A View-based Multiple Objects Tracking and Human Action Recognition for Interactive Virtual Environments

Jin Choi[1], Yong-il Cho[1], Kyusung Cho[1], Sujung Bae[1], Hyun S. Yang[1]

[1] AIM Lab., Computer Science Dept., KAIST,
Daejeon, South Korea
{jin_choi, caelus, qtboy, sjbae, hsyang}@paradise.kaist.ac.kr

Abstract. As environments become smart in accordance with advances in ubiquitous computing technology, researchers are struggling to satisfy users' diverse and sophisticated demands. The aim of the present work is to enable multiple persons in a interactive virtual environment to simultaneously and conveniently interact with virtual agents. To this end, we propose a real-time system that robustly tracks multiple persons in virtual environments and recognizes their actions through image sequences acquired from a single fixed camera. The proposed system is compromised of three components: blob extraction, object tracking, and human action recognition. Given an image, we extract blobs using the Mixture of Gaussians algorithm with a hierarchical data structure and we additionally remove shadows and highlights in order to obtain a more accurate object silhouette. We then track multiple objects using a motion-based object model and an inference graph for handling grouping and fragment problems. Finally, we model an action as a Motion History Image (MHI) based on given object tracks, normalize the MHI, reduce the MHI using PCA, and classify an action using a multi-layer perceptron. To evaluate the performance of the proposed system, we employed it in an augmented reality application where multiple persons can interact with a virtual pet. The results confirm that reliable object tracking is achieved and multiple persons' actions can be recognized for applications in interactive virtual environments.

Keywords: object tracking, human action recognition, foreground detection, motion history image, HCI

1 Introduction

As environments become smart with advances in ubiquitous computing technology, many researchers are struggling to satisfy users' diverse and sophisticated demands. Because knowing the location, identity, and behavior of people is necessary for users to interact with virtual agents, many view-based human tracking and action recognition methods have been studied [1][2][3][4]. However, most previous studies considered only one person or two persons in a given environment.

The aim of the present work is to enable multiple persons in a virtual environment to simultaneously and conveniently interact with virtual agents. To this end, we propose a real-time system that robustly tracks multiple persons in a virtual environment and recognizes their actions through image sequences acquired from a

single fixed camera. Because it does not employ a class-specific object model or domain knowledge, the proposed system can be readily used in a wide range of applications, including video surveillance.

In the next section we propose a multiple objects tracking and human action recognition system. We then evaluate the proposed system employing it in an augmented reality application in section 3 and summarize our conclusions in the final section.

2 The Proposed Multiple Objects Tracking and Human Action Recognition System

Our proposed real-time system can track multiple objects and recognize simple human actions such as walking, running, sitting, standing, falling, punching, kicking and turning through image sequences obtained from a single fixed camera. As shown in **Fig. 1**, the proposed system consists of three parts: blob extraction, object tracking, and human action recognition. The details of each part are explained below.

Fig. 1. The overview of the proposed system.

2.1 Blob Extraction

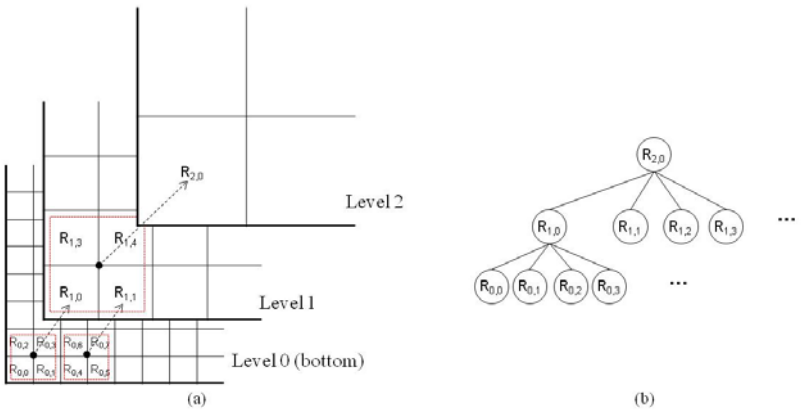

Fig. 2. The process of decomposing an image bottom-up to build a quad-tree (a) and a built quad-tree (b)

In the first part, we segment an input image as foreground and background. Foreground is expected as objects of interest to be a target for tracking and action recognition. The ability to rapidly extract correct object's appearance from an image is essential because object tracking and action modeling are based on it. To address this need, we have developed a modified Mixture of Gaussians (MoG) algorithm with hierarchical data structure. The MoG algorithm with a hierarchical data structure, proposed by Park et al. [5], reportedly enhances the processing speed significantly and yields results that are very similar to the results of the standard MoG algorithm. In Park's method [5], a quadtree of the hierarchical data structure is constructed top-down at the beginning. In this method, however, the user is unable to control the minimum region corresponding to leaf nodes because the image is recursively decomposed into four equal regions. To adjust the shape and size of minimum region on demand, we constructed a quadtree by decomposing an image bottom-up as shown in Fig. 2.

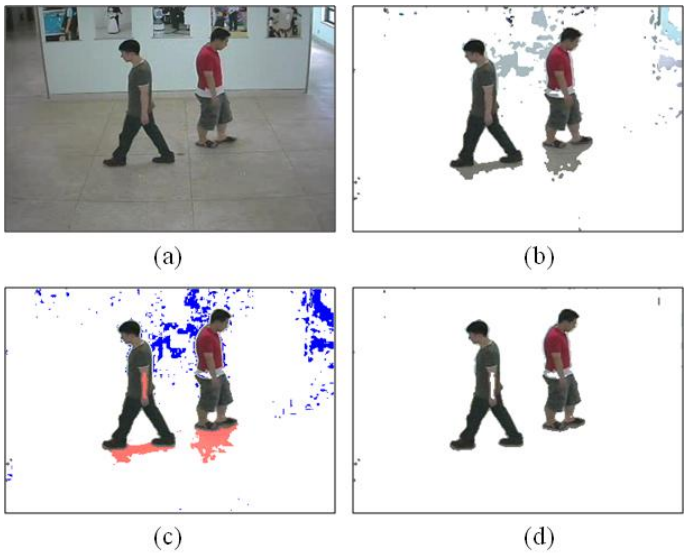

Fig. 3. (a) An input image. (b) A result of foreground extraction. (c) Detected shadow (marked in red) and highlight (marked in blue). (d) A result of foreground extraction without shadow and highlight.

Once we build a quad-tree in the initial step, we can effectively extract foreground in successive frames using the Park's algorithm [5] searching the quadtree. As shown in Fig. 3 (b), if we apply the above foreground extraction method, most of the results of foreground extraction include false positives of shadow and highlight induced by moving objects. To remove shadow and highlight from the result of foreground extraction, we assume that the intensity of a shadow pixel is scale-down of the intensity of the corresponding pixel in the background model, and we adopt the shadow and highlight detection algorithm proposed by Jacques Jr. et al. [6]. After removing shadow and highlight region, we get the final result of foreground extraction as shown in Fig. 3 (d). When obtained foreground pixels at frame t, through

connected component analysis, they are clustered into a set of blobs $B^t = \{b_i^t | i \text{ is an integer and } 0 \leq i\}$, where a blob b_i^t is a ith set of connected foreground pixels.

2.2 Object Tracking

In the ideal case, a blob represents an object, but in the real, multiple objects may appear as a single blob (grouping), or an object may be broken into several blobs (fragmentation). To handle these problems, as shown in Fig. 4, we develop an online multiple objects tracking framework based on the framework of Bose et al. [7]. In particular, given B^t, we get a set of object track O^{t-k+1} at frame t-k+1 where k is the size of a sliding window. Firstly, we detect blob association events by associating B^t with B^{t-1}, and update the blob inference graph according to blob association events, and we label each vertex as one of *Fragment, Object*, and *Group*. Lastly, we localize objects by using the blob inference graph and blob association event log.

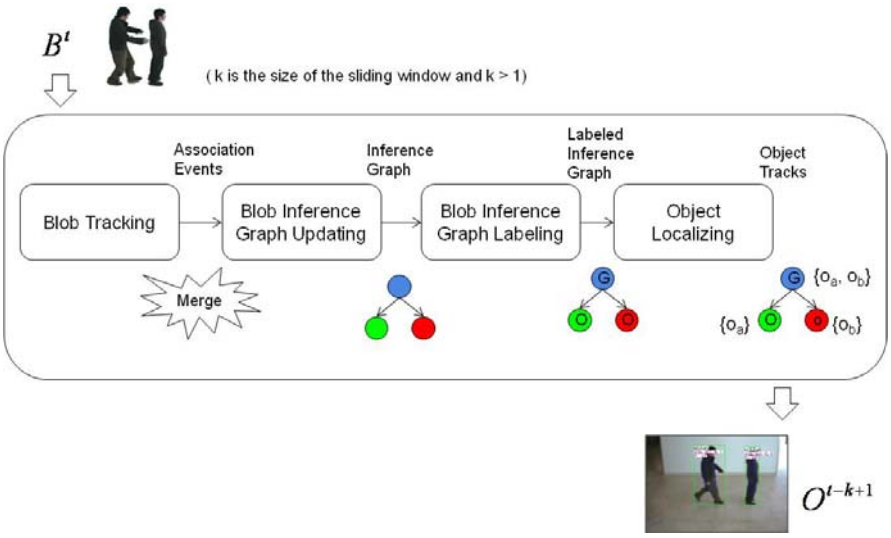

Fig. 4. The online multiple objects tracking framework

Blob Tracking. Given B^{t-1} and B^t extracted from two consecutive frames, we maintain blob tracks by inferring blob association events. As shown in Fig. 5, blob association events are classified into five events as follows: *continue*, *merge*, *split*, *appear*, and *disappear*. They can be inferred by a $|B^{t-1}| \times |B^t|$ correspondence matrix **M**.

To make a correspondence matrix, we compare regions corresponding to B^{t-1} and B^t. And an element of **M** is set as below:

$$\mathbf{M}[i,j] = \begin{cases} 0 & b_i^{t-1} \cap b_j^t \cong \emptyset \\ 1 & b_i^{t-1} \cap b_j^t \cong b_i^{t-1} \text{ or } b_i^{t-1} \cap b_j^t \cong b_j^t \end{cases}$$

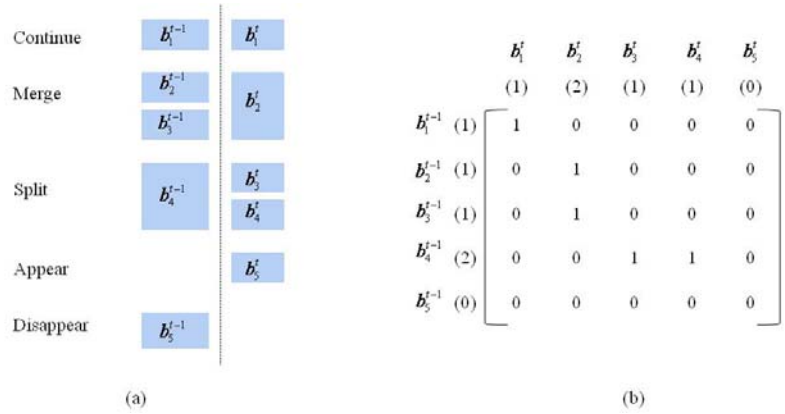

Fig. 5. (a) Five blob association events. (b) A blob correspondence matrix.

When $\mathbf{M}$ is determined, blob association events are inferred as below:

Appear. b_j^t appears if $\sum_{i=1}^{|B^{t-1}|} \mathbf{M}[i,j] = 0$

Disappear. b_i^{t-1} disappears if $\sum_{j=1}^{|B^t|} \mathbf{M}[i,j] = 0$

Continue. $b_i^{t-1} = b_j^t$ if $\sum_{i=1}^{|B^{t-1}|} \mathbf{M}[i,j] = 1$ and $\sum_{j=1}^{|B^t|} \mathbf{M}[i,j] = 1$.

Merge. $\{b_i^{t-1} | \mathbf{M}[i,j] = 1\}$ merge into b_j^t if $\sum_{i=1}^{|B^{t-1}|} \mathbf{M}[i,j] > 1$.

Split. b_i^{t-1} splits up into $\{b_j^t | \mathbf{M}[i,j] = 1\}$ if $\sum_{j=1}^{|B^t|} \mathbf{M}[i,j] > 1$.

Blob Inference Graph Updating & Labeling. We use a blob inference graph $\mathcal{G}$ to infer blob's label and localize objects. A vertex is associated with more than one blob, and a directed edge represents spatial relation between two vertexes. In Bose et al.'s tracking algorithm [7], they build a new inference graph at every frame, we, on the other hand, maintain only one inference graph during whole tracking, and that allows us to localize objects based on the inference graph. We update $\mathcal{G}$ according to blob association events as Table 1.

Table 1. Updating $\mathcal{G}$

Blob Association Event	Description
Appear $b_j^t \cap B^{t-1} = \emptyset$	Add a new vertex V_i and associate b_j^t with V_i
Continue $b_j^t = b_i^{t-1}$	Associate b_j^t with V_i corresponding to b_i^{t-1}
Merge	Add a new vertex V_i corresponding to b_j^t and directed edges

$b_j^t = \bigcup_i b_i^{t-1}$	whose tails are V_i and whose heads are vertexes corresponding to $\{b_i^{t-1} \mid b_i^{t-1} \subset b_j^t\}$
Split $b_i^{t-1} = \bigcup_j b_j^t$	Add new vertexes corresponding to $\{b_j^t \mid b_j^t \subset b_i^{t-1}\}$ and directed edges whose tails are V_i corresponding to b_i^{t-1} and whose heads are vertexes corresponding to $\{b_j^t \mid b_j^t \subset b_i^{t-1}\}$

After updating $\mathcal{G}$, we need to refine $\mathcal{G}$ for later labeling and localizing to satisfy two properties: if two vertexes are equivalent, their tracks should be associated with the same vertex; V_j is the descendant of V_i iff V_j is subset of V_i. For that reason, a vertex is represented by more than one Vertex Units Set (VUS), where a Vertex Unit (VU) is a leaf of $\mathcal{G}$. VUS is also updated according to blob association events. We can compare two vertexes through VUSs.

We label each vertex as one of *Fragment* (F), *Object* (O), and *Group* (G) using Bose et al.'s inference-graph labeling algorithm that stitches together blobs that belong to the same object. The loss of tracking information is, however, induced by the limit of the number of frames considered for tracking. To overcome this problem, we add a reliable variable to the vertex. If a vertex is reliable, its state is not initialized so that we can get more correct tracks.

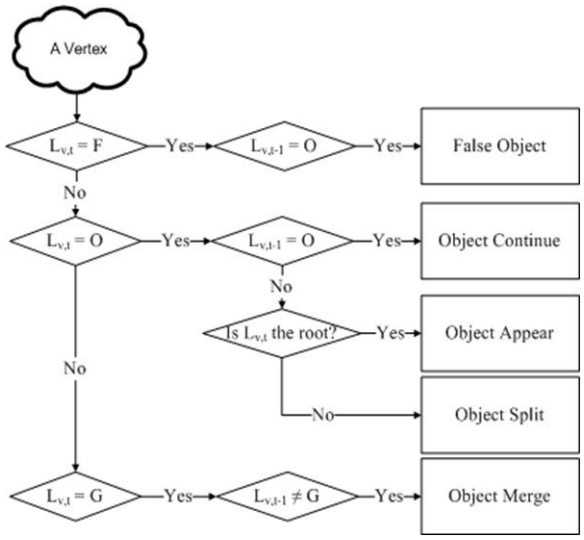

Fig. 6. A flow chart for object association event inference ($L_{v,t}$ is the label of a vertex v at frame t).

Object Localizing. We infer object tracks depending on the state change of vertex's label caused by blob association events as shown in Fig. 6. This state change occasionally does not occur right after blob association events but later. Therefore we store blob association event log in the corresponding vertex, and we localize objects using it. In particular, to establish object identities following *object split,* we

enumerate all candidate assignments, and we find the best one computing the likelihood based on the normalized Bhattacharyya distance between hue histograms.

2.3 Human Action Recognition

Action Modeling. In contrast to the process of posture recognition, which involves a specific image, action recognition involves consideration of a sequence of images. Given a sequence of images, we adapt a representation of motion history image (MHI) for the purpose of modeling an action. The MHI collapses an image sequence into a 2-D image that captures spatial and temporal information about motion [8]. The MHI is known for its fast processing speed and ability to represent short-duration movement.

An MHI at time t is updated as

$$\mathrm{MHI}_{\delta}^{t}(x,y)=\begin{cases} t/\delta & \text{if } \Psi(I^{t}(x,y))\neq 0 \\ \mathrm{MHI}_{\delta}^{t-1}(x,y) & \text{otherwise} \end{cases} \quad (1)$$

where δ is the number of images used for the collapse, $I^t(x,y)$ is the current image and Ψ signals the presence of a blob at pixel (x,y). Fig. 7 shows an example of an MHI of a person falling. The first four images from the left of Fig. 7 show extracted silhouettes, and the image on the right-hand side is the corresponding MHI.

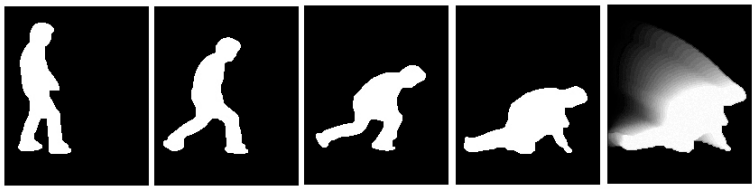

Fig. 7. Selected frames of a person falling and a corresponding MHI

Action Classifying. Given an MHI, to extract features, we search a bounding box from it. The bounding box is then normalized as a 32x32 gray image (1024 features). To reduce the feature dimension, we use PCA, and we get 80 features. We add two additional features. One is the ratio of the width to the height of the bounding box in order to classify actions such as walking and running which are similar to the normalized MHI. The other is the number of objects in the same MHI in order to consider group actions.

Fig. 8. Sample normalized MHIs (from the left: walking, running, sitting, standing, falling, punching, kicking, and turning) for the training of a multi-layer perceptron.

The task of formalizing actions is difficult because people rarely act in the same way. Hence, to classify actions, we use a multi-layer perceptron (MLP), which is a sort of robust neural network. We define eight classes: walking, running, sitting, standing, falling, punching, kicking, and turning. Fig. 8 shows examples of MHIs that are used to train an MLP. We train the MLP using 320 actions obtained from four subjects.

3 A Virtual Pet System using Augmented Reality

To evaluate the proposed system, we employed it in a virtual pet system based on augmented reality. As shown in Fig. 9, though there exist just a person and a dog's house in a real environment, the dog called 'cho-rong-i' exists together with them in the augmented view. With the multiple objects tracking result and the human action recognition result supported by the proposed system and the voice recognition, persons in the real environment can experience diverse interactions so that they feel immersion and coexistence with cho-rong-i.

Fig. 9. *(left)* the configuration and *(right)* the augmented view of the virtual pet system

Fig. 10 shows the workflow of the virtual pet system. The proposed system tracks multiple objects, and it conveys the silhouette and the bottom-center coordinate of each object in an input image to the virtual pet system. The ground of the real environment is calibrated in the initial stage so that the bottom-center coordinate of each object can be converted into the coordinate in the ground and we can obtain the depth from a camera for each object. If cho-rong-i is partially occluded by other objects, preceding objects by depth order should be re-projected after cho-rong-i is rendered like Fig. 11. Moreover, the coordinate of the ground is very useful in implementing various scenarios. Voice is recognized by the voice recognition process, where the dynamic time warping (DTW) algorithm is used. The voice recognition result and the action recognition result are used in generating cho-rong-i's actions. If

there is no user's command, cho-rong-i walks, sits, lies, sniffs, or barks according to its motion model.

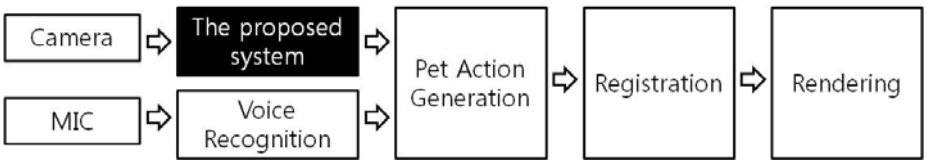

Fig. 10. The workflow of the virtual pet system

Fig. 11. Cho-rong-i is partially occluded by other objects.

We made several complex scenarios in order to demonstrate the performance of the proposed system as follow: Cho-rong-i follows only the owener among some persons, follows a rolling real ball, moves on toward the sitting person, pretends to die when someone pretends to shot, and passes between legs when a user stretchs his legs. We conducted experiments according to above scenarios, and we got successful results as below. Fig. 12 (a) shows that cho-rong-i follows only owner who is the front person in the situation where one person overlaps the other. In Fig. 12 (b), cho-rong-i follows a rolling ball and in Fig. 12 (c) it moves on toward the sitting person. While Fig. 12 (d) demostrates that cho-rong-i pretends to die when someone pretends to shot, Fig. 12 (e) shows that cho-rong-i passes between legs while a user is stretching his legs.

4 Conclusion

We proposed a real-time system that robustly tracks multiple persons and recognizes their actions through image sequences acquired from a single fixed camera, in order to enable multiple persons in a virtual environment to simultaneously and conveniently interact with virtual agents. The proposed system is compromised of three components. Given an image, we extract blobs using the Mixture of Gaussians algorithm with a hierarchical data structure and we additionally remove shadows and highlights in order to obtain a more accurate object silhouette. We then track multiple objects using a motion-based object model and an inference graph for handling grouping and fragment problems. Finally, we model an action as a Motion History Image (MHI) based on given object tracks, normalize the MHI, reduce the MHI using PCA, and classify an action using a multi-layer perceptron. To evaluate the performance of the proposed system, we employed it in an augmented reality

application where multiple persons can interact with a virtual pet. The results confirm that the proposed system can be successfully employed in interacting with virtual agents.

To enhance the proposed system, we will test for various datasets including massive objects and complex backgrounds, and the recognition of interactions between people will be considered.

Acknowledgments. This research is supported by Foundation of ubiquitous computing and networking project (UCN) Project, the Ministry of Knowledge Economy(MKE) 21st Century Frontier R&D Program in Korea and a result of subproject UCN 08B3-O4-10M.

References

1. C. Wren, A. Azarbayejani, T. Darell, and A. Pentland, "Pfinder: Real-time tracking of the human body", IEEE Trans. Pattern Anal. Machine Intell., vol. 19, pp. 780--785, 1997
2. J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer, "Multi-Camera Multi-Person Tracking for EasyLiving", IEEE Workshop on Visual Surveillance, July 2000
3. T. Henry, E. Janapriya, and L. Silva, “An Automatic System for Multiple Human Tracking & Actions Recognition in Office Environment”, IEEE Int. Conf. Acoustics, Speech, Signal Processing, 2003
4. J. K. Aggarwal and Q. Cai, "Human Motion Analysis: A Review", Computer Vision and Image Understanding: CVIU, vol. 73, no. 3, pp. 428-440, 1999
5. J. Park, A. Tabb, and A. C. Kak, "Hierarchical Data Structure for Real-Time Background Subtraction", IEEE International Conference on Image Processing, 2006
6. J. Jacques Jr., C. Jung, and S. Musse, “A Background Subtraction Model Adapted to Illumination Changes”, IEEE International Conference on Image Processing, 2006
7. B. Bose, X. Wang, and E. Grimson, “Multi-class object tracking algorithm that handles fragmentation and grouping”, CVPR 2007, June, 2007
8. A. Bobick and J. Davis, "The recognition of human movement using temporal templates", IEEE Trans. Patt. Analy. And Mach. Intell., 23(3):257-267, 2001

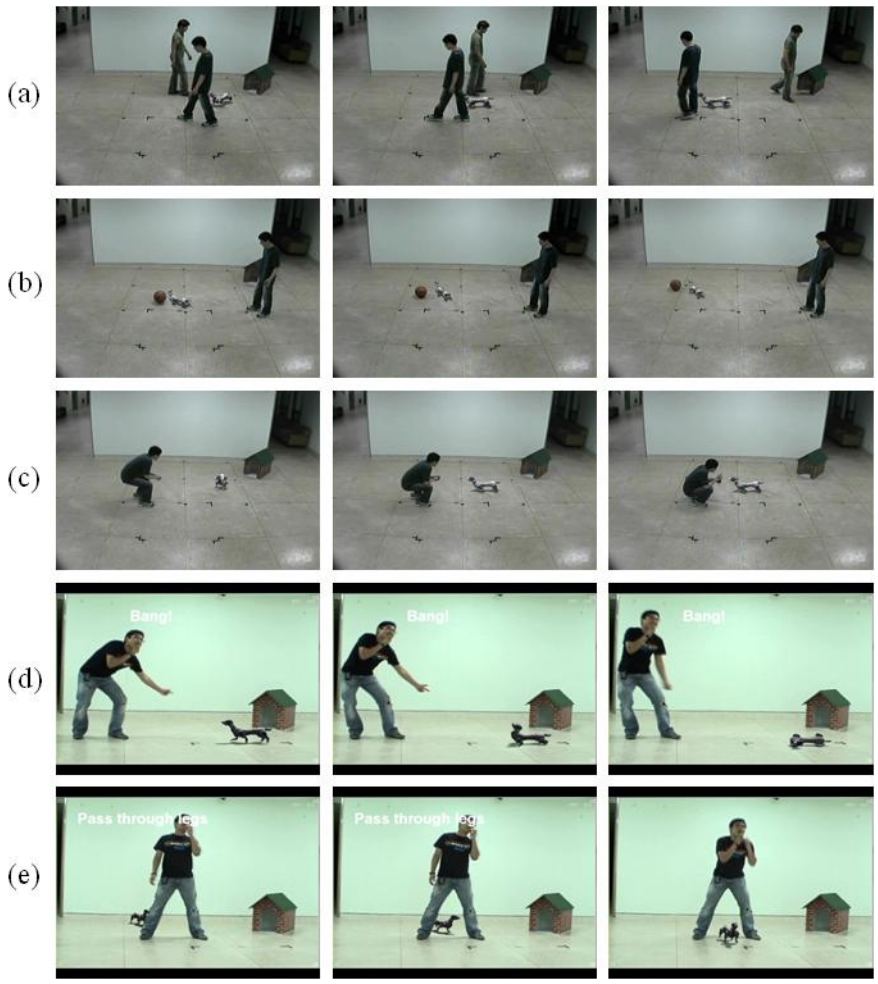

Fig. 12. The demonstration through the virtual pet system