Image-Based Automatic Detection of Indoor Scene Events and Interactive Inquiry

Kazuhiro MAKI† , Noriaki KATAYAMA† , Nobutaka SHIMADA† and Yoshiaki SHIRAI†

†:Ritsumeikan University, Shiga, Japan
E-mail :†{maki,katayama}@i.ci.ritsumei.ac.jp, {shimada,shirai}@ci.ritsumei.ac.jp

Abstract

This paper proposes a system for an automatic detection of indoor scene events with interactive inquiry based on speech dialog and gesture recognition. The system detects the events that various objects are brought in or taken out by image recognition. The user of the system inquires the stored events in the past by pointing the objects or space and using speech dialog. Since automatic event detection may fail in complicated indoor scene, the system can use interactive inquiry to correct such failures.

1. Introduction

Recently, there is a growing necessity of surveillance camera systems for security purposes. For the application of the environmental camera system, automatic detection of scene events such as human entrance or object moving in the indoor or outdoor scenes are proposed [1]-[3]. Kawamura et al.[4] proposed a system for supporting Object-finding by video recorded with a wearable camera set on a user's head. Coen [5] proposed an intelligent room that analyzes human behaviors and automatically controls facilities like light or TV. Makihara et al.[6] proposed a service robot that recognizes and brings user-specified objects. Although the recognition of the objects and the scene transition are researched in the field of computer vision, complete recognition is still difficult in full-automatic manner. There is a research that obtains information to complete a task by interacting with the user when a system makes mistakes in recognition [7]. Even when the system fails to recognize a scene event, the human user can detect and correct the failure by interacting with the system. While human has very superior recognition ability, human feels painful to repeat simple tasks for long time such as watching long term video sequences of the indoor or outdoor scenes. Moreover, human often overlooks important scenes.

To solve these problems, we develop a video surveillance system based on a novel concept of human-computer co-operation by employing verbal and gestural interaction mode. The system tries to detect the events of bringing in or taking out objects by automatic manner of image understanding and then stores the detected events to an event-database. The user of our system can use two interactive modes of gesture and speech utterance in order to inquire the stored events, which enables to directly specify the interested object or space in the real environment. Since the automatic event detection and interpretation may fail in complicated indoor scenes, the user also can use the interactive modes to correct the system's failures of understanding the scene events or to seek the overlooked scene.

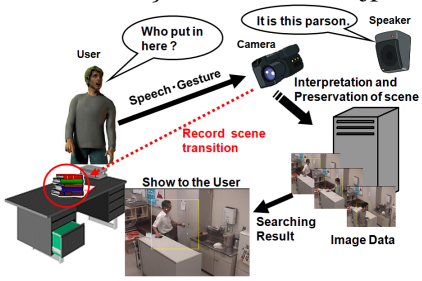

Figure 1. The concept of the system

2. System Overview

Fig.1 shows the concept of a system for an automatic detection of indoor scene events with interactive inquiry based on speech dialog and gesture recognition. The system has four modules: *human observation*, *object detection*, *event interpretation*, *interaction with user*.

Human observation module detects human face and body, and pays attention to human posture and motions. When the module detects human, it stores the images and the detected time into a human-database.

Object detection module detects the objects that human brings in and stores scenes where human brings them in or takes them out in an event-database.

Event interpretation module interprets each stored event as a scene of "bringing-in" or "taking-out" with a layer description and stores the time detected event and the event interpretation etc into the detected-event-log.

Interaction module accepts the user's inquiry of scene events. It recognizes the user-specified object or space by using the human motion and posture information obtained by human observation module, and recognizes speech input to specify a user's task by using voice recognition software "Julius/Julian" [10]. When the user inquires the desired scenes, the system searches the event-database for them and presents them to the user.

3 Automatic Detection of Indoor Scene Event

3.1 Foreground Detection

Human region and object region are detected from the camera image sequence as the foreground regions by background subtraction. Since the background often changes gradually due to calm sunlight change or rapidly due to switching the room lightings, adaptive update of the background is needed. Therefore, the system first applies the background subtraction to the current image

978-1-4244-2175-6/08/$25.00 ©2008 IEEE

(a) (b)

Figure 2. The gesture of pointing out

frame based on the current background image. The background pixels in the current image are integrated into the updated background image by a robust background maintenance algorithm [8].

3.2 Human Motion Detection

Human region is assumed to be a large area and has the face, the hair and the hand. If the region has the hair-colored region on the skin colored region in the upper part and the area is large enough, the system regards the region as a human candidate region. If the face pattern is detected on the human candidate region by using computer vision library "OpenCV" [11], the system determines the area as human region and stores the images and the detected time in a human-database.

When the system detects the human, it tries to detect the hand and the tip of a finger. The skin colored regions away from the face is determined as the hand and the pixel most distant from the face as the tip of a finger. When the user inquires the desired scenes with voice, the system checks whether the user presents a pointing gesture. If the tip of a finger is away enough from the center of human region (Fig.2-(a)), the system recognizes that the human is pointing a certain place in the real space. If the pointing gesture is detected, it determines the region near the tip of a finger as the user-specified region and sends the position of the user-specified region to the interaction module. If human points at his front(Fig.2-(b)), the system obtains the tip of finger and determines the user-specified region when the user inquires with the voice.

3.3 Object Detection

The detected foregrounds excluding the human region include objects, object traces [1], and noise-perturbed regions. The system detects the objects and object traces by using the time-series data of the regions detected by background subtraction. The algorithm to detect the object is as follows.

1. Collect the pixels detected more than 8 frames in the past 10 frames.
2. If the number of the collected pixels is large enough, the system determines the collected pixels as the object or the object trace.
3. Store the images of past 10 frames from the detected frame into an event-database and send the detected region to interpret the detected event.
4. The detected region's pixels are integrated into the background image immediately, because the system detects newly appeared objects only.

[1] When human or an object moves away, the background gets visible again on that place, which can be also detected by background subtraction. Here we call that "object trace".

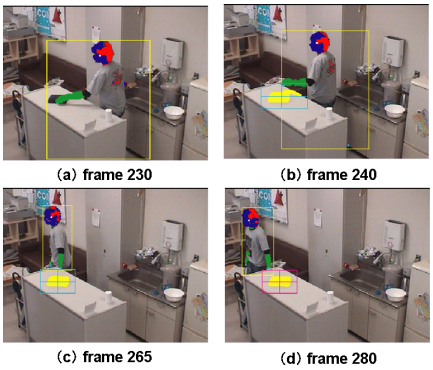

Figure 3. Detection of object brought in

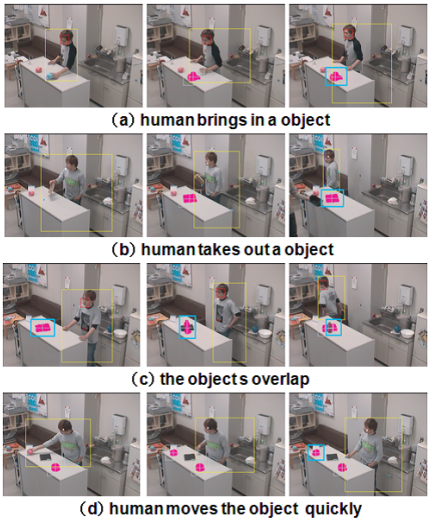

Figure 4. Typical example of detected scene

Fig.3 shows an example of detecting an object which was brought in. Fig.3-(a) shows an image of bringing in an object. At this moment, the system cannot detect whether the object is put, because the human region includes the object. Fig.3-(b) and (c) show the moment of detecting the region of object candidate (i.e. the object is separated from the human region). The system continues to observe the detected region as the region of object candidate at these frames since the candidate region has not been observed for a certain period yet. In fig.3-(d), the system finally detects the object because it has been observed the object candidates for enough frames.

3.4 Experiment of Event Detection

We made an experiment to detect the scene events. We ran the event detection system in 24 hours and continued the load test for a week. In current implementation, the system captures 6 frames per second.

The number of detected event is 667 and the system stored 10 images per event. The detected scenes are classified by human, and typical scenes are shown in Fig.4. Fig.4-(a) shows human brings in one object. Fig.4-(b) shows human takes out another object. The system correctly detects the single objects. Fig.4-(c) shows the scene in which multiple objects overlap on the image. A left image of Fig.4-(c) shows human puts one object. A center image of Fig.4-(c) shows human puts a new object in front of the object. A right image of Fig.4-(c) shows human takes out the one behind. Even if the multiple ob-

jects overlap, the system correctly detects the object region. Fig.4-(d) shows human moves an object quickly. In that case, the system detects the object trace region by taking out the object. The detection rate of objects is almost 80%.

4 Interpretation of Scene

The event interpretation module interprets the detected events as "bringing-in" or "taking-out" by analyzing textures and shapes of the object region or object trace region. Then it stores event indexes into a detected-event-log. The event indexes include the shape and texture of the objects (What), the position of the objects (Where), the detected time (When), and the event interpretation (How). Additionally the system can make the index "Who" by an automatic face recognition module or interaction with the user.

Fig.5 shows the example of the event and the concept to interpret the scene event. First human brings in an object in Fig.5-(a). Next the system obtains the object region in Fig.5-(b). Then human takes out the object in Fig.5-(c). Finally the system gets the object trace region in Fig.5-(d). When Fig.5-(b) is compared to Fig.5-(d), the object region registered in Fig.5-(b) and the object trace detected in Fig.5-(d) are the same shape. Moreover the texture of background registered in Fig.5-(b) and that detected in Fig.5-(d) is same. Therefore, the system decides whether the scene is "bringing-in" or "taking-out" by the shape of detected object regions or object traces and the texture of background. If the object trace and the object region are same shape and the texture of background can be expected (the background before the object was put), the scene is assumed to "taking-out". If not, the scene is assumed to "bringing-in".

In actual situations, the shape of the object may not be detected perfectly because the mutual occlusion of the objects frequently occurs. In order to treat the object occlusions, the layered detection method [9] is employed. In order to appropriately treat multiple object occlusions, we represent each detected object region as a layer in a layered scene description stack structure so that the system predicts the change of the layered stack when the object is taken away.

Let consider a situation shown as Fig.6-(a) where object-1 was put first, next object-2 was put in front of object-1, and then object-3 was put on the place which is in front of object-1 and behind object-2. In this situation, the system predicts "next state" of each object; i.e. the shape and texture of the region to appear when taking out the object and expects the background subtraction result to be detected. Fig.6-(a) shows the "next state" of each object. If the detected region (shape and texture) is identical to "next state" of object-1, it recognizes that object-1 is taken out. If no state matches to the detected region it recognizes that a new object is brought in.

When object-2 is taken out (Fig.6-(b)), the system observes the unseen part occluded by object-2 ("?"-marked region in Fig.6-(b)). Then the system treats the region as "unconfirmed region", because the system cannot recognize whether the region is a part of object or one object or a background. In Fig.6-(b) situation, the system expects the following situation will happen next:

1. object1 will be taken out

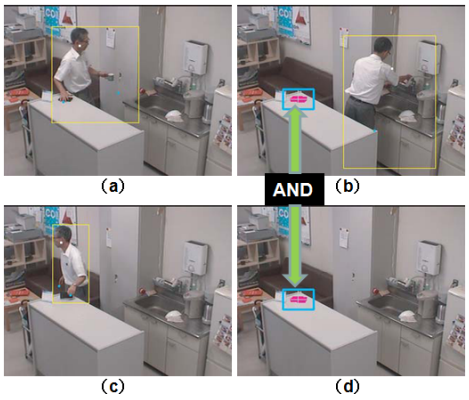

Figure 5. Example of "bringing-in" and "taking-out"

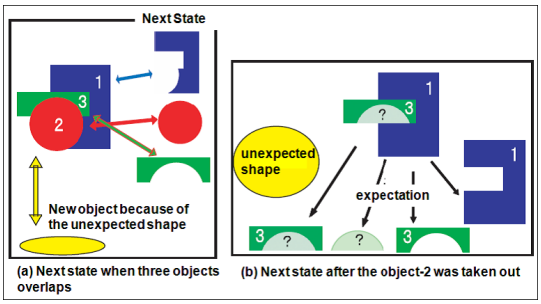

Figure 6. The concept of layer description

2. object3 will be taken out
3. the "unconfirmed region" will be taken out
4. object3 and "unconfirmed region" will be taken out

The algorithm of interpretating the scene event by using this layer description expecting next states is as follows:

- Match the shape of the region detected by object module and the shape of "next state" of all layers.
- If the coincident region is found in the expected next states, the matched object is interpreted as "taken-out" and the system deletes the layer from the current layer description stack.
- If the system deletes the layer in Step2, the system proceeds to Step5.
- If the compared shape is not coincident, the detected region is interpreted as "bringing-in" and is added as a new layer to the current layer description stack.
- Update the expectations of the "next state" for all layers.

Fig.7 shows the result when the above algorithm is applied to an occluding scene in a real environment.

5. User Interface

Fig.8 shows the concept of searching the desired scene. The user asks the detected-event-log for the desired scene by giving the system the information of spatial position, type of events ("bringing-in", "taking-out") or date / time of the scene. First, the system searches for

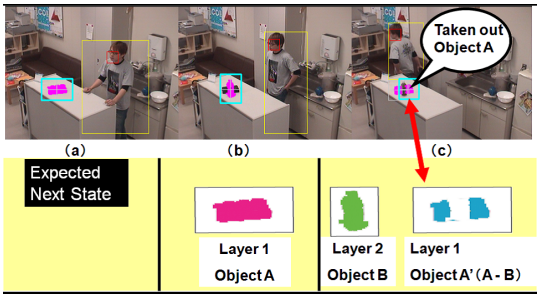

Figure 7. The real scene of multiple overlapping object

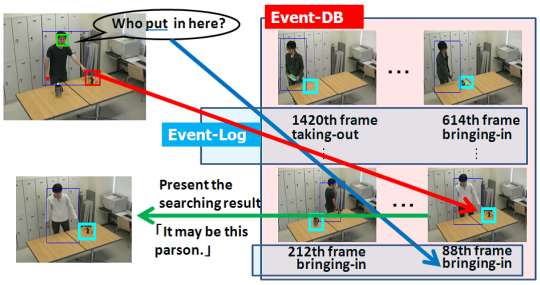

Figure 8. The concept of searching the desired scene

candidate scenes that happened around the user-specified position, which is obtained from the human observation module. Then, the candidates are also limited by type of events or date / time specified by the keyword given by the user speech via voice recognition. Finally it presents the best candidate and requires the user to check it with the desired scene by speech mode. If the user indicates it is not the desired scene, the system continues showing another candidate and requiring user to check it until the user finds the desired scene. Since the user can understand the presented scene correctly, even if the event detection module detects a false event and shows it as a result, the user can tell the system that the presented scene is misunderstood and advise the system to correct it. This advice may help the system correct other false event interpretations related with the event.

The full-automatic event detection occasionally overlooks some scene events. However, there may be the desired scene in the human-database if human entrance was detected at that scene. Therefore, the system can seeks it from the human-database by using interactive inquiry with the user. Fig.9 shows an example of searching the overlooked event based on such user-interactions in both gestural and verbal modes. The user can find the scene about the pointing object and additionally the system can correct the scene interpretation based on the user's advice.

6. Conclusion

We proposed a system that detects and interprets the scene events, and develops the inquiry system user can inquire easily by verbal and gestural interaction. In addition, we proposed the interactive inquiry to find the overlooked events. Future works are to recognize whether human really has the objects, to recognize what human did and to increase the events that the system recognizes.

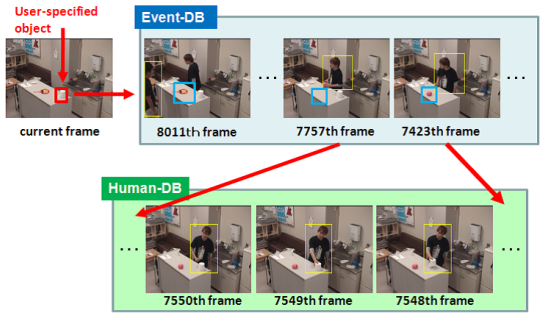

Figure 9. Example of searching the overlooked event

References

[1] Kazumasa Yamazawa and Naokazu Yokoya: "Detecting moving objects from omnidirectional dynamic images based on adaptive background", Proc. 10th IEEE Int. Conf. on Image Processing, Vol.III, pp. 953-956, 2003.

[2] Kosaku Matsui, Reiko Hamada, Ichiro Ide, Shuichi Sakai: "Indexing of surveillance video based on object relocation (in Japanese)", Proc. IPSJ 67th Bi-Annual Convention, 4K-3; Vol.3 pp79-80, 2005.

[3] R. Collins, A. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver,N. Enomoto, and O. Hasegawa: "A system for video surveillance and monitoring: VSAM nal report", In Technicalreport CMU-RI-TR-00-12, Robotics Institute, CMU, May 2000.

[4] Tatsuyuki Kawamura, Takahiro Ueoka, Yasuyuki Kono, Masatsugu Kidode: "Evaluation of View Angle for a First-person Video to Support an Object-finding Task", The 5th Int. Conf. of the Cognitive Science, 2006.

[5] Michael Coen:"The Future of Human-Computer Interaction or How I learned to stop worrying and love my Intelligent Room", IEEE Intelligent Systems, vol.14, no.2, pp.8-19,1999.

[6] Y. Makihara, M. Takizawa, Y. Shirai, and N. Shimada: "Object Recognition under Various Lighting Conditions", Proc. of 13th Scandinavian Conf. on Image Analysis, pp899-906, 2003.

[7] Y. Makihara, J. Miura, Y. Shirai, and N. Shimada: "Strategy for Displaying the Recognition Result in Interactive Vision", Proc. of 2nd Int. WorkShop on Language Understanding and Agents for Real World Interaction, pp.467-474, 2005.

[8] H. Shimai, T. Mishima, T. Kurita, S. Umeyama: "Adaptive background estimation from image sequence by on-line M-estimation and its application to detection of moving objects", Proc. of Infotech Oulu Workshop on Real-Time Image Sequence Analysis, pp. 99-108, 2000.

[9] Hironobu FUJIYOSHI, Takeo KANADE: "Layered Detection for Multiple Overlapping Objects", IEICE TRANSACTIONS on Information and Systems Vol.E87-D No.12 pp.2821-2827, 2004.

[10] "Julius -an Open-Source Large Vocabulary CSR Engine-", http://julius.sourceforge.jp/

[11] "Open Source Computer Vision Library OpenCV", http://www.intel.com/technology/computing/opencv/index.htm