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. he 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 cameras systems for security reasons. 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], [2]. Kawamura et al. [3] proposed a system for supporting Object-finding by video recorded with a wearable camera set on a user's head. Makihara et al. [4] 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, the complete recognition is 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 [5]. 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 uses 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 find the overlooked scene.

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 is composed of a PC (Pentium4 2.66GHz), a large storage device to store the video sequences, a network camera (SONY EVI-D100) set up in the ceiling, a speaker, and a microphone. The system has four modules:

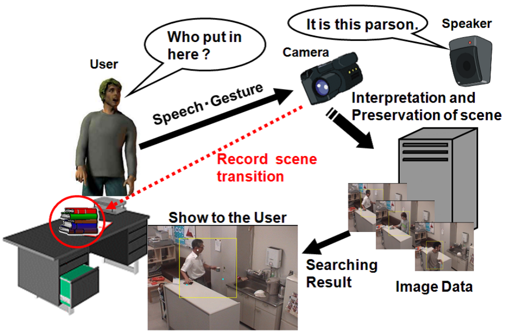


Fig. 1: The concept of the system

- 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 human observation module detects human, it stores the images and the detected time in a human-database.

Object detection module detects the objects human brings in by using the detected human motion and posture and stores scenes where human brings them in or takes them out in an event-database.

Event interpretation module interprets the stored events as a scene of "bringing-in" or "taking-out" by understanding the object overlapping with a layer description and stores the time detected event and an 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" [8]. 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

For all scene event detection and interpretation, human region and object region should be detected from the camera image sequence. Those regions can be detected 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 in order to keep the background appropriate. Therefore, the system first applies the background subtraction to the current image frame based on the current background image. Then the current image is divided into the foreground region such as human or newly appeared objects and the background region. The background pixels in the current image are integrated into the updated background image by a robust background maintenance algorithm [6].

Once the foreground is well-detected, it should be divided into human, objects, object traces[1] and noise region (including intensity-changed regions due to illumination and shadowing or noise-perturbed region) by the methods of the following sections.

3.2 Human Motion Detection

Human region is assumed to be the 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" [8], the system determines the area as human region and stores the images and the detected time in a human-database. The human-database is used to provide information to the user and to search the overlooked scenes. 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.

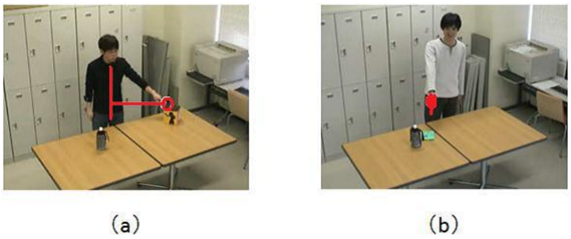


Fig. 2: The gesture of pointing out

3.3 Object Detection

The foregrounds excluding the human region detected by human observation module include objects, object traces, and noise regions. To eliminate the noise 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 including the current frame.

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

[1] When human or an object moves away from a certain place in the image, the background becomes to visible again on that place, which can be also detected by background subtraction in addition to the moving object itself. Here we call that "object trace", and noise regions (including intensity-changed regions due to illumination and shadowing or noise-perturbed region).

the detected region to the interpretation module 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.

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.

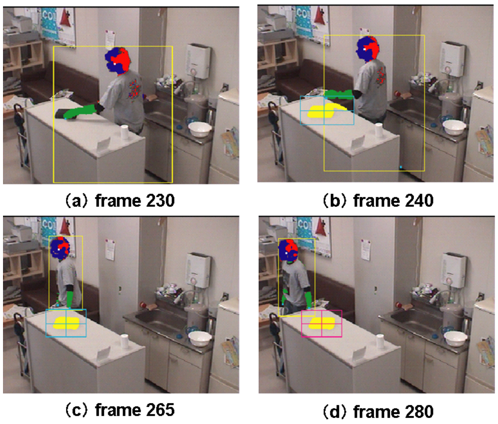


Fig. 3: Detection of object brought in

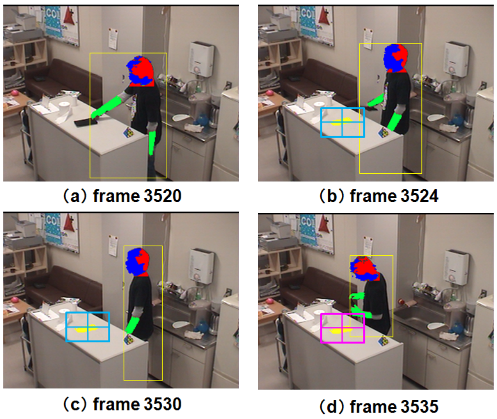


Fig. 4: Detection of object taken out

Fig.4 shows an example of detecting an object which was taken out. Fig.4-(a) shows human taken out an object. Fig.4-(b) and (c) show detecting the regions of object candidate. In the frame of Fig.4-(d), the system detects the object trace. Even if human passes in front of the object, the system can detect the object with stability by observing for a certain period to recognize the object.

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. Fig.5 shows the place used by the experiment. There are a sink, a refrigerator, and a coffee maker etc. therefore the human visits this place frequently.

Fig. 5: The place used by the experiment

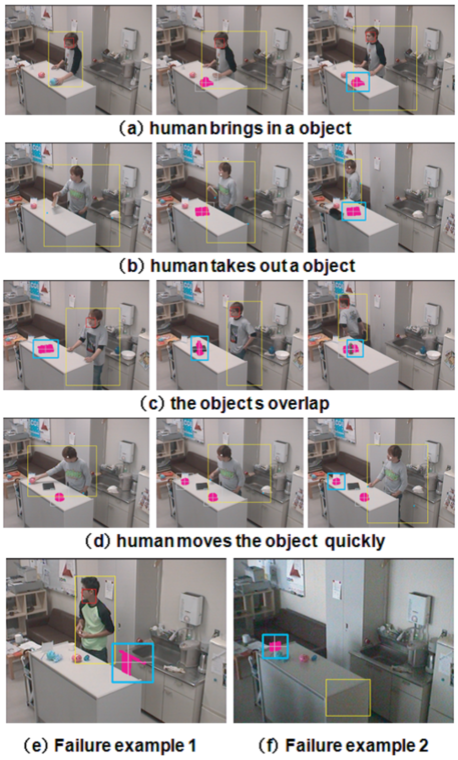

(a) human brings in a object
(b) human takes out a object
(c) the objects overlap
(d) human moves the object quickly
(e) Failure example 1
(f) Failure example 2

Fig. 6: Typical example of detected scene

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.6. Fig.6-(a) shows human brings in one object. Fig.6-(b) shows human takes out another object. The system correctly detects the single objects. Fig.6-(c) shows the scene in which multiple objects overlap on the image. A left image of Fig.6-(c) shows human puts one object. A center image of Fig.6-(c) shows human puts a new object in front of the object. A right image of Fig.6-(c) shows human takes out the one behind. Even if the multiple objects overlap, the system correctly detects the object region. Fig.6-(d) shows human moves an object quickly. In that case, the system detects the object trace region by taking out the object. Additionally, the system detects the object region by putting the object. These two regions are detected at the same time.

The detection rate of objects is 80%. Many detected scenes were movements that poured the drink from a refrigerator into the glass and took it away. This scene was able to be detected by almost 100%, because the object was separated from human when human puts the glass and opens a refrigerator. The false detection rate (the number of false detection / the number of total detection) is 50%. A lot of falsely detected scenes include shadows as objects (Fig.6-(e)) and the lights such as rising sun as objects (Fig.6-(f)). These false-detections are corrected by the interaction with the user.

4 Interpretation of Scene

The events detected by object detection module include the scenes of bringing in or taking out objects. Therefore, 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. Additionally, the event interpretation module stores event indexes into a detected-event-log. The event indexes include "When, Where, Who, What". In concrete saying, it is the area of objects, the center of objects (Where), the detected time (When), event interpretation (What). Although the current system does not automatically identify the person that makes the events, if the user can recognize him to watch the scene, the system can make the index "Who" by interaction with the user.

4.1 Identification of "bringing-in" or "taking-out"

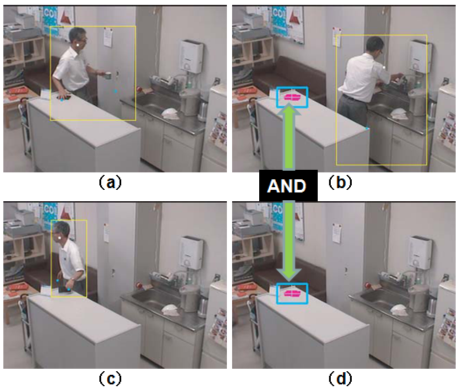
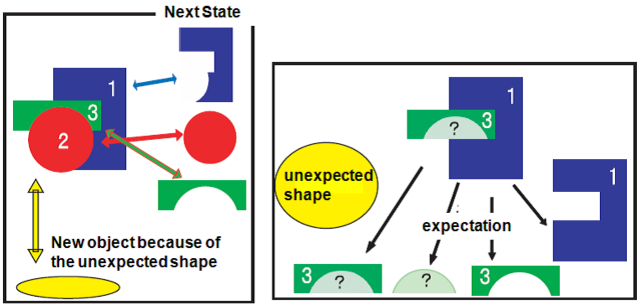


Fig. 7: Example of "bringing-in" and "taking-out"

Fig.7 shows the example of the event and the concept to interpret the scene event. First human brings in an object in Fig.7-(a). Next the system obtains the object region in Fig.7-(b). Then human takes out the object in Fig.7-(c). Finally the system gets the object trace region in Fig.7-(d). When Fig.7-(b) is compared to Fig.7-(d), the object region registered in Fig.7-(b) and the object trace detected in Fig.7-(d) are the same shape. Moreover the texture of background registered in Fig.7-(b) and that detected in Fig.7-(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 is 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".

4.2 Occlusion of the object

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 [7] is employed. However this method cannot be applied to the scene with the occlusions. For example, when human takes out the object behind an object, it does not recognize this scene. Therefore, we represent the detected object region as a layer structure and the system predicts the change of the layer description when the object is taken away. Fig.8 shows the concept of layer description. Let consider a situation shown as Fig.8-(a) where an object-1 was put first, next an object-2 was put



(a) Next state when three objects overlaps

(b) Next state after the object-2 was taken out

Fig. 8: Concept of layer description

in front of the object-1, and then an object-3 was put on the place which is in front of the object-1 and behind the object-2. In this situation, the system expects "next state" of each object. We define the shape and texture of the region to be expected to appear by taking out the other object as the "next state". The system generates the "next state" for each object by using the object regions detected so far. Fig.8-(a) shows the "next state" of each object. If the system detects the region (shape and texture) identical to "next state" of the object-1, it recognizes that the object-1 is taken out. Similarly, the system detects the region (shape and texture) identical to "next state" of the object-2 or object-3, it recognizes that the object is taken out. If the system detects the region excluding the "next state" of each object, it recognizes that new object is brought in. Fig.8-(b) is the situation in which the object-2 was taken out in Fig.8-(a). When the system expects the "next state", the system obtains the region (shape and texture) that was not registered in past. The system determined 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.8-(b) situation, the system expects the following situation will happens next.

- object1 will be taken out
- object3 will be taken out
- the "unconfirmed region" will be taken out
- object3 and "unconfirmed region" will be taken out

The system expects the "next state" of all situation so that the system can handle taking out of any object currently in the field of view.

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

1. Match the shape of the region detected by object module and the shape of "next state" of all layers.
2. If the coincident region is found in the expected next states, the object of the matched layer is interpreted as "taken-out" and the system deletes the layer from the current layer description stack.
3. If the system deletes the layer in Step2, the system proceeds to Step5.
4. 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.
5. Update the expectations of the "next state" for all layers.

4.3 Example of event interpretation

Fig.9 shows the scene of multiple overlapping objects and the expected "next state". Fig.9-(a) shows human brings ObjectA in. ObjectA is added as layer1 and the system expects the "next state" of ObjectA. Fig.9-(b) shows human brings ObjectB in front of ObjectA. ObjectB is added as layer2, because the region of ObjectB is not coincident with the expected "next state" of ObjectA. The system expects the "next state" of ObjectB and that of ObjectA in consideration of ObjectB, because the layer of ObjectB is in front of that of ObjectA. Fig.9-(c) shows human takes out Object A behind ObjectB. The system can recognize that ObjectA is taken out, because the detected region is coincident with the expected "next state" of ObjectA.

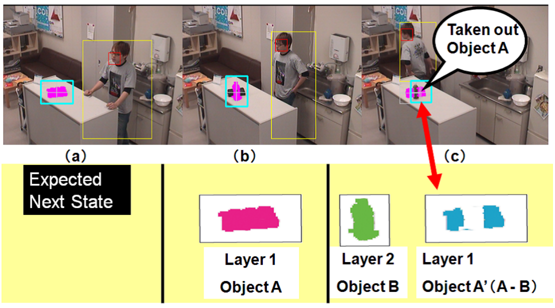


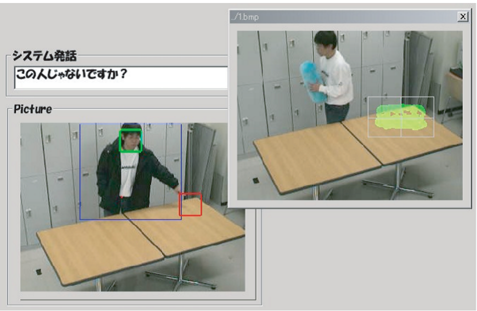
Fig. 9: The scene of multiple overlapping objects

5 User Interface

5.1 Structure of speech dialog interface

The interaction with the user uses the detected-event-log stored by event interpretation part. To search the desired scene from the event-database by using the task, the time, and the user-specified position, the detected-event-log was bound to the images stored in the event-database. The spacial position specified by the user is given by the human observation module. The inquired task and the time are specified by the keyword given by the user speech via voice recognition. There are two kinds of the tasks the user can inquire: "Who put the object?" and "Who took out the object?". When the user inquires them, the system searches the detected-event-log for the desired scene based on the specified kind of task, the time, and the spacial position. Next, the system requires the user to confirm the search results by presenting the retrieved images or a movie of the candidate scene. Then the user chooses the desired scene among them and also can find the false detection by object detection module and eliminates them from the candidates.

We implemented the event inquiry module using the speech and gestural modes into our scene event detection system. Fig.10 shows an operation screen of the inquiry system. In fig.10, the user inquires in the real environment by pointing and voice, and the system presents the searching result.



この人じゃないですか？

Fig. 10: An operation screen of inquiry system

5.2 Proposal of interactive inquiry to find the overlooked events

The full-automatic event detection occasionally overlooks some scene events. If the user wants the overlooked event and inquires to the system, the system does not find it from the event-database. However, there may be the desired scene in the human-database if human entrance was detected at that scene. Therefore, the system searches it from the human-database by using interactive inquiry with the user. Fig.11 shows an example of searching the overlooked event. First, the user inquires to the system about the object put in current frame. Next, the system searches the event-database for the desired scene. However the system does not find the desired scene and asks the frame to search the scene from the human-database to the user by presenting the images stored in the event-database. In fig.11, the user-specified object has already existed at the frame The system showed to the user the images from preserved in the human-database. As a result, the user was able to find the scene around the frame Additionally, the system can recognize the scene by user's advice and store it in the event-database.

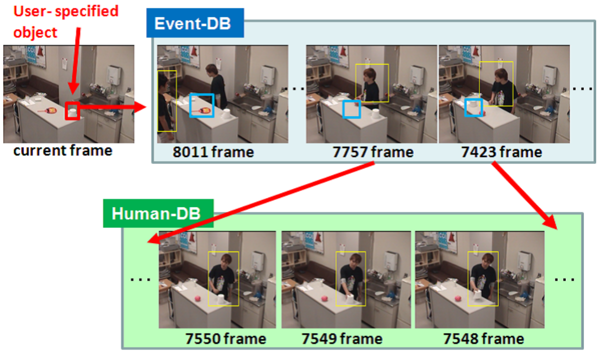


Fig. 11: Example of searching the overlooked event

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 propsed the interactive inquiry to find the overlooked events. Future works are as follows.

- Recognize whether human really has the objects.
- Recognize what human did.
- Increase the events that the system recognizes.

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. , pp. 953-956, 2003.9

[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, Mar 2005

[3] 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 International Conference of the Cognitive Science, July 2006

[4] 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, Goteborg, Sweden, Jul 2003

[5] 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, Singaporu, Singaporu, Nov. 2005

[6] 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, October, 2000

[7] Hironobu FUJIYOSHI, Takeo KANADE: "Layered Detection for Multiple Overlapping Objects", IEICE TRANSACTIONS on Information and Systems Vol.E87-D No.12 pp.2821-2827, 2004.12

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

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