mathbf{r}_{t-n}))$
7	$P_n(\mathrm{dist}(\mathbf{l}_t, \mathbf{l}_{t-m}))P_d(\mathbf{ld}_f, \mathbf{l}_t, \mathbf{l}_{t-m})\tau(P_d(\mathbf{ld}_s, \mathbf{l}_t, \mathbf{r}_t))P_s(\mathrm{dist}(\mathbf{r}_t, \mathbf{r}_{t-n}))$
8	$P_n(\mathrm{dist}(\mathbf{l}_t, \mathbf{l}_{t-m}))P_d(\mathbf{rd}_f, \mathbf{l}_t, \mathbf{r}_{t-m})(1 - \tau(P_d(\mathbf{ld}_s, \mathbf{l}_t, \mathbf{r}_t)))P_s(\mathrm{dist}(\mathbf{r}_t, \mathbf{r}_{t-n}))$
9-14	The conditional probability distributions of State 9 to State 14 can be defined in the same ways as State 3 to State 8.

D. Learning Probability Transition Table

When a user moves his legs freely, he can change his leg state from one of the fourteen states to any one of them. That is, all of the elements of the probability transition table are not zero. However, some of the state transitions for the user are easy to conduct but some of them are difficult for the user. In general, the frequency for a user to do a simple leg motion is higher than the frequency of a difficult one. On the other hand, if the user always moves his legs in order to issue the defined commands, then some state transitions will never occur (for example, changing the leg state from the fourth state to the twelfth state) and the probabilities of the state transitions will be zero. As a consequence, the actions of a user can be classified to two kinds: the first kind is to issue some control commands, and the other kind of actions is meaningless movement, and thus, no control command is associated with the actions. Because the user may perform some meaningless actions in the system, the probability transition table must be able to describe the two different kinds of behaviors of a user. In this study, the transition probability from the state i to the state j is defined by a weighted sum of two terms

$$P(j|i) = \beta P_1(j|i) + (1-\beta)P_2(j|i) \tag{6}$$

where P_1 and P_2, respectively, correspond to the probability transition tables of the first and the second kinds of actions and $0 \le \beta \le 1$. Thus, if β is equal to one, then only the state transitions related to the control commands are allowed. If β is equal to zero, then the user is regarded to have no particular intent to issue the control commands.

In this study, the transition probabilities $P_1(j|i)$, i, $j = 1, 2, \ldots, 14$, are computed by Bayes's rule

$$P_1(j|i) = \frac{P_1(\text{current state} = i,\ \text{next state} = j)}{P_1(i)} \quad \text{for } i, j = 1, 2, \ldots, 14$$

where the definitions of P_1(current state $= i$, next state = j) and $P_1(i)$ are given as follows:

$$P_1\begin{pmatrix}\text{current state} = i,\\ \text{next state} = j\end{pmatrix} = \frac{\text{the number of occurrence of the state } i \text{ followed by the state } j}{\text{the number of training frames} - 1}$$

$$P_1(i) = \frac{\text{the number of occurrence of the state } i}{\text{the number of training frames}} \tag{7}$$

and P_1(current state = i, next state = j) and $P_1(i)$, i, $j = 1, 2, \ldots, 14$ can be obtained by a training procedure. The transition probabilities $P_2(j|i)$, i, $j = 1, 2, \ldots, 14$, are assigned manually according to the level of difficulty for changing the leg state from the state i to the state j. In this study, five difficulty levels $0, 1, \ldots, 4$, ranked from the most difficult to the simplest level, are defined. Suppose that the probabilities of the transition from the state i with the difficulty level g and the transition from the state i with the difficulty level $g+1$ are related by a factor γ, where $\gamma \ge 1$. Accordingly, the transition probabilities $P_2(j|i)$, $i, j = 1, 2, \ldots, 14$ can be computed by

$$P_2(j|i) = \frac{\gamma^{\mathbf{DL}(j|i)}}{\sum_{m=1}^{14} \gamma^{\mathbf{DL}(m|i)}}$$

where $\mathbf{DL}(j|i)$ denotes the level of difficulty for the transition from the state i to the state j. The difficulty levels of state transitions defined in this study are listed in Table IV.

E. Determination of Leg State in Ready Mode

When the system is in the *ready mode*, we only need to know whether the user tries to start the system by standing still with two feet close to each other. Therefore, there are only two leg states in the *ready mode:* the state that the user stands still with

TABLE IV
DIFFICULTY LEVELS DEFINED IN THIS STUDY

		j													
	$\mathbf{DL}(j\mid i)$	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
	1	0	1	1	3	1	3	1	3	1	3	1	3	1	3
	2	1	0	3	4	3	4	3	4	3	4	3	4	3	4
	3	1	3	0	2	2	3	1	3	1	4	2	4	2	4
	4	1	3	1	0	2	3	2	3	1	4	2	4	3	4
	5	1	3	2	2	0	1	2	3	2	4	1	4	2	4
	6	1	3	3	3	1	0	2	3	2	3	2	3	2	4
i	**7**	1	3	1	2	2	2	0	2	2	4	2	4	1	4
	8	1	3	2	3	2	3	1	0	4	4	2	3	2	4
	9	1	3	1	4	2	4	2	4	0	2	2	3	1	3
	10	1	3	1	4	2	4	3	4	1	0	2	3	2	3
	11	1	3	2	4	1	4	2	4	2	2	0	1	2	3
	12	1	3	2	3	2	3	2	4	3	3	1	0	2	3
	13	1	3	2	4	2	4	1	4	1	2	2	2	0	2
	14	1	3	4	4	2	3	2	4	2	3	2	3	1	0

two feet close enough and its complement. Although, at time t, the probability of two legs being stationary can be known by $P_{\text{stationary}}(t)$, this probability measure cannot tell us whether the user's two feet stand on the ground or not. Thus, it is inadequate to determine whether the user starts the system or not by just using $P_{\text{stationary}}(t)$. For example, if the user raises up his left leg and keeps stationary about 3 s, then the system will be started because $P_{\text{stationary}}(t)$ has a high value, and the leg state will be erroneously recognized to be <s,s> and the six directional vectors will be computed with a great amount of error, leading to incorrect determination of subsequent leg states. In this study, when the system is in the *ready mode*, if the user keeps stationary for about 3 s, we assume that the two feet are both on the ground and the 3-D positions of the marks on the legs can be computed. If the distance between the two marks is shorter than a predefined threshold value, the two feet are regarded to be on the ground. This simple rule works well because if a foot does not stand on the ground, the computed 3-D position will be far away from the actual position and the computed distance will be much larger than the actual distance between the two marks. Consequently, the rule to determine the leg state in the *ready mode* is as follows:

$$s_t = \begin{cases} 15, & \text{if } P_{\text{stationary}}(t)\tau(P_d([1 \quad 0]^t, \mathbf{l}_t, \mathbf{r}_t)) > 0.5 \text{ and} \\ & \quad \text{the 3-D distance between the two marks} \\ & \quad \text{is smaller than a predefined threshold} \\ 16, & \text{otherwise.} \end{cases} \tag{8}$$

IV. DETERMINATION OF CONTROL COMMANDS

To continuously recognize the meaning of a sequence of leg movement, Mealy machines are used in this study. A Mealy machine $\mathbf{M}$ is a sequential machine which can be described by a six-tuple $\mathbf{M} = (Q, \Sigma, \Delta, \delta, \lambda, q_0)$ where Q is a set of states, Σ is a set of input symbols, Δ is a set of output symbols, δ is a state transition function which maps $Q \times \Sigma$ to Q, λ is an output function which maps $Q \times \Sigma$ to Δ, and q_0 is the initial state. In this section, a Mealy machine $\mathbf{M}_2$ which translates the leg states into a set of symbols is described first. Then, the other Mealy machine $\mathbf{M}_1$ using the symbols generated by $\mathbf{M}_2$ to determine the control command is introduced.

A. Translation of Leg States into Input Symbols

The Mealy machine $\mathbf{M}_2 = (Q', \Sigma', \Delta', \delta', \lambda', q'_0)$ translates the leg states into the input symbols of $\mathbf{M}_1$ by mapping sequences of leg states to single symbols in order to simplify the design of the Mealy machine $\mathbf{M}_1$. If the current leg state is undetermined, the state of $\mathbf{M}_2$ is not changed because we do not want to make any decision when we are not sure of the current leg state. The details of $\mathbf{M}_2$ is described as follows.

1) Q' represents a set of states, and $Q' = \{q'_0, q'_1, q'_2, \ldots, q'_{3k-2}\}$ where k = frame rate.
2) $\Sigma' = \{<1>, <2>, <3>, <4>, <5>, <6>, <7>, <8>, <9>, <10>, <11>, <12>, <13>, <14>, <15>, <16>,$ <X>$\}$ are the set of input symbols where the symbols $<1>, <2>, \ldots$

TABLE V
DEFINITIONS OF δ' AND λ' OF $\mathbf{M}_2$

Next state / Output	State												
Input	q_0'	q_1'	q_2'	...	q_k'	q_{k+1}'	q_{k+2}'	...	q_{2k-1}'	q_{2k}'	q_{2k+1}'	...	q_{3k-2}'
<1>	-	q_2' / <s,s>	q_3' / <s,s>	...	q_k' / <long s,s>	q_1' / <s,s>	q_2' / <s,s>	...	q_2' / <s,s>	-	-	...	-
<2>	-	q_1' / <v,v>	q_1' / <v,v>	...	q_1' / <v,v>	q_1' / <v,v>	q_1' / <v,v>	...	q_1' / <v,v>	-	-	...	-
<3>	-	q_1' / <v,s>	q_1' / <v,s>	...	q_1' / <v,s>	q_1' / <v,s>	q_1' / <v,s>	...	q_1' / <v,s>	-	-	...	-
<4>	-	q_{k+1}' / <cv,s>	q_{k+1}' / <cv,s>	...	q_{k+1}' / <cv,s>	q_{k+2}' / <cv,s>	q_{k+3}' / <cv,s>	...	q_0' / <stop>	-	-	...	-
<5>	-	q_1' / <h,s>	q_1' / <h,s>	...	q_1' / <h,s>	q_1' / <h,s>	q_1' / <h,s>	...	q_1' / <h,s>	-	-	...	-
<6>	-	q_{k+1}' / <ch,s>	q_{k+1}' / <ch,s>	...	q_{k+1}' / <ch,s>	q_{k+2}' / <ch,s>	q_{k+3}' / <ch,s>	...	q_0' / <stop>	-	-	...	-
<7>	-	q_1' / <f,s>	q_1' / <f,s>	...	q_1' / <f,s>	q_1' / <f,s>	q_1' / <f,s>	...	q_1' / <f,s>	-	-	...	-
<8>	-	q_{k+1}' / <cf,s>	q_{k+1}' / <cf,s>	...	q_{k+1}' / <cf,s>	q_{k+2}' / <cf,s>	q_{k+3}' / <cf,s>	...	q_0' / <stop>	-	-	...	-
<9>	-	q_1' / <s,v>	q_1' / <s,v>	...	q_1' / <s,v>	q_1' / <s,v>	q_1' / <s,v>	...	q_1' / <s,v>	-	-	...	-
<10>	-	q_{k+1}' / <s,cv>	q_{k+1}' / <s,cv>	...	q_{k+1}' / <s,cv>	q_{k+2}' / <s,cv>	q_{k+3}' / <s,cv>	...	q_0' / <stop>	-	-	...	-
<11>	-	q_1' / <s,h>	q_1' / <s,h>	...	q_1' / <s,h>	q_1' / <s,h>	q_1' / <s,h>	...	q_1' / <s,h>	-	-	...	-
<12>	-	q_{k+1}' / <s,ch>	q_{k+1}' / <s,ch>	...	q_{k+1}' / <s,ch>	q_{k+2}' / <s,ch>	q_{k+3}' / <s,ch>	...	q_0' / <stop>	-	-	...	-
<13>	-	q_1' / <s,f>	q_1' / <s,f>	...	q_1' / <s,f>	q_1' / <s,f>	q_1' / <s,f>	...	q_1' / <s,f>	-	-	...	-
<14>	-	q_{k+1}' / <s,cf>	q_{k+1}' / <s,cf>	...	q_{k+1}' / <s,cf>	q_{k+2}' / <s,cf>	q_{k+3}' / <s,cf>	...	q_0' / <stop>	-	-	...	-
<15>	q_{2k}' / <X>	-	-	...	-	-	-	...	-	q_{2k+1}' / <X>	q_{2k+2}' / <X>	...	q_1' / <start>
<16>	q_0' / <X>	-	-	...	-	-	-	...	-	q_0' / <X>	q_0' / <X>	...	q_0' / <X>
<X>	-	q_1' / <X>	q_2' / <X>	...	q_k' / <X>	q_{k+1}' / <X>	q_{k+2}' / <X>	...	q_{2k}' / <X>	-	-	...	-

, and, $< 16 >$ correspond to the leg states 1, 2, ..., and 16, respectively, and <X> means that the leg state is not determined.

3) $\Delta' = \{$<s,s>, <v,v>, <v,s>, <cv,s>, <h,s>, <ch,s>, <f,s>, <cf,s>, <s,v>, <s,cv>, <s,h>, <s,ch>, <s,f>, <s,cf>, <long s,s>, <start>, <stop>, <X>$\}$ are the output symbols where <s,s>, <v,v>, ..., and <s,cf> represent the leg states and their definitions are given in Table II, <long s,s> and <start> mean that the leg state remains in the state 1 and state 15, respectively, at least 3 s and <stop> means that the user has crossed his legs at least 3 s, and <X> means that the leg state cannot be resolved.

4) δ' and λ' are defined in Table V.

B. Update of State Parameters

Because the position and orientation of a user may be changed, $\mathbf{e}_s$, $\mathbf{e}_f$, and the six directional vectors $\mathbf{ld}_v$, $\mathbf{ld}_s$, $\mathbf{ld}_f$, $\mathbf{rd}_v$, $\mathbf{rd}_s$, and $\mathbf{rd}_f$ must be updated accordingly. As discussed in the previous section, the directional vectors can be obtained when the feet are on the ground. That is, when $\mathbf{M}_2$ outputs the symbols <v,s>, <cv,s>, <h,s>, <ch,s>, <f,s>, and <cf,s>, the last initial position of the right mark can be updated and the directional vectors $\mathbf{rd}_v$, $\mathbf{rd}_s$, and $\mathbf{rd}_f$ of the right mark can be computed by (1). Similarly, when $\mathbf{M}_2$ outputs the symbols <s,v>, <s,cv>, <s,h>, <s,ch>, <s,f>, and <s,cf>, the last initial position of the left mark and the directional vectors $\mathbf{ld}_v$, $\mathbf{ld}_s$, and $\mathbf{ld}_f$ of the left mark can be updated. In addition, when $\mathbf{M}_2$ outputs the symbol <start>, the six directional vectors are all updated. However, before updating the directional vectors, $\mathbf{e}_s$ and $\mathbf{e}_f$ must be computed first.

C. Determination of Commands Associated with Leg Motion by the Mealy Machine M_1

The Mealy machine $\mathbf{M}_1 = (Q, \Sigma, \Delta, \delta, \lambda, q_0)$ receives the output symbol of $\mathbf{M}_2$ as its input symbol and outputs the control command. The state of M_1 is unchanged if the input symbol is <X>. That is, $\mathbf{M}_1$ does not make any decision until it has enough information. The definition of $\mathbf{M}_1$ is described in the following.

1) $Q = \{q_0, q_1, q_2, \ldots, q_{12}\}$.
2) $\Sigma = \Delta'$.

TABLE VI
DEFINITIONS OF δ AND λ OF $\mathbf{M}_1$

Next state / Output	State												
Input	q_0	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9	q_{10}	q_{11}	q_{12}
<s,s>	-	q_1/no	q_1/no	q_9/no	q_1/no	q_1/no	q_{11}/no	q_1/no	q_1/no	q_9/no	q_1/b	q_{11}/no	q_1/f
<v,v>	-	q_2/j	q_2/no	q_2/j	q_2/j	q_2/j	q_2/j	q_2/j	q_2/j	q_2/j	q_2/j	q_2/j	q_2/j
<v,s>	-	q_3/no	q_3/no	q_3/no	q_3/no	q_3/no	q_3/w	q_3/no	q_3/no	q_{10}/no	q_{10}/no	q_3/w	q_3/f
<cv,s>	-	q_3/no	q_3/no	q_3/no	q_3/no	q_3/no	q_3/w	q_3/no	q_3/no	q_{10}/no	q_{10}/no	q_3/w	q_3/f
<h,s>	-	q_4/ml	q_4/ml	q_4/ml	q_4/no	q_4/ml	q_4/ml	q_4/ml	q_4/ml	q_4/ml	q_4/ml	q_4/ml	q_4/f+ml
<ch,s>	-	q_4/no	q_4/no	q_4/no	q_4/no	q_4/no	q_4/no	q_4/no	q_4/no	q_4/no	q_4/no	q_4/no	q_4/no
<f,s>	-	q_5/tl	q_5/tl	q_5/lk	q_5/tl	q_5/no	q_5/tl	q_5/tl	q_5/tl	q_5/lk	q_5/lk	q_5/tl	q_5/f+tl
<cf.s>	-	q_5/no	q_5/no	q_5/no	q_5/no	q_5/no	q_5/no	q_5/no	q_5/no	q_5/no	q_5/no	q_5/no	q_5/no
<s,v>	-	q_6/no	q_6/no	q_6/w	q_6/no	q_6/no	q_6/no	q_6/no	q_6/no	q_6/w	q_6/b	q_{12}/no	q_{12}/no
<s.cv>	-	q_6/no	q_6/no	q_6/w	q_6/no	q_6/no	q_6/no	q_6/no	q_6/no	q_6/w	q_6/b	q_{12}/no	q_{12}/no
<s.h>	-	q_7/mr	q_7/mr	q_7/mr	q_7/mr	q_7/mr	q_7/mr	q_7/no	q_7/mr	q_7/mr	q_7/b+mr	q_7/mr	q_7/mr
<s,ch>	-	q_7/no	q_7/no	q_7/no	q_7/no	q_7/no	q_7/no	q_7/no	q_7/no	q_7/no	q_7/no	q_7/no	q_7/no
<s,f>	-	q_8/tr	q_8/tr	q_8/tr	q_8/tr	q_8/tr	q_8/tr	q_8/tr	q_8/no	q_8/tr	q_8/b+tr	q_8/rk	q_8/rk
<s,cf>	-	q_8/no	q_8/no	q_8/no	q_8/no	q_8/no	q_8/no	q_8/no	q_8/no	q_8/no	q_8/no	q_8/no	q_8/no
<long s,s>	-	q_1/no	q_1/no	q_1/no	q_1/no	q_1/no	q_1/no	q_1/no	q_1/no	q_1/no	q_1/no	q_1/no	q_1/no
<start>	q_1/st	-	-	-	-	-	-	-	-	-	-	-	-
<stop>	–	q_0/sp	q_0/sp	q_0/sp	q_0/sp	q_0/sp	q_0/sp	q_0/sp	q_0/sp	q_0/sp	q_0/sp	q_0/sp	q_0/sp
<X>	q_0/no	q_1/no	q_2/no	q_3/no	q_4/no	q_5/no	q_6/no	q_7/no	q_8/no	q_9/no	q_{10}/no	q_{11}/no	q_{12}/no

The output symbols < no action>, <walk>, <jump> , <stop> , <turn left>, <turn right>, <move left>, <move right>, <left kick>, <right kick>, <forward>, <backward>, <backward+turn right>, <backward+move right>, <forward+turn left>, AND <forward+move left> are abbreviated to no, w, j, st, sp, tl, tr, ml, mr, lk, rk, f, b, b+tr,b+mr, f+tl, AND f+ml, respectively.

3) Δ = {<start>, <stop>, <jump>, <walk>, <turn left>, <turn right>, <move left>, <move right>, <left kick>, <right kick>, <forward>, <backward>, <no action>, <forward+move left>, <forward+turn left>, <backward+move right>, <backward+turn right>}. The definitions of these output symbols are given in Table I except the symbols <no action>, <forward+ move left>, <forward+turn left>, <backward+move right>, and <backward+turn right>, where <no action> means that no control command is generated; <forward+move left> denotes the concatenation of the two control commands, <forward> and <move left>, meaning the change of the moving direction to forward followed by a left move; and <forward+turn left>, <backward+move right>, and <backward+turn right> are defined similarly.

4) δ and λ are defined in Table VI where the control commands are generated only when the input symbols are changed.

V. MARK DETECTION

To match the color of the marks on the user's legs, a uniform color space is preferred because the corresponding color similarity measure is simple. In this study, the $L^*u^*v^*$ coordinate system is adopted because it is a C.I.E. standard for a uniform color space [19]. The transformation from the N.T.S.C. RGB coordinate system R_N, G_N, and B_N to the $L^*u^*v^*$ coordinate system L^*, u^* and v^* can be found in [19].

To detect the marks, the models of the marks are built first by averaging the $L^*u^*v^*$ values of the pixels in the regions of the marks. The model of the mark on the right leg is denoted by $L_r(0)$, $u_r(0)$, and $v_r(0)$, and the model of the left leg is denoted by $L_l(0)$, $u_l(0)$, and $v_l(0)$. To adapt to the variation of illumination and shadow effects, the models of the marks, (L_l, u_l, v_l) and (L_r, u_r, v_r), used to find the marks at time t are adjusted according to the color of the marks at time $t-1$ using the formulas described below:

$$\begin{aligned} L_l &= \theta L_l(t-1) + (1-\theta)L_l(0) \\ L_r &= \theta L_r(t-1) + (1-\theta)L_r(0) \\ u_l &= \theta u_l(t-1) + (1-\theta)u_l(0) \\ u_r &= \theta u_r(t-1) + (1-\theta)u_r(0) \\ v_l &= \theta v_l(t-1) + (1-\theta)v_l(0) \\ v_r &= \theta v_r(t-1) + (1-\theta)v_r(0) \end{aligned} \tag{9}$$

where θ ranging from zero to one represents the weight of the models at time t correlative to the those at time $t-1$. In this study, $\theta = 0.5$.

For each frame which is a color image of size 320 × 240 pixels, the centroids of the marks are found by the following steps.

1) Divide the input image into nonoverlapped blocks of size 5 × 5 pixels and take the RGB values of each block to be the average of those of the pixels in the block.
2) Compute the color distances between each of the blocks to the color models of the marks in the $L^*u^*v^*$ coordinate system. If the color distance between a block and the left mark is smaller than a pre-defined threshold value, the block is called a "1" block. If the color distance between a block and the right mark is smaller than a pre-defined threshold value, the block is called a "2" block. Otherwise, the block is called a "0" block.
3) Perform the component labeling algorithm to find the connected components consisting of "1" blocks, called "1" components, and the connected components consisting of "2" blocks, called "2" components. Eliminate connected components with areas smaller than a pre-defined threshold value.
4) Compute the centroid of each component. The "1" component with the centroid closest to the position of the left mark in the previous frame is selected as a candidate component for the left mark in the current frame. The candidate component for the right mark is obtained similarly.
5) For each mark, if the candidate component of the mark is found, then apply the region growing algorithm to expand the candidate component where the threshold value for the color similarity measure used in the expansion process is a quarter of the threshold value used in the second step, compute the centroid of the expanded component as the position of the mark, average the L^*, u^*, and v^* values of the blocks in the expanded component as the model of the mark at the current frame, and adjust the models of the marks using (9) for detecting the mark in the next frame. If the candidate of the mark is not found (because the mark may be occluded by another leg), the x-position of the mark is set to be the x-position of the other mark and the y-position of the mark is taken to be the y-position of the mark in the previous frame, and the model of the mark is unchanged.

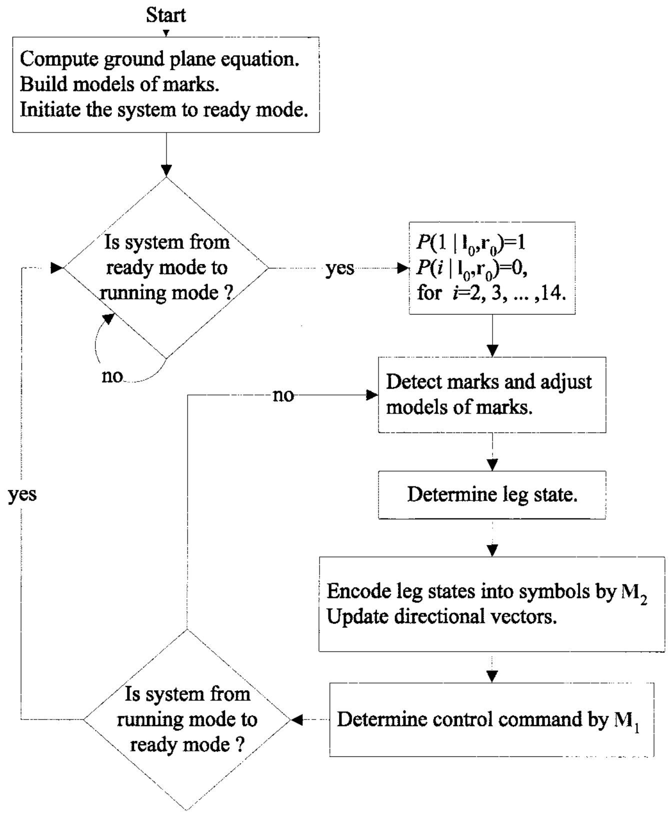


Fig. 4. Flowchart of the proposed system.

Initially, the system is in the *ready mode*. When the system is in the *ready mode*, the leg state is determined by (8) until $\mathbf{M}_1$ outputs the symbol <start>, i.e., until the system goes into the *running mode*. When the system is in the *running mode*, the leg state is determined by (4). If $\mathbf{M}_1$ outputs the symbol <stop>, then the system goes into the *ready mode*. As a summary, the flow of the proposed system is illustrated in Fig. 4.

VI. Experimental Results

In this study, the proposed system was implemented on Windows 95 by a Pentium-233 with MMX PC and the input color images were obtained by a commercial video capture card. To speed up the processing time, some computations are pre-calculated, and the results were stored in look-up tables (e.g., (5) and some of the intermediate computations of the transformation from the RGB coordinate system to the $L^*u^*v^*$ coordinate system). The frame rate of the proposed system is 14 Hz.

A. Analysis of Conditional Probability Distributions in Table III

To analyze the robustness and sensitivity of the conditional probability distributions defined in Table III, plots of these functions with respect to a typical leg configuration are given in Fig. 5. Only the plots of the conditional probability distributions

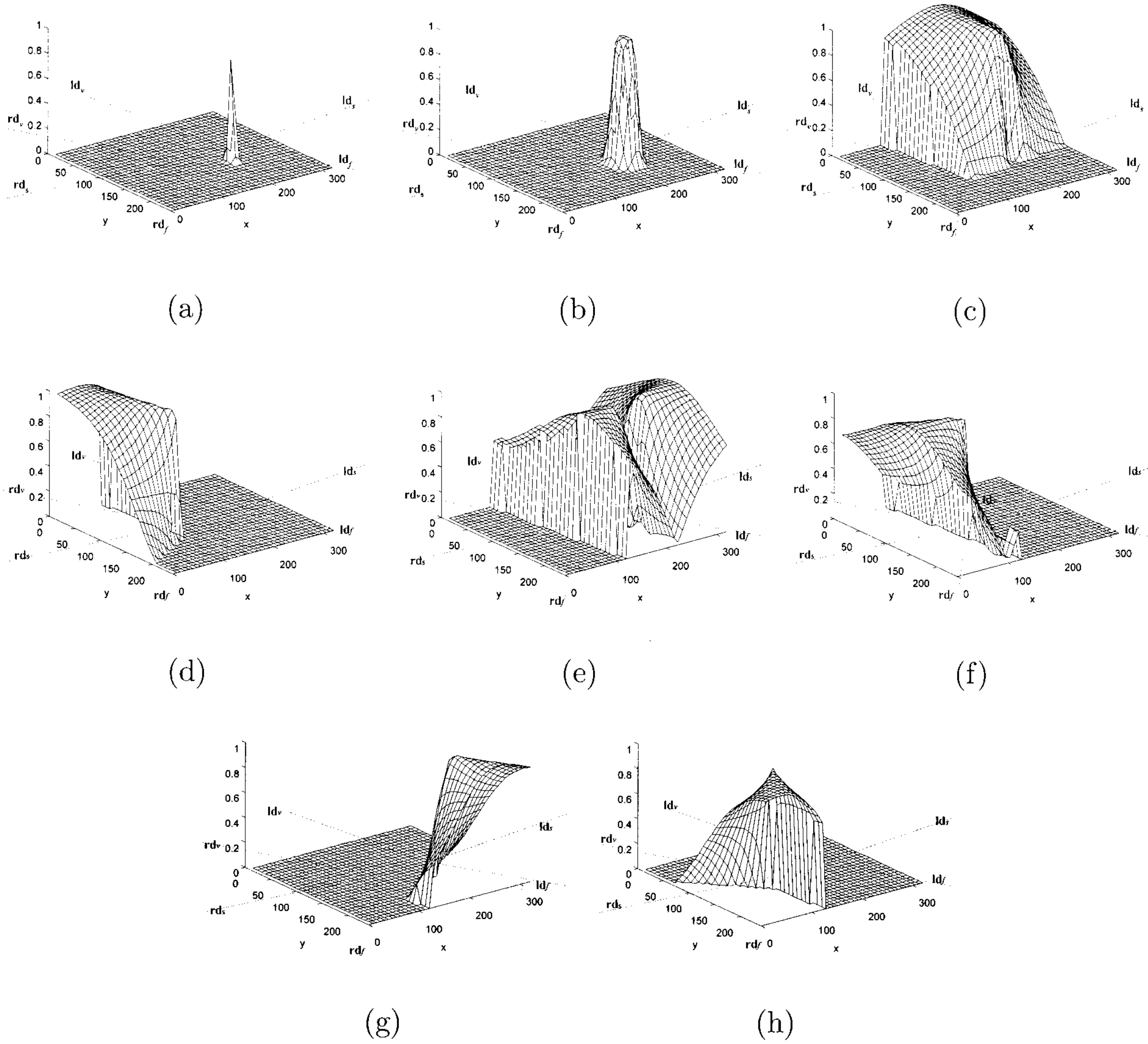


Fig. 5. Plots of the conditional probability distributions for the leg states with respect to a typical leg configuration. (a), (b) Leg state 1 with α = 5 and 20, respectively. (c)–(h) Leg states 3–8 with α = 20, respectively.

for the leg states with the right leg being stationary are shown because the behaviors of the other conditional probability distributions can be easily realized from them. To clearly show the shapes of the conditional probability distributions, the current position of the right mark was assumed in an initial position; that is, $P_s(\text{dist}(\mathbf{r}_t, \mathbf{r}_{t-m})) = 1$. Fig. 5(a) and (b) show the conditional probability distributions for the leg state 1 with different sensitivity levels for detecting the movement of the left leg. By adjusting α in (5) to control the sensitivity for detecting point movement, it can be seen that the area for supporting that the left leg being stationary in Fig. 5(a) with $\alpha = 5$ is smaller than that in Fig. 5(b) with $\alpha = 20$; that is, a smaller value of α is more sensitive for detecting the movement of legs than a larger one. Fig. 5(c)–(h) are the plots of the conditional probability distributions for the leg states 3, 4, 5, 6, 7, and 8, respectively. They show that the regions for supporting the legs in those leg states are along the desired directions of those state. As a consequence, the definitions of the conditional probability distributions in Table III coincide with our expectation. Fig. 6 shows the curves of the *a posteriori* probabilities of the fourteen leg states over 103 frames when the system is in the *running mode*. The control commands are <walk>, <walk>, <walk>, <walk>, <walk>, <move right>, <move left>, and <move right>, which are recognized in the 25th frame, the 33rd frame, the 41st frame, the 49th frame, the 58th frame, the 67th frame, the 78th frame, and the 103rd frame, respectively. We can see that all decisions are made when the *a posteriori* probability of the selected leg state is larger than 0.5.

B. Performance Evaluation

To know the performance of the system, each of the control commands was repeatedly issued 100 times to test the proposed system. The result is shown in Table VII and the recognition rate is 97.8%. In addition, about two minutes of operation was performed to test the long-term performance of the system. The statistics of the operation is shown in Table VIII, and the recognition rate is 96.3%.

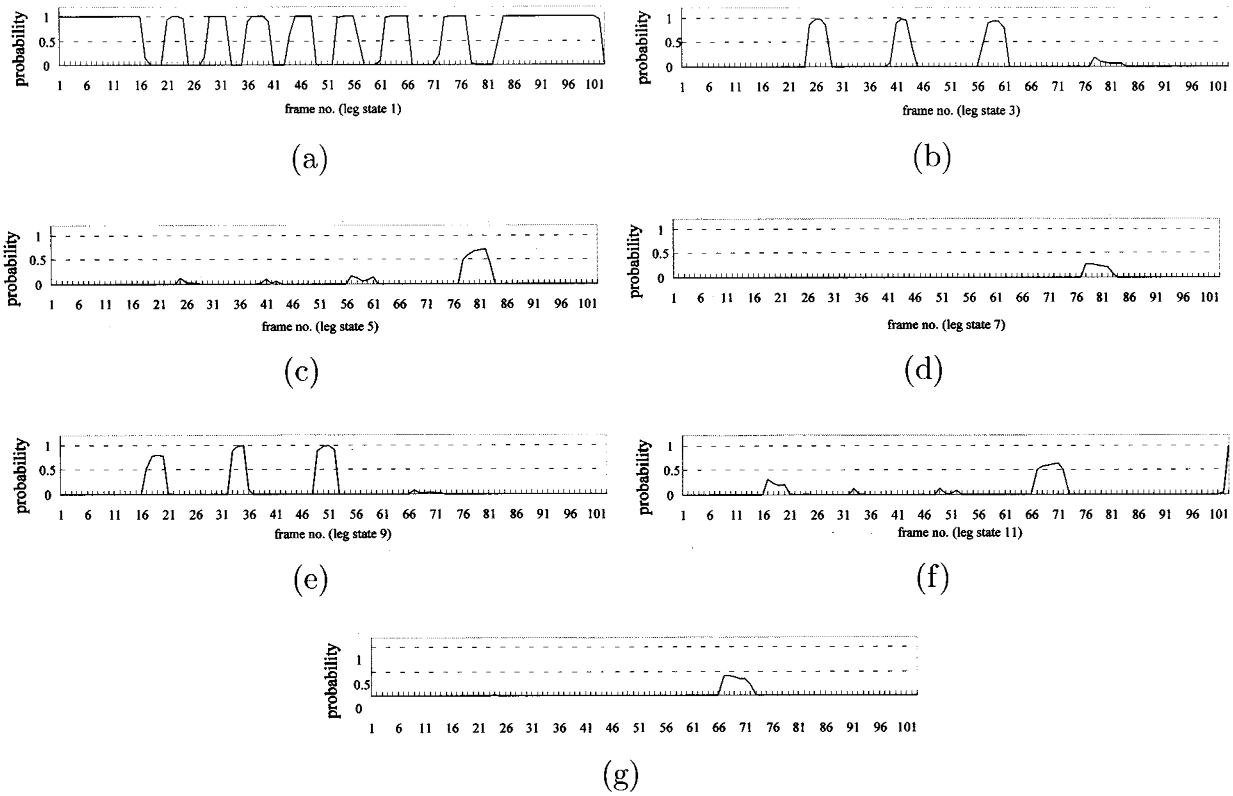


Fig. 6. A plot of the *a posteriori* probabilities of the leg states over 103 frames where the leg states with tiny *a posteriori* probabilities are not shown, where (a) to (g) are for the leg states 1, 3, 5, 7, 9, 11, and 13, respectively.

TABLE VII
RECOGNITION RATES FOR EACH OF CONTROL COMMANDS

command	recognition rate	command	recognition rate
<start>	100	<move left>	97
<stop>	100	<move right>	98
<walk>	98	<turn left>	98
<jump>	100	<turn right>	97
<forward>	97	<left kick>	95
<backward>	98	<right kick>	96

C. Discussions

Two types of errors are found in the experimental results. The first type of error is missing the control command issued. The other type of error is incorrect recognition of the control command. From the experimental results, we found that most of the errors were coming from the first type of errors. A common error source for the two types of errors is the failure in detecting the color bands due to sudden change of illumination or serious shadow effects. In addition, the first type of errors can be also caused by fast or inapparent movement of a user, and another possible error source for the second type of errors is that the action of moving a leg forward may be confused with the action of moving the leg sideways especially when the user does not face the camera.

Since the control commands are recognized from a sequence of leg states, erroneous recognition or missing of leg states may lead to failure of the subsequent recognition process. However, we find that if the state parameters are still correct, then the system can be back to normal after the user stands still (i.e., the state of the user's legs is 1) for a few frames. The reason is that based on the correct state parameters, it is possible for the Markov process to recognize the leg states in those frames correctly, and thus the Mealy machine $\mathbf{M}_1$ will be back to the state q_1, which is the beginning state of $\mathbf{M}_1$ when the system switches from the *ready mode* to the *running mode*, according to state transition table of the Mealy machine $\mathbf{M}_1$. Nevertheless, once the state parameters are incorrect, the system may suffer in an error loop. A simple scheme for the system to recover from the error is to "reset" itself; that is, the system performs the steps of switching from the *ready mode* to the *running mode* to re-initialize its parameters. Accordingly, in this study, a simple module for the system to recover from the error can be easily added by employing the technique described in Section III-E to check whether the user keeps stationary with two feet close to each other for a certain seconds. That is, in the *running mode*, once the system finds that the user keeps stationary with two feet close to each other for a certain seconds, the system will automatically reset itself. Thus, if the user finds that the system malfunctions, he can stand still for a while to recover the system.

TABLE VIII
EXPERIMENTAL RESULTS FOR LONG-TERM TEST

command	no. of times issued	recognized	command	no. of times issued	recognized
<start>	1	1	<move left>	10	10
<stop>	1	1	<move right>	10	9
<walk>	46	45	<turn left>	10	9
<jump>	5	5	<turn right>	10	10
<forward>	2	2	<left kick>	5	5
<backward>	2	2	<right kick>	5	4

VII. CONCLUDING REMARKS

A vision-based human–computer interaction system must be low-cost, real-time, and robust to the change of environment [4]. In this study, a vision-based human–computer interaction system is developed on a commercial PC to allow a user to communicate with the computer by issuing defined control commands via his legs. In the proposed system, only a single camera is used and no special hardware is required. By tracking the color bands on the user's feet, the leg motion can be modeled by a first-order Markov process, and thus the recognition of leg states and the update of state parameters can be done in a systematic way. In addition, a method to learn the probability transition table for the Markov process is also proposed. To continuously recognize the meaning of a sequence of leg motions, Mealy machines are adopted in this study. Since the transition function of a Mealy machine is deterministic, the response time of the system is short. In addition, by using the output function of a Mealy machine to yield recognized control commands, the implementation of the proposed approach is straight-forward and simple. To adapt to color change of the marks because of the variation of illumination and shadow effects, a simple method is also proposed to adjust the color models of the marks. The frame rate of the proposed system is 14 Hz and the experimental results show the feasibility of the proposed system.

REFERENCES

[1] V. I. Pavlovic, R. Sharma, and T. S. Huang, "Visual interpretation of hand gestures for human–computer interaction: A review," *IEEE Trans. Patert. Anal. Machine Intell.*, vol. 19, no. 7, pp. 677–695, 1997.
[2] W. T. Freeman, K. Tanaka, J. Ohta, and K. Kyuma, "Computer vision for computer games," in *Proc. Int. Conf. Automatic Face and Gesture Recognition*, Killington, VT, Oct. 1996, pp. 100–105.
[3] F. Biocca, "Virtual reality technology: A tutorial," *J. Commun.*, vol. 42, no. 2, pp. 23–72, 1992.
[4] F. K. H. Quek, "Eyes in the interface," *Image and Vis. Comput.*, vol. 13, no. 6, pp. 511–525, 1995.
[5] D. H. Ballard and C. M. Brown, *Computer Vision*. Englewood Cliffs, NJ: Prentice-Hall, 1982.
[6] H. J. Lee and Z. Chen, "Determination of 3D human body postures from a single view," *Comput. Vis., Graphics, and Image Processing*, vol. 33, pp. 148–168, 1985.
[7] Z. Chen and H. J. Lee, "Knowledge-guided visual perception of 3D human gait from a singele image sequence," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, no. 2, pp. 336–342, 1992.
[8] K. Rohr, "Toward model-based recognition of human movements in image sequences," *CVGIP: Image Understanding*, vol. 59, no. 1, pp. 94–115, 1994.
[9] D. M. Gavrila and L. S. Davis, "Tracking humans in action: A 3D model-based approach," in *Proc. Image Understanding Workshop*, Palm Springs, CA, 1996, pp. 737–746.
[10] S. X. Ju, M. J. Black, and Y. Yacoob, "Carboard people: A parameterized model of articulated image motion," in *Proc. International Conference on Automatic Force and Gesture Recognition*, Killington, VT, Oct. 1996, pp. 38–43.
[11] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Trans. Pattern. Anal. Machine Intell.*, vol. 19, pp. 780–785, 1997.
[12] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models—their training and application," *Comput. Vis. Image Understanding*, vol. 61, no. 1, pp. 38–59, 1995.
[13] A. Blake, B. Bascle, M. Isard, and J. MacCormick, "Statistical models of visual shape and motion," *Phil. Trans. Roy. Soc. Lond. A*, vol. 365, no. 4, pp. 1283–1302, May 1998.
[14] T. Darrell and A. Pentland, "Space-time gestures," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1993, pp. 335–340.
[15] J. Davis and M. Shah, "Visual gesture recognition," *Proc. Inst. Elect. Eng.—Vis. Image Signal Process*, vol. 141, no. 2, pp. 101–106, 1994.
[16] M. Nadler and E. P. Smith, *Pattern Recognition Engineering*. New York: Wiley, 1993.
[17] S. Y. Chen and W. H. Tsai, "A systematic approach to analytic determination of camera parameters by line features," *Pattern Recognit.*, vol. 23, no. 8, pp. 859–877, 1990.
[18] P. A. Devijver and J. Kittler, *Pattern Recognition A Statistical Approach*. London, U.K.: Prentice-Hall, 1982.
[19] W. K. Pratt, *Digital Image Processing*, 2nd ed. New York: Wiley, 1991.

Chin-Chun Chang received the B.S. degree from the Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan, in 1989, and M.S. degree from the Institute of Computer and Information Science, National Chiao Tung University, in 1991, where he is currently working toward the Ph.D. degree.

From 1993 to 1994, he was with the Telecommunication Laboratories, Ministry of Transportation and Communications, Taiwan, R.O.C. His research interests include computer vision, virtual reality, and pattern recognition.

Wen-Hsiang Tsai (S'78–M'85–SM'91) was born in Tainan, Taiwan, R.O.C., in 1951. He received the B.S. degree in electrical engineering (with major in computer engineering) from National Taiwan University, Taipei, Taiwan, R.O.C., in 1973, the M.S. degree in electrical engineering (with major in computer science) from Brown University, Providence, RI, in 1977, and the Ph.D. degree (with major in computer engineering) in electrical engineering from Purdue University, West Lafayette, IN, in 1979.

He joined the faculty of National Chiao Tung University, Hsinchu, Taiwan, in November 1979, where he is currently a Professor in the Department of Computer and Information Science and the Dean of Academic Affairs of the University. He has been an Associate Professor of the Department of Computer Engineering (now the Department of Computer Science and Information Engineering) and the Acting Director of the Institute of Computer Engineering, the Head of the Department of Computer and Information Science (1984–1988), the Associate Director of the Microelectronics and Information System Research Center (1984–1987), and the Dean of General Affairs of the University (1995–1996). Additionally, he has served as or is now a Consultant to several major research and development institutions in Taiwan, including Chun-Shan Institute of Science and Technology, the Industrial Technology Research Institute, the Information Industry Institute, and China Textile Research Center. He has acted as the Coordinator of Computer Science in the Engineering Section of the National Science Council, and was a member of the Counselor Committee of the Institute of Information Science of Academia Sinica in Taipei. His major research interests include image processing, pattern recognition, computer vision, neural networks, and Chinese information processing. To date, he has published about 240 academic papers, including 103 journal papers and 137 conference papers. He has supervised the thesis studies of 24 Ph.D. students and 91 M.S. students.

Dr. Tsai has served as Editor of several academic journals, including COMPUTER QUARTERLY (now JOURNAL OF COMPUTERS), PROCEEDINGS OF THE NATIONAL SCIENCE COUNCIL, INTERNATIONAL JOURNAL OF PATTERN RECOGNITION AND ARTIFICIAL INTELLIGENCE, JOURNAL OF INFORMATION SCIENCE AND ENGINEERING AND PATTERN RECOGNITION. He is also the Editor-in-Chief of the JOURNAL OF INFORMATION SCIENCE AND ENGINEERING, the Editor of the JOURNAL OF THE CHINESE ENGINEERS, and the Chairman of the Chinese Image Processing and Pattern Recognition Society at Taiwan. He is the recipient of one Distinguished Research Award, four Outstanding Research Awards, and two Special Research Project Awards, all of the National Science Council between 1987–2001. He was the recipient of the 13th Annual Best Paper Award of the Pattern Recognition Society, and the 31st Annual Science Research Award of S. K. Chuang's Public Welfare Foundation. He was elected as an Outstanding Talent of Information Science and Technology of the Republic of China in 1986, received the Best Teacher Award of the Ministry of Education in 1989, and was the recipient of the Distinguished Official Award of the Ministry of Education in 1994. He received the Acer Long-Term Ph.D. and Master's Thesis Supervision Awards in 1989, 1992, 1994, and 1996, and the Xerox Ph.D. Dissertation Supervision Award in 1992. He was the recipient of many Academic Paper Awards made by several academic societies, including one by the Computer Society of the Republic of China in 1989, nine by the Chinese Image Processing and Pattern Recognition Society, and one by the Federation of Image Product Companies. He is a member of the Chinese Image Processing and Pattern Recognition Society, the Medical Engineering Society of the Republic of China, and the International Chinese Computer Society.