Interactive Inquiry of Indoor Scene Transition with Awareness and Automatic Correction of Mis-understanding

Kazuhiro Maki[1], Nobutaka Shimada[1], and Yoshiaki Shirai[1]

Dept. of Human Computer Intelligence, Ritsumeikan University, 1-1-1 Nojihigashi, Kusatsu, Shiga, Japan. {maki, shimada, shirai}@i.ci.ritsumei.ac.jp

Abstract. It is difficult to perfectly recognize scenes and objects by an automatic manner of current image understanding techniques. In recognition of scene transition, one false scene recognition may cause other faults. We propose a method to be aware of the existence of the false scene recognition from user's suggestions via an interactive inquiry system. Furthermore, the system not only corrects the false recognition but also verifies other scene recognitions related to the corrected scene and corrects them if needed. Experiments on the system with the proposed method have shown that the system can find false scene recognitions and maintain the scene descriptions and event recognition correct by combining minimal correctional advise by the user with its automatic propagation to other related scenes.

1 Introduction

Recently, there is a growing necessity of surveillance camera systems for security purposes. Applications of the environmental camera system like 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 through a wearable camera set on a user's head. Coen [4] proposed an intelligent room that analyzes human behaviors and automatically controls facilities like illuminations and TV. Makihara et al.[5] proposed a service robot that recognizes and brings user-specified objects.

Although the recognition of the objects and the scene transition are being researched in the field of computer vision, complete recognition is still difficult in full-automatic manner. Even for the difficult scenes system fails to recognize, the human user can detect and correct the failure by interacting with the system. There is a research based on such a concept that the system gathers information to complete a task by interacting with the user when the system makes mistakes in recognition [6].

While human has very superior recognition ability, human feels painful to repeat simple tasks such as watching long term video sequences through the monitors. Moreover, human often overlooks important scenes. For human perception it seems more intuitive and comfortable to specify objects by directly pointing or even handling them *just on site*. If the on-site-users can make such gestural and speech queries for suspicious objects just in front of their eyes like, "When is this brought in here?" or "Who took away my book from here?", they can keep their environments neat and secure by themselves, without outside monitor watchers.

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

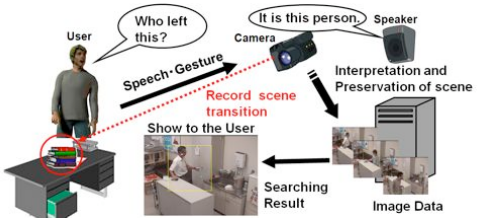

Fig. 1. The concept of the system

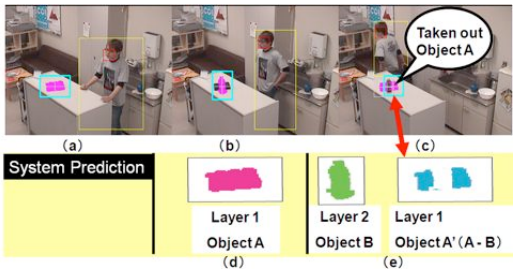

Fig. 2. The real scene of multiple overlapping object

order to inquire the stored events, which enable to directly specify the interested object or space in the real environment.

While a event query system where the user makes queries interactively and repeatedly[7], it requires complete recognition of the scenes and user inputs with no mistakes. Automatic image recognition may often generates incomplete information by mistakenly understandings or detection failures. Moreover one mis-understanding may causes another failures of recognition for the successive scene events since each event recognition is done by considering the scene transition in time elapsed. For building more robust and useful query system, therefore, the ability to be aware of mis-understandings of automatic recognition and correct them in order to maintain the recognized event and the scene transitions are consistent.

This paper proposes a method being aware of the existence of the system's mis-understandings based on the suggestions the user makes in the scene queries. Furthermore, the system not only corrects the mis-understandings but also verifies other scene recognitions related to the corrected scene and corrects them if needed. The user makes queries just for the scene he/she wants and then the system detects the false recognitions if they exist, and efficiently corrects the multiple scene transitions. In that way, the system can find false automatic recognitions and maintain the scene descriptions and event recognition correct by combining minimal correctional advise by the user with automatic propagating the correction to other related scenes.

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: *event detection*, *event interpretation*, *interaction with user*.

Event detection 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-DB". When the module detects the objects that human brings in and stores scenes where human brings them in or takes them out in an "Event-DB".

Event interpretation module interprets each stored event as a scene of "bring-in" or "take-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

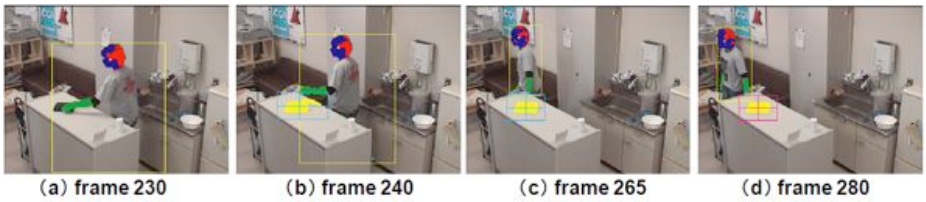

Fig. 3. Detection of object brought in

human observation module, and recognizes speech input to specify a user's task by using voice recognition software "Julius/Julian" [12]. When the user inquires the desired scenes, the system searches the "Event-DB" for them and presents them to the user.

3 Recognition of Indoor Scene Transition

3.1 Automatic Detection of Scene Event

The system detect indoor scene event by using the method [8]. First, it detects foreground regions from the camera image sequence by robust background for illumination changes [9].

Next, it removes shadow regions from foreground regions [10] and detects human region. Human are detected by the largest linked foreground region with face, hair, hands. When the system detects human, it stores the images and the detected time in the "Human-DB". Then, Non-human regions are the region of object candidate.

Finally, the object candidates that stays steadily for some periods will be the object to be detected. When the system detects objects, it stores the images of past 10 frames from the detected frame into an "Event-DB".

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.

3.2 Interpretation of Scene Event

The event interpretation module interprets the detected events as "bring-in" or "take-out" by analyzing textures and shapes of the object region or object trace region. Then it stores event indexes into a detected-event-log.

Event Interpretation Based on Predicting Scene Transition The system decides whether the scene is "bring-in" or "take-out" by the shape of detected object regions or object traces [1] and the texture of background [8]. 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 "take-out". If not, the scene is assumed to "bring-in".

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.

[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".

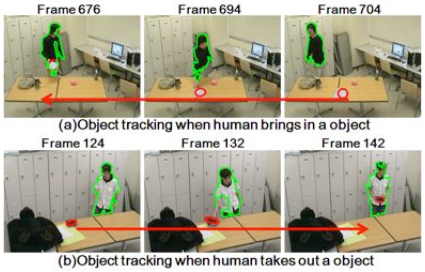

Fig. 4. Verification Using Human Region

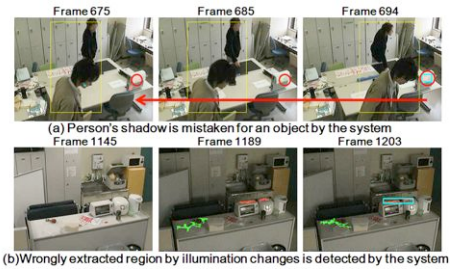

Fig. 5. Rejection of False Detections

Fig.2 shows the result when the above algorithm is applied to an occluding scene in a real environment.

Fig.2-(a): object A is initially placed. At the moment, since the existing object is only one, the system prediction of the expected object trace (image subtraction) when removing A is just the same as the shape of original A(Fig.2-(d)).

Fig.2-(b): then object B is placed in front of A. Since the object trace does not match with the prediction (d), this will be added as a new object. Since B is placed after A is placed, B is added to the 2nd layer over A's layer. While the observation prediction for removing B is the just same as the shape of B since B is on the top layer, that for removing A is modified to the shape A', the shape of A minus B (Fig.2-(e)).

When A had been taken out, the observed shape do match to A', and it can be successfully recognized as "A is taken out".

Event Verification Using Human Body Detection While most of the scene transitions in generic in-door scenes are caused by human's object handling, illumination conditions or shadow effects sometimes causes the false image change. Such false change is not predictable by the layered scene description model and may be treated as the new object brought in the scene. Our system additionally employs the result of human body detection in order to verify whether the detected image change is really 'bring-in' or 'take-out' of objects.

When the scene change is assumed to 'bring-in' by matching with the layered scene description, the system rewinds the image frames from the detected frame and tries to track back the detected object region by using CamShift algorithm[11]. Then the system verifies whether the detected human region and the tracked object region join together, and if so, it is determined as 'bring-in'. When the scene is assumed to 'take-out', the system tracks the detected object forward from just before the detected frame and does the same verification.

Fig.4-(a) shows that an object is detected at the 704th frame and it is tracked back to the past frames. In the 676th frame, the tracked object and the detected human region joins together and the scene is determined as 'bring-in'. In contrast Fig.4-(b) shows that an object is detected at the 142nd frame and it is tracked forward from the 124th frame, a little bit before the detection. the object and the human region joins together at the 142nd frame and the scene is determined as 'take-out'.

By employing this manner, the system can treat properly the mis-detection scene shown as Fig.5. In Fig.5-(a), a shadow region of the human is mis-detected as a newly brought object at the 694th frame. In Fig.5-(b), illumination change are falsely detected at the 1203rd frame. Since the detected objects in both cases never joins with the human region, the scenes are determined to false detections.

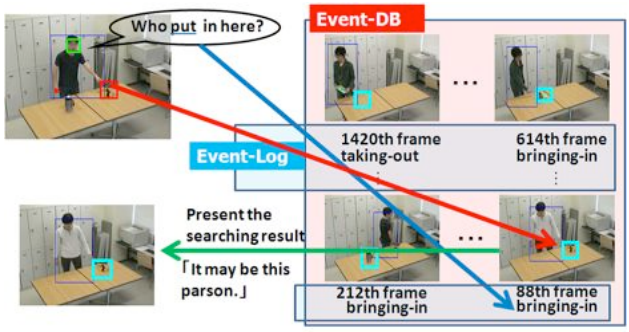

Fig. 6. The concept of searching the desired scene

3.3 Performance Evaluation of Event Detection and Interpretation

An experimental result of the performance evaluation of the automatic scene event detection and interpretation ability is shown here. In the experiment, we targeted to a kitchen space where several persons often visited repeatedly and a resting space near windows strongly affected by the illumination change from outside. There we captured and analyze about 230 thousand image frames, total 32GB amount, during about 5 hours.

The automatic scene interpretation gave results that 93% of the event to be detected were successfully detected, the rest 7% were not detected because of the object color was similar to the background's, rejected as shadow regions or too small to detect. 23% of all detections were mis-interpreted because of mis-detection of the part of human body or shadow, or human body just passing through was mistakenly detected and tracked as an object. 83% of the correctly detected scene were successfully interpreted.

4 Detection and Correction of Mis-understandings Using Progress of Query Interaction

The automatic image understandings includes mistakes in high possibility. When the user desires such not-detected or mistakenly understood scenes, the system cannot present those scenes based on the automatic functions only.

Even if all scene candidates are presented the scene the user wants may not be found because of the mis-understanding or mis-detecting scenes. Considering that human's perception ability is so high level, the system should recognize that the missing queried scene may be mis-understood by the automatic recognition. Therefore when the system detects such possibility of mis-understandings, the system switches the search mode from the automatic way to an interactive mode which presents the scene candidate possibly misunderstood to the user and specifies the scene by employing the suggestions the user makes in the scene queries.

4.1 Scene Inquiry Using Speech and Pointing Gesture

Fig.6 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 (”bring-in”, ”take-out”) or date / time of the scene. First, the system searches for 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.

4.2 Interactive Identification of Mis-understood Scene

The mis-understandings the system can make are classified into two categories: not-detected and mis-interpreted. When the system notices the existence of the mis-understandings, the system cannot immediately make judgment which category of mistake occurs. The system once assumes that both mistakes occur and searches for the candidate of the queried scene. Then the system displays each candidate scene and asks the user to confirm. The system identifies the scene based on the given confirmation and the user's additional suggestions.

Identifying Not-detected Scene Since the queried scene is not found in "Event-DB" in case of 'not-detected' mis-understanding category, the system attempts to seek the scene in "Human-DB" where the image sequences including the detected human activities are registered from the most recent event to the past backward. For more high sensitivity to detect, the detection conditions looser than Sec.3.1 are employed:

- lower threshold for image subtraction
- not employing rejecting shadow region.
- taking into account smaller or larger regions.

The scenes already detected and registered in "Event-DB" are doubly detected but such scenes are rejected.

The seeking job is started in background process and the system asks the user the following time information during the scene seeking:

- (Querying the object existing in the current scene): the most recent date and time when the user is convinced that the object had not existed there yet
- (Querying the object missing in the current scene): the most past date and time when the user is convinced that the object had already existed there.

The secondary seeking is simultaneously started forward from the given date and time to the current time. The concurrent searches (forward and backward) are done for the time efficiency and high response performance. Additionally, when the user specifies the place of the object by pointing gesture, the seeking is limited around the specified region.

Each of newly found candidate scenes is presented to the user as a motion movie and the system asks the user to confirm it. If that is the scene the user intended, the scene is registered into "Event-DB", otherwise the system asks the user whether the intended object is seen in the movie or not. The answer helps limit the next search area.

Identifying Mis-interpreted Scene In case of 'mis-interpreted' mis-understanding category, the queried scene should be found in "Event-DB". Therefore, the system displays each scene registered in "Event-DB" one to another and asks the user to specify the intended scene. As the same way as treating the 'not-detected' case, the system first asks the user to give date and time information, and then presents the scenes detected after that time and interpreted as the same event either 'bring-in' or 'take-out'. If the desired scene is not found in spite of presenting all the scenes, the system considers the event was mis-interpreted: 'bring-in' as 'take-out' or vice versa, and then presents the scenes interpreted as the other event.

When the user is asking about the existing objects (i.e. seeking 'bring-in' events) and that object is mistakenly interpreted as already taken away, the scene of bringing it in is

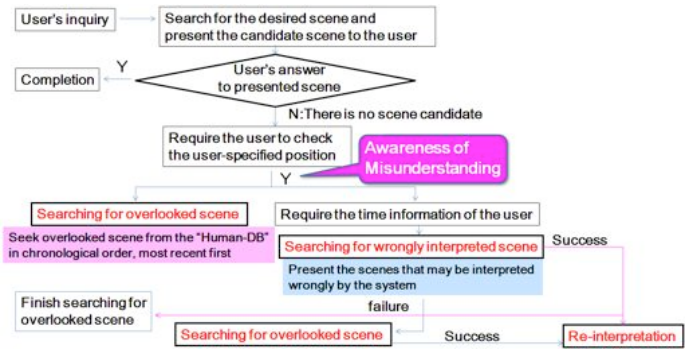

Fig. 7. Flow of specifying mis-understood scene

excluded from the presented scenes because the user must not specify the not-existing object. Therefore, when the user-intended scene is not found in the case of querying the existing object, the system suspects that an event for another object was mis-interpreted as the user-specified one in the past and then additionally shows the bring-in scenes of the already taken-away objects.

Scene Seeking Algorithm with Identifying Mis-understandings When the system detects the existence of the mis-understandings of automatic recognition, the category of mis-understandings cannot be immediately determined. The method specifying the category of mis-understandings is shown in Fig.7.

The system presents the candidate scene for the user's query. If all the presented scene is not the scene the user intended, the system asks the user whether the place the user specified is correctly recognized. If correct, then the system suspects a mis-understandings occurred in the past event recognitions and starts to specify the mis-understood event.

In specification of mis-understood event, the system first assumes the 'mis-interpreting' occurs in a certain moment. Then the system asks the user to give the time when the object had already existed or disappeared and seeks and presents the scene in 'Event-DB' around the given time (see Sec.4.2). Simultaneously, the system also starts the seeking in 'Human-DB' for the 'not-detected' category (see Sec.4.2) as a background job. In the foreground job, the system presents the found scene candidates (i.e. mis-interpreted ones) and ask the user to confirm it by speech dialog. If the user specified scene is successfully found, namely the mis-understood scene is specified and the system also can stop the background job. If not found, the system suspects the mistake is not-detected one, the scenes found in 'Human-DB' by the background job already started are shown to the user. If the user confirms it, the mis-understood scene is specified.

4.3 Propagative Correction of Mis-understanding of Scene Transitions

After specifying the mis-understood scene, the scene should be corrected. If that scene is the 'not-detected' one, add it to "Event-DB". If 'mis-interpreted' one, correct its event interpretation. Such a modification for one event necessarily affects the interpretations of the following events.

For examples, if a new 'bring-in' event is added, the event concerning 'taking-out' the new object may be mistakenly understood with high possibility. If a 'take-out' event is corrected as 'bring-in', the true 'take-out' event of the object may be mis-interpreted as another object's 'bring-in'. Therefore one correction of a mis-interpretation causes a number of chain reactions of interpretation corrections for the following events.

Based on the above analysis, our system tries to *propagate* one primary correction to the events following the corrected one as automatically as possible. When more events to

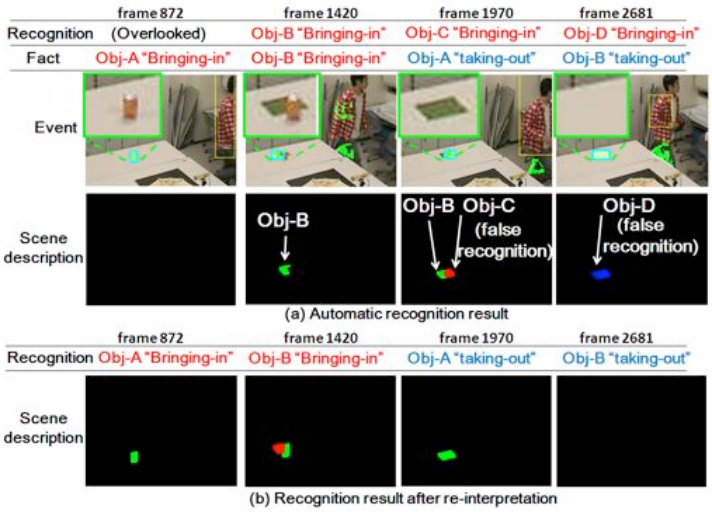

Fig. 8. Automatic Recognition of Scene Transitions and Corrected Ones

correct their interpretations are found, the correction for the events are recursively activated and each secondary corrections are propagated to further events again. If any ambiguous cases are found and needs user's help, the system shows those scenes and asks the user to give suggestions interactively.

This propagative correction of mis-interpretation can maintain the scene descriptions and event recognition almost correct by minimal correctional advises by the user.

4.4 Examples of Recursive of Corrections Scene Transitions

Here we show the examples of the recursive corrections of the scene interpretations triggered by one interactive corrections.

First Fig.8-(a) shows a automatic scene recognition result. At the 872nd frame, an new object A brought into the scene was overlooked by automatic recognition. The misunderstandings caused another mistake at the 1970th frame in which the 'take-out' event of object A was mis-interpreted as a 'bring-in' event of a new object C. This mis-interpretation causes a mistake of the scene change prediction at the following frames and further misinterpretation at the 2681st frame in which the 'take-out' event of object B was misinterpreted as a 'bring-in' event of a new object D.

When the user makes a query about the scene removing object A under the above situation, such a scene is not found in the database but the user can specify it by interactive seeking described in Sec.4.2. Then the specified 1970th frame's event is corrected as 'take-out' event and next the system specifies which object is taken out. In this case no object matched to the appearance change is not found in the predictions (see Sec.3.2) because its 'bring-in' scene is missed. Thus the system tries to interactively find it among the missed scenes registered in "Human-DB", and it is found at the 872 frame and added into "Event-DB". This additional event activates the recursive re-interpretations for the following events.

After the above processes, the transition of the scene descriptions and interpretations can be corrected as shown in Fig.8-(b). Since the scene bringing-in object A is newly added, the system obtains new information by comparing the background behind object A between before bring-in and after take-out of object A. The new information leads the conclusion that a part of object B, brought-in later than object A, is actually occluded. Based on this analysis, the layered scene description at the 1970th frame is modified, then

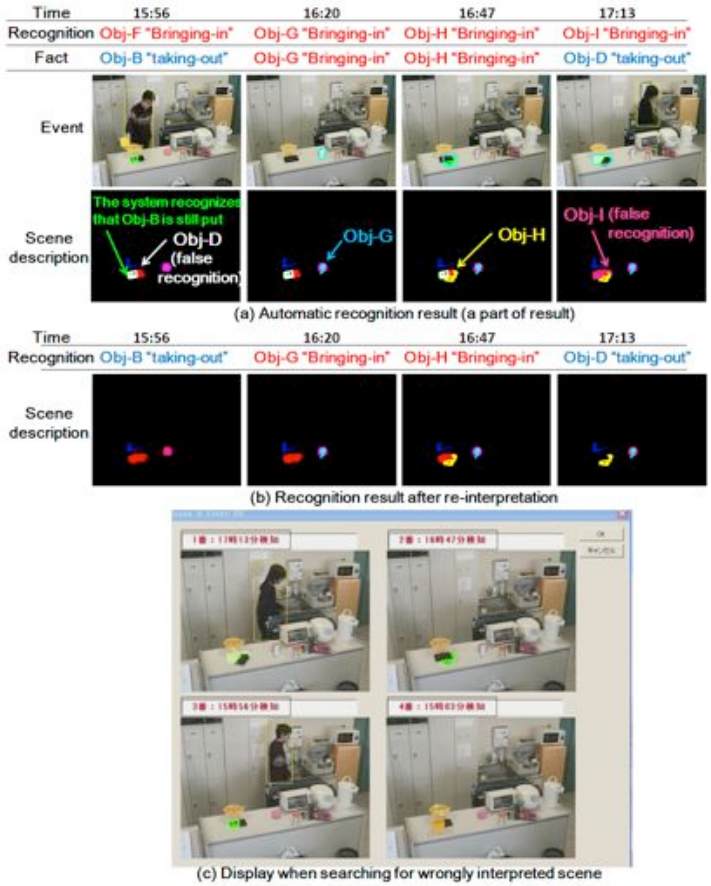

Fig. 9. Automatic recognition of Scene Transition

the scene change at the 2681st frame becomes correctly predicted, and then finally the 'take-out' event of object B at that frame is successfully recognized.

5 Experimental Results of Interactive Query

We shows the result of interactive query for the mis-understood scene.

Fig.9-(a) shows the fully automatic scene recognition results. Because a scene detected at 15:56 (actually 'take-out' scene of object B) was mis-interpreted, a following scene at 17:13 was also mis-interpreted. When the user made a query for removing of object B, only the scene of removing object A was shown as the scene candidate.

Then the system switched into interactive seeking mode and asked the user to give the time when that object had already been there. Since the user responded the object had already existed around 15:00, the system displayed the 'bring-in' scenes registered in "Event-DB" detected later than it (see Fig.9-(c), actually shown as movie).

Then the user found the desired scene at the lower left window and specified it via speech (or touch panel can be employed). The user satisfied the query result and left away there but the system rebuilt the interpretation of the whole scene transitions by itself. Since the system obtained the new interpretation that the scene at 15:56 was not a 'bring-in' but actually a 'take-out', it determined the taken-out object was object B. Then the layered scene description at 15:56 was rebuilt as shown in Fig.9-(b).

In the description, a part of object D (represented as red color) was found to be occluded by object B. This correct description of object D lead another correction of the scene at 17:13 to be the 'take-out' event of object D.

When another user made a query for the removing object D later, total three scenes including the correct scene were shown as the candidates and the user could select the scene easily.

6 Conclusion

This paper introduced the query system of missing or abandoned objects in in-door scenes. We proposed a method detecting mis-understandings occurred in the automatic image recognition and correcting based on the progress of interactive scene query. Triggered by correcting one mis-understanding, other concerned scene interpretations are recursively verified and corrected if needed. It is shown by experimental results that this architecture helps achieve not only efficient search but also efficient recovery of mis-understandings of automatic image recognition. The extensions to recognize more complicated events including moving, changing orientation, storing into bag, etc. are future works.

References

1. R. Collins, et al.: "A system for video surveillance and monitoring: VSAM final report", Technical report CMU-RI-TR-00-12, Robotics Institute, CMU, May 2000.
2. 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.
3. Tatsuyuki Kawamura, Takahiro Ueoka, Yasuyuki Kono, Masatsugu Kidode: "Evaluation of View Angle for a First-person Video to Support an Object-finding Task", 5th Int. Conf. of the Cognitive Science, 2006.
4. 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.
5. 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.
6. 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
7. Kazunori Komatani, Naoyuki Kanda, Tetsuya Ogata, Hiroshi G. Okuno: Contextual Constraints based on Dialogue Models in Database Search Task for Spoken Dialogue Systems, Proc. of 9th European Conf. on Speech Communication and Technology (Interspeech-2005), 877-880, Lisboa, Sep. 2005.
8. Kazuhiro Maki, Noriaki Katayama, Nobutaka Shimada, Yoshiaki Shirai: *"Image-Based Automatic Detection of Indoor Scene Events and Interactive Inquiry"*, 19th Int. Conf. on PATTERN RECOGNITION (ICPR2008),TuAT8.13,2008
9. 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.
10. Daniel Grest, Jan-Michael Frahm and Reinhard Koch: "A Color Similarity Measure for Robust Shadow Removal in Real-Time", VMV (Vision, Modeling and Visualization),2003
11. Gary.R.Bradski: "Computer vision face tracking for use in a perceptual userinterface", Intel Technology Journal, no.2nd Quarter, p.15,1998
12. "Open-Source Large Vocabulary CSR Engine Julius", http://julius.sourceforge.jp/