The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Household Object Management via Integration of Object Movement Detection from Multiple Cameras

Shigeyuki Odashima, Tomomasa Sato and Taketoshi Mori

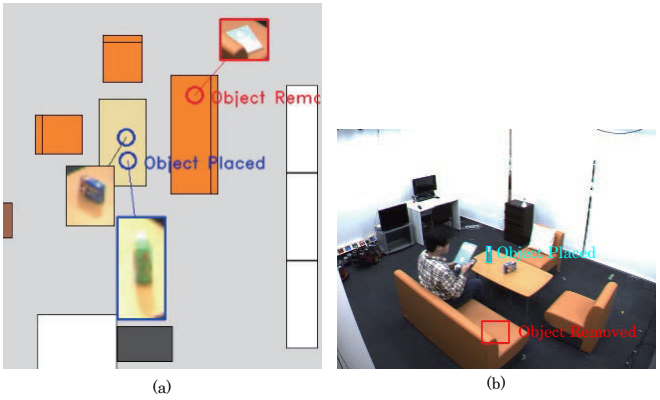
Abstract— **This paper proposes an object movement detection method covering large areas of a room by using multiple cameras. When object movement detection for whole of a room is performed, there are several challenging difficulties: sizes of objects on the camera images are small, non-objects such as humans also exist on the images, objects are sometimes difficult to detect in specific viewpoints because of occlusion by humans or furniture or color similarity to near objects. In this work, to detect object movements robustly though the object sizes are small, we apply multiple view integration via features extracted from "stable changes" on each viewpoint. To discriminate between object and non-object, we focus on motion of changed regions. Our experiment in a room environment shows the multiple view integration method improves recall rate of object detection performance by about 0.2 when false positive rate is over 0.1.**

I. INTRODUCTION

Object management in the intelligent household environments can give information of "where the object is now" or "when the object used", and it leads to the systems which give robots object information to bring [1], tell people where the lost object is, and support people by observing human-object interactions [2]. To realize object management, we first need to know "where and when the object moved" - especially, "object placement" and "object removal". In this paper, we propose an object movement detection and management system covering large area of a room via environment-embedded cameras. Fig. 1 shows an detection result of the proposed system. Fig. 1 (a) shows managed object movements in the room, and Fig. 1 (b) shows the image on a environment-embedded camera. The proposed system uses environment-embedded cameras in 4 viewpoints, and each viewpoint has two pair of cameras. In Fig. 1 (b), rectangles are overlaid on detected object regions.

There are several challenging problems to realize the object movement management system. 1) When the object movement management system covers whole area of a room, object sizes on the images are small due to limitation of camera resolution (as shown in overlaid rectangles of Fig. 1 (b)). You can use zoom cameras to get high-resolution images of objects, but the covering area of the system is limited or frequency of detecting object becomes very low. Therefore, the object detection method must work even if the object sizes on the images are small, so for this application, the approaches which works well only if the object sizes on images are big are difficult to use (e.g. recognizing objects on the images and detecting object movements from recognizing results). 2) There are humans in the images captured by the environment-embedded cameras. The detection system covers whole area of the room, so arranging cameras not to take the humans in the images is difficult. So, the detection system must discriminate between objects and non-objects such as a human. Nowadays, many human-detection approaches are appearance-based [3], [4], but especially in the household environment, robust appearance-based human detection is difficult due to occlusions by furniture (for instance in the Fig. 1 (b), the person's lower half of the body is occluded by a sofa). 3) Objects are sometimes difficult to detect in the specific viewpoints because objects are occluded by humans or furniture, or objects have similar color to near objects'. So robust object movement detection is difficult by using cameras only in a single viewpoint. When the detection system use several cameras in multiple viewpoints, the system need to integrate object movements detected in each viewpoint because a single object movement might be detected repeatedly in multiple viewpoints. But recognition-based integrating strategy does not work well because of low resolution of object as mentioned above.

In this paper, we propose an object movement management framework with the following strategies. 1) The proposed system detects "stable changes" caused by object movements. The stable change is the state that the region is changing from those which the system records, but the change is settled (e.g. when a book is placed on a table, the object



(a) (b)

Fig. 1. Object movement management results of the proposed system. (a): Detected object movements in the room. (b): An image captured by one of the environment embedded cameras. In (b), blue rectangles are overlaid on the detected object placement regions, and red rectangles are overlaid on the detected object removal regions.

Shigeyuki Odashima, Tomomasa Sato and Taketoshi Mori are with Graduate School of Information Science and Technology, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan. {odashima, tsato, tmori}@ics.t.u-tokyo.ac.jp

region is changing from “table”, but the changing region remains as “book”). The stable changes can be detected by background subtraction method even if the object size on the image is small. 2) To discriminate between objects and non-objects, the system employs the state machine of driven by motion of the changed regions. 3) To integrate a single object movement as a single “event” even if the object movement is detected repeatedly in multiple viewpoints, the system determines whether these detected object movements are caused by the same object movement or not via features extracted from the object movement itself.

This paper is organized as follows. The rest of this section discusses related works. Then, we provide an overview to our household object movement detection and management system in section II. The proposed system first detects object movements on a single image by detecting stable changes (section III). Then the system calculates where the object movement occurred in the room by using the stereo images in each viewpoint (section IV-A), and finally the object movements detected in each viewpoint are integrated (section IV-B) to manage object movements. In section V, we mention experimental results that the proposed system works well even if the object is difficult to detect in several viewpoints and the object size is small. Finally, conclusion is discussed in section VI.

Related Works. Many approaches have been proposed for object detection. Object recognition frameworks [1], [5], [6], [7] are useful for object detection, and these frameworks can be applied for object movement detection by recognizing objects on the images and detecting object placement and removal from recognizing results. These approaches are robust for shadows and illumination changes, and these approaches can work with moving cameras. But, as mentioned above, these approaches are not suitable for object movement detection system covering whole of a room. When object movements are considered as “highly-featured events”, approaches with attention point detection [8], [9] or anomaly detection [10] can be applied. These approaches will be able to work even if the object size on images is small, and these approaches can work even with moving cameras. However, if object movements occurs frequently, anomaly of object movements will be low. Moreover, the scene will be less featured if objects are removed, so approaches detecting highly-featured events will be difficult for robust detection of object placement and removal.

Object detection frameworks via stable changes are mainly based on background subtraction method [11], [12], [13] and our previous work with the state machine driven by motion of changed regions [14] is also based on background subtraction method. But these detection methods usually use only a single camera, so these methods cannot detect where the object moved in the room. In contrast, the proposed system can calculate the position of object movement, and moreover, the proposed system can detect object movements more robustly by integrating multiple viewpoints.

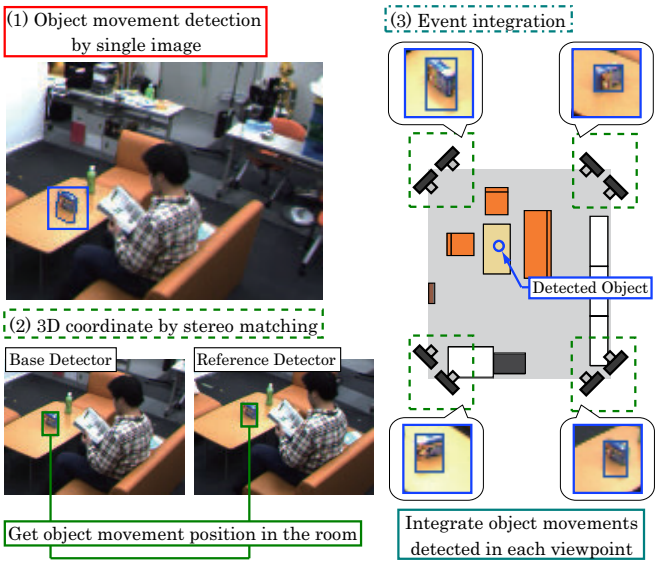


Fig. 2. Overview of the proposed method

II. OVERVIEW OF THE PROPOSED METHOD

The proposed system detects object movements by using multiple stereo cameras attached to the ceiling. Fig. 2 shows an overview of the proposed system. The proposed system first detects object movements in each viewpoint independently, and then integrate these repeatedly detected movements of a single object as a single “event”.

The proposed system has three major stages. First, our system detects object movement (object placement and object removal) from stable changes of the one image of the stereo camera in each viewpoint. Second, the other image of the stereo camera is gathered, and the position of the object movement in the room coordinate is calculated. Third, object movements of same object detected in multiple viewpoints repeatedly are integrated into a single event via features extracted from object movements.

As mentioned above, when object movement detection is performed in the whole of a room, the approaches based on object recognition do not work well because object sizes on image are small. To detect object movements robustly even if the object size is small, the proposed system first detects stable image changes caused by object movements, and then integrates object movements by the features extracted from themselves (in this method, HSV color histogram and object position is used as features).

III. OBJECT MOVEMENT DETECTION BY A SINGLE IMAGE

This section describes the method to detect object movements via a single image in each viewpoint. Fig. 3 depicts an overview of the object movement detection process from a single image. In this stage, (1) the system first extracts changed pixels by background subtraction method and categorizes them into “something-inserted” state and “something-removed” state, and (2) employs blob detection algorithm to the changed pixels and extracts regions.

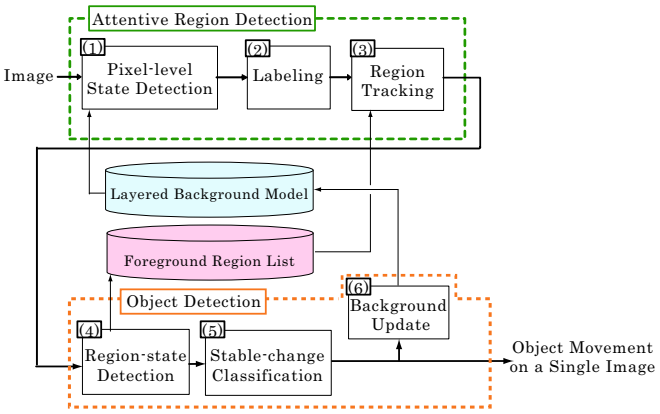


Fig. 3. Overview of object detection on a single image

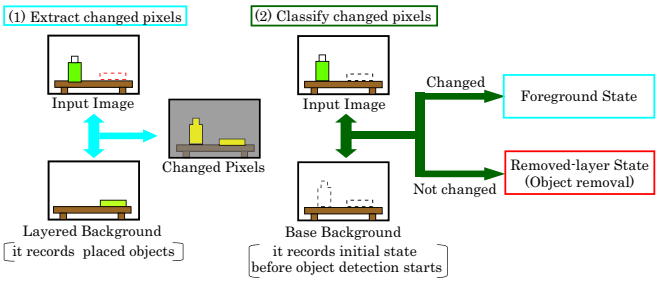


Fig. 4. Layered background model

Then, (3) the system tracks the extracted regions, and (4) discriminates between non-object state regions and object state regions via their motion detection result for past several frames, called motion history. (5) The system then classifies the object-state regions into object placement and object removal via edge subtraction on their boundaries. Finally, (6) the system updates its background models according to the object detection result.

In the rest of this section, two main aspects of this stage is described: object movement detection from background subtraction and edge subtraction (section III-A), and classification of object and non-object via motion history (section III-B).

A. Object Movement Detection via Layered Background Model and Edge Subtraction

To detect object movements, the proposed system first extracts changed pixels by background subtraction method (step (1) in Fig. 3). We apply Shimosaka's background subtraction method [15] with energy minimization via min-cut / max-flow algorithm [16], which is robust for background clutter and shadows.

Object placement and object removal generates changed pixels equally, so we need to classify the changed pixels into the pixels generated by object placement, called foreground state, and the pixels generated by object removal, called removed-layer state. To classify the changed pixels into two states, we adopt the multiple-layered background model, called layered background model [11], [12], [14]. Fig. 4 depicts an overview of our layered background model. The layered background model consists of two background models: the base background and the layered background. The base background records the static background (e.g. furniture), and the layered background overlays placed objects on the base background. The system generates the base background when object movement detection starts. The system inserts detected objects into the layered background when object placement is detected, and remove detected object from the layered background when object removal is detected.

Extraction of the foreground state pixels and the removed-layer state pixels is performed as follows. First, the input image is compared with the layered background, and changed pixels are extracted. Next, the changed pixels in the input image are compared with the base background. If the pixel is changing from the layered background but not changing from the base background, the pixel is changing after object placement but is not changing before object placement, so it represents "something removed" (removed-layer state). On the other hand, the pixel is changing from both of the layered background and the base background, the pixel is classified as the foreground state.

Classification of object placement and object removal via the layered background model can detect easily which object in the detected objects is removed. But, if the objects which existed before object movement detection performing are removed, the regions of the removed objects change from both of the layered background and the base background, the pixels of the removed object are classified as foreground state. So, only with layered background model, object removal in the initial state cannot be handled.

To classify placement of objects and removal of objects which existed in the initial state, we apply a classification method based on edge subtraction [17] (step (5) in Fig. 3). Generally, the region where objects does not exist is less textured than where object exist. So, textures of the region will increase when an object is placed, and will decrease when an object is removed. In this research, the amount of edge energy in boundary of the foreground region of the input image and the layered background are extracted, and if the edge energy of input image is greater than the one of the layered background, the region is classified as object placement. Otherwise, the region is classified as object removal.

B. Object and Non-Object Classification via Motion History

The proposed system classifies objects and non-objects via motion detection result of foreground regions for past several frames, called motion history [14]. This motion based approach can classify non-objects robustly if the human body is occluded by furniture.

The motion-based classification is operated as follows. First, the extracted foreground regions are tracked by a keypoint-based tracking method (step (3) in Fig. 3). In the keypoint-based tracking method, keypoint patches in the foreground regions are extracted by FAST-10 operator [18], and the foreground regions detected in current frame are

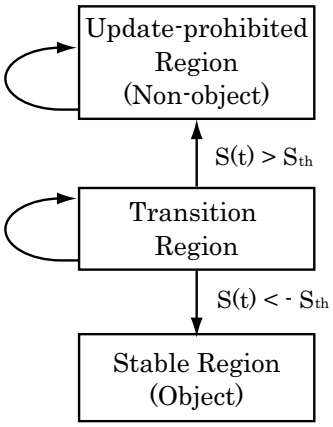


Fig. 5. State machine of object and non-object classification via motion history

matched with foreground regions detected in previous frames by using these keypoint patches. Second, motion in the tracked region is detected by frame subtraction technique, and the region is classified into object state and non-object state by a state machine based on motion history (step (4) in Fig. 3). Fig. 5 depicts the state diagram of the state machine. Each region has stability value $\mathbf{S}(t)$, and if the region is detected as moving by frame subtraction, the stability value is incremented ($\mathbf{S}(t) = \mathbf{S}(t-1) + k$), and if the region is detected as not moving, the stability value is decremented ($\mathbf{S}(t) = \mathbf{S}(t-1) - k$). To avoid unstable detection, we set the incremental parameter $k = 2$ if the region's motion detection result is equal to the previous result, and $k = 1$ if not. The state machine of each region transits its state according to the stability value $\mathbf{S}(t)$. Initially, each region transits to the transition state. If $\mathbf{S}(t)$ of the region is over the fixed threshold $\mathbf{S}_{th}$ (if the region moves for a long time), the region transits to the update-prohibited region (non-object state). Otherwise, if $\mathbf{S}(t)$ of the region is under the fixed threshold $-\mathbf{S}_{th}$ (if the region does not move for a long time), the region transits to the stable region (object state). But especially in the household environment, non-objects do not move for a long time in several cases (e.g. a person is sitting down and reading books). In these cases, non-objects are expected to move for a long time before being stable, so the update-prohibit regions transit only to update-prohibit regions to avoid misclassifying non-objects as objects. In our implementation, the threshold parameter $\mathbf{S}_{th}$ is set to 20.

IV. OBJECT MOVEMENT DETECTION IN THE ROOM

After detecting object movement on a single image, the system calculates where the object moved via stereo images. And then, the system integrates object movements of the same object detected repeatedly in multiple viewpoints.

A. 3D Coordinate Calculation by Stereo Matching

The object movement detected on a single image is redundant in the direction of depth, so the position where the object moved in the room cannot decide from only a single image. The proposed system calculates the position where object moved (3D coordinate) by stereo matching. When stereo matching performed, the system gives different roles for each camera. The system uses one camera as the object detector on a single image as mentioned in section III (base detector), and the other camera as a reference view for stereo matching (reference detector). In the reference detector, the system does not perform the object detection. Compared with an approach detecting objects in each camera, this approach is not affected by difference of object detection timing between each camera, and can reduce computational costs. Also, this approach has advantages compared with an approach using two single camera nodes as a wide baseline stereo camera. When the baseline width of two cameras are short, the direction of two cameras are almost same, so the difference of the object images in two cameras is regarded to be only in translation. At the same time, the object is hardly occluded only in one camera image of the stereo cameras. So with this approach, you can easily match the object regions of the two cameras.

The system calculates the position of object movement by matching the object region detected on the base detector to the reference detector images. If the system operates stereo matching by using only input images, robust position detection is difficult when object removal occurs because the region where object removal occurs is less textured, so stereo matching does not work well. The proposed system has background images in the reference detectors, and operates stereo matching via both of the input image and the background image, to detect position robustly when object removal occurred.

When stereo matching is operated, the system detects a region on the reference detector where SSD score at the object region on the base detector is minimized. The SSD score $SSD(R)$ for the object region R is calculated as follows:

$$\begin{aligned} &SSD(R) \\ &= \sum_{(u,v)\in R} \{(I_{im}(x_m+u, y_m+v) - I_{is}(x_r+u, y_r+v))^2 \\ &\quad +(I_{bm}(x_m+u, y_m+v) - I_{bs}(x_r+u, y_r+v))^2\} \end{aligned} \quad (1)$$

where I_{im} and I_{is} are the input images of the base detector and the reference detector, I_{bm} and I_{bs} are the background images of the base detector and the reference detector, respectively. The matching procedure is operated on an epipolar line in the stereo camera.

B. Event Integration

After calculating the position of object movements in the viewpoint, then the system determines whether the object movement has already been detected in the other viewpoints by event matching. Objects are expected to be same color and position regardless of viewpoints. The proposed system

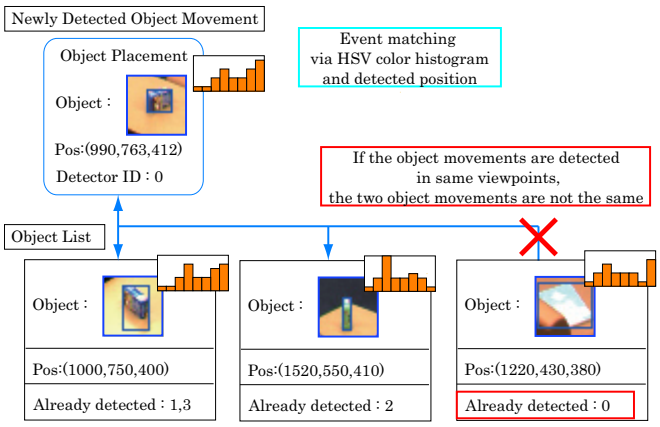


Fig. 6. Event matching of object movements

matches the newly detected object movement to object movement events detected in the previous frames by evaluating similarities between them via their color histogram and calculated position.

Fig. 6 depicts an overview of event matching. The object movement events detected in previous frames are stored into the object list of the system, and newly detected objected movements are matched to the events in the object list. The system stores color histogram, detected position and detected viewpoints of the events in the object list. The similarity $S_e(E_n, E_p)$ between the newly detected object movement E_n and an already detected object movement event E_p is calculated as follows:

$$S_e(E_n, E_p) = \omega e^{-\frac{d_p}{\lambda_p}} + (1-\omega)e^{-\frac{d_h}{\lambda_h}} \quad (2)$$

where ω, λ_p and λ_h are constant. d_p is the position likelihood term, which is Euclidean distance between the position of E_n and E_p. d_h is the color likelihood term, which is difference between color histogram between the color histogram of E_n and the one of E_p. We apply Pérez's HSV color histogram [19]. This HSV color histogram is composed with HS histogram (hue and saturation direction: $N_H N_S$ bins) and V histogram (brightness direction: N_V bins), and this color histogram is composed of total $N = N_H N_S + N_V$ bins. In this research, N_H, N_S, N_V are set to 10 ($N = 110$). Also, we set $\omega = 0.5$, $\lambda_p = 1000$[mm], $\lambda_h = 0.7$.

We apply Bhattacharrya coefficient as distance metric of color histograms. The distance between a histogram $q(E_n)$ of the newly detected object movement E_n and a histogram $q(E_p)$ of the object movement event in the object list E_p is calculated as follows.

$$d_h(q(E_n), q(E_p)) = \sqrt{1 - \sum_{k=1}^{N} \sqrt{q(E_n;k)q(E_p;k)}} \quad (3)$$

The newly detected object movement is matched to each stored object movement event and similarity score S_e to each event is calculated. If the maximum similarity among them is below the fixed threshold S_{eth}, the system determines the newly detected object movement has not been detected in another viewpoints, and stores the object movement into the object list. Otherwise, the system determines that the newly detected object movement has been detected in another viewpoints, and integrates the newly detected object movement to the event with maximum similarity in the object list. In the integration operation, the system adds the viewpoint of newly detected object movement to the matched event.

The system detects an object movement repeatedly in the different viewpoints, but only once in a single viewpoint because the system updates its model according to detection results. So, in the event matching operation, if the newly detected object and the compared event in the object list are detected in the same viewpoint, the system determines the two objects different regardless of their similarity. By using this event matching restriction, the system can detect two objects separately if the two objects have same color and are placed closely but their placed timing are apart, because the system detects and integrates the first object placement before the second object placement occurs, so the system determines the second object movement different from the first object movement according to this matching restriction.

V. EXPERIMENTAL RESULTS

We evaluate the performance of the proposed system on video sequences captured by the stereo cameras in 4 viewpoints (2 cameras in each viewpoint, total 8 cameras). The baseline length of the stereo cameras was roughly 200 [mm]. We calibrated each camera by Zhang's chessboard calibration method [20]. All sequences consist of images of 320 × 240 resolution recorded at 7.5 fps.

In the experimental results mentioned below, to reject the changes caused without object movements (e.g. shadows, small object shift), we set fixed threshold parameters in the process of object movement detection by a single image (mentioned in section III) - sizes of extracted regions (R_{th}), HSV histogram difference of input image and background image in the extracted regions (C_{th}), ratio of major axis to minor axis of the region by approximating the region to ellipse (E_{rth}) and length of minor axis (L_{mth}), and the average width of the region (W_{th}). Also, in the event matching procedure (mentioned in section IV-B), if the distance between the two detected object movements are over the fixed threshold (D_{th}), the two object movements are regarded as different object movements.

In the following, we first show accuracy of the calculated position of detected object movements and then discuss object movement detection performance of the system.

A. Accuracy of the calculated position

We evaluate the calculated positions of the detected object movements by the proposed system on 2 video sequences (total 1609 frames in each camera). We compute errors of correctly detected object movements between the calculated position and the true position. We extract the true position from the motion data collected by a motion capture system

TABLE I

ACCURACY OF THE POSITION OF DETECTED OBJECT MOVEMENTS

average error [mm]	Event integration	Single viewpoint
2D position	7 × 10	7 × 10
3D position	9 × 10	9 × 10

NaturalPoint Optitrack. The video sequences contain 17 object placements and 17 object removals (total 34 object movements) of a object, and all object movements occurs in the center area of the room. The object used in this experiment is green plastic bottle, which size is roughly 100[mm] × 200[mm] × 80[mm]. We perform the experiment when the system integrates multiple viewpoints and when the system uses a single viewpoint. In the evaluation of the case of using a single viewpoint, object movements detected repeatedly in multiple viewpoints are treated different object movements.

In this experiment, we set threshold parameters $R_{th} = 20$[pixels], $C_{th} = 0.5$, $E_{rth} = 10$, $L_{mth} = 2$[pixel], $W_{th} = 2$[pixel], $D_{th} = 1000$[mm] and event similarity threshold $S_{eth} = 0.5$.

Table I shows accuracy of the calculated position. In Table I, 2D position is the position error excluding height, and 3D position is the position error including height. In this evaluation, correct 17 object placements and correct 17 object removals are detected when the system integrates multiple viewpoints, and correct 52 object placements and correct 52 object removals are detected when the system uses only a single viewpoint. Regardless of event integration, the calculated position error is in the range of 100 [mm], so the system calculates position of object movements sufficiently for searching objects.

B. Object movement detection performance

We evaluate the object detection performance of the proposed system on 6 video sequences (total 4294 frames in each camera). The sequences contain 46 object placements and 42 object removals (total 88 object movements) in the experiment area (roughly 4.0 meters wide and 4.7 meters long). In the video sequences, a person places and removes objects in the room, so the system need to detect object movements without detecting the person as an object.

In the experiment, we employ false positive and recall as the performance evaluation measures, as defined below.

$$\text{false positive} = 1 - \frac{\text{correctly detected object movements}}{\text{total detected object movements}} \tag{4}$$

$$\text{recall} = \frac{\text{correctly detected object movements}}{\text{total object movements}} \tag{5}$$

In this experiment, we calculated the performance of the proposed system under variant threshold parameters R_{th}, C_{th}, E_{rth}, L_{mth}, W_{th}, D_{th} and S_{eth}. Fig. 7 shows the resulting detection performance with various parameters. The blue line in Fig. 7 is the detection result when the system integrates detects object movements in multiple viewpoints, and the green dotted line is the average detection result when the system uses only a single viewpoint. As can be seen from the graph, where false positive is over 0.1, recall is improved by event integration of multiple viewpoints (about 0.2 improvement). This is because the system can detect object movements more robustly for background color or occlusions by using multiple viewpoints. Fig. 8 shows an example that the proposed system detected placement of a white telephone (below, the shown results are taken with threshold parameters in section V-A, when false positive = 0.21 and recall = 0.92 in this experiment). In Fig. 8, the left image is the result of object placement detected by the system in a viewpoint, and the right images are input images on the same frame in the other viewpoints. In the detection result, the left image represents the calculated position of object movement, and the upper-right image and lower-right image represent background image and input image when the object is detected, respectively. In the background images and the input images, blue rectangles are overlaid on the object regions (blue solid line if the system detects object placement in the viewpoint, and blue dotted line if fails to detect in the viewpoint). In this result, the system detects object placement in two viewpoints (detector 2 and 3) and integrates the two object movements but the system failed to detect in two viewpoints (detector 0 and 1). In this case, the white telephone is placed near white curtains, so the telephone is hard to see in several viewpoints. But by integrating multiple viewpoints, the system detects the event of object placement via the viewpoint easy to see the object. At the same time, the detected object size on the image is small (15 × 13 pixels), but the proposed system detects object placement.

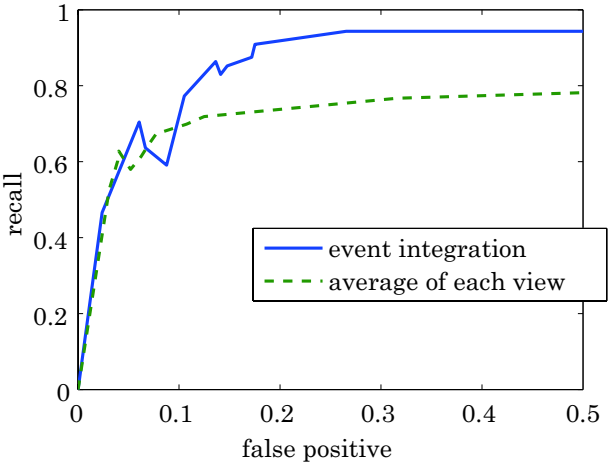


Fig. 7. ROC curves of the proposed system

Fig. 9 shows another example of detected object movement. In Fig. 9, the left image is the detection result of placement of a object (plastic bottle), and the right image is the detection result of removal of the object. In these results, the system detects object placement and removal in

different viewpoint because the system uses detection results of the viewpoint which detects the object movement most quickly by integrating multiple viewpoints. At the same time, a person is sitting down in this case, but the system does not detect the person as object but detects object movement. So, the motion-based classification method of object and non-object works robustly.

C. Discussion

The proposed system has several limitations. First, the background subtraction method is constructed assuming the lighting condition is constant, so the system cannot handle strong illumination changes (e.g. switching off the lights). Second, the proposed system cannot handle movement of furniture (e.g. opening and closing of a door, rotation of a chair) because they are intermediate state of "object placement" and "object removal". Third, the event integration method does not work well in low false positive and low recall area (under 0.1 false positives in Fig. 7). This is because the system detects object movements only in specific viewpoints because of low recall parameters, so determination whether the object movement is newly detected or already detected by event similarity threshold S_{eth} does not work well. With these parameters, if S_{eth} is set high, the system determines object movements of the same object as different object movements, and if S_{eth} is set low, the system determines object movements of different objects as a same object movement. So the system detects events wrongly regardless of S_{eth}. But, for the aim of "logging of object movements", the priority of high precision of object movement detection is low, so this limitation of event integration does not matter for practical use.

The proposed method has several threshold parameters, but the patterns of false positives are few and are dependent on the environment (e.g. small shift of the sofa, shadow on the display), so the method can determine the parameters automatically by using the false positive data detected on the object movement detection system. In this work, the proposed method is implemented on a single laptop PC and works only in offline. But, the average calculation time in a single viewpoint is roughly 130[ms/frame]on single-thread by Intel Core 2 Duo 2.5 GHz processor, so the proposed method will work in real-time with distributed computation.

VI. CONCLUSION

This paper has proposed a novel method for detecting object movements in the household environment. To detect object movements robustly even though the object size on images is small, the system detects object movements via stable image changes in each viewpoint, and then integrates these detected object movements via the features extracted from themselves. Also, to classify objects and non-objects robustly though they are occluded, the system uses motion history of extracted changes.

Experimental results show the proposed system detects object movements robustly in the household environments, and the proposed event integration method can improve object detection performance when the objects has similar color to near objects'. Future tasks are handling large object shift such as movement of furniture, and developing the system for capable of long-term logging.

APPENDIX

We would like to thank the NEDO project "Development of Intelligence Technology for the Next Generation Robots" for its financial support.

REFERENCES

[1] Kimitoshi Yamazaki and Masayuki Inaba. A cloth detection method based on wrinkle features for daily assistive robots. In *MVA*, 2009.
[2] A. Gupta, A. Kembhavi, and L.S. Davis. Observing human-object interactions: using spatial and functional compatibility for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(10):1775–1789, 2009.
[3] Dalal Navneet and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.
[4] P. Sabzmeydani and G. Mori. Detecting pedestrians by learning shapelet features. In *CVPR*, 2007.
[5] Josef Sivic and Andrew Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, 2003.
[6] D. Liu and Tsuhan Chen. DISCOV: A framework for discovering objects in video. *IEEE Transactions on Multimedia*, 10(2):200–208, 2008.
[7] Hideki Nakayama, Tatsuya Harada, and Yasuo Kuniyoshi. AI Goggles: real-time description and retrieval in the real world wih online learning. In *CRV*, 2009.
[8] L. Itti and P. Baldi. A principled approach to detecting surprising events in video. In *CVPR*, 2005.
[9] S. Gould, J. Arfvidsson, A. Kaehler, B. Sapp, M. Messner, G.R. Bradski, P. Baumstarck, S. Chung, and A.Y. Ng. Peripheral-foveal vision for real-time object recognition and tracking in video. In *IJCAI*, 2007.
[10] R. Matsumoto, H. Nakayama, T. Harada, and Y. Kuniyoshi. Journalist robot: robot system making news articles from real world. In *IROS*, 2007.
[11] K. Kim, T.H. Chalidabhongse, D. Harwood, and L. Davis. Real-time foreground–background segmentation using codebook model. *Real-Time Imaging*, 11(3):172–185, 2005.
[12] Kazuhiro Maki, Noriaki Katayama, Nobutaka Shimada, and Yoshiaki Shirai. Image-based automatic detection of indoor scene events and interactive inquiry. In *ICPR*, 2008.
[13] Y. Tian, M. Lu, and A. Hampapur. Robust and efficient foreground analysis for real-time video surveillance. In *CVPR*, 2005.
[14] S. Odashima, T. Mori, M. Shimosaka, H. Noguchi, and T. Sato. Object movement event detection for household environments via layered-background model and keypoint-based tracking. In *International workshop on video event categorization, tagging and retrieval*, 2009.
[15] Masamichi Shimosaka, Kazuhiko Murasaki, Taketoshi Mori, and Tomomasa Sato. Human shape reconstruction via graph cuts for voxel-based markerless motion capture in intelligent environment. In *IUCS*, 2009.
[16] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, 2004.
[17] J. Connell, A.W. Senior, A. Hampapur, Y.-L. Tian, L. Brown, and S. Pankanti. Detection and tracking in the IBM peoplevision system. In *ICME*, 2004.
[18] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *ECCV*, 2006.
[19] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In *ECCV*, 2002.
[20] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.

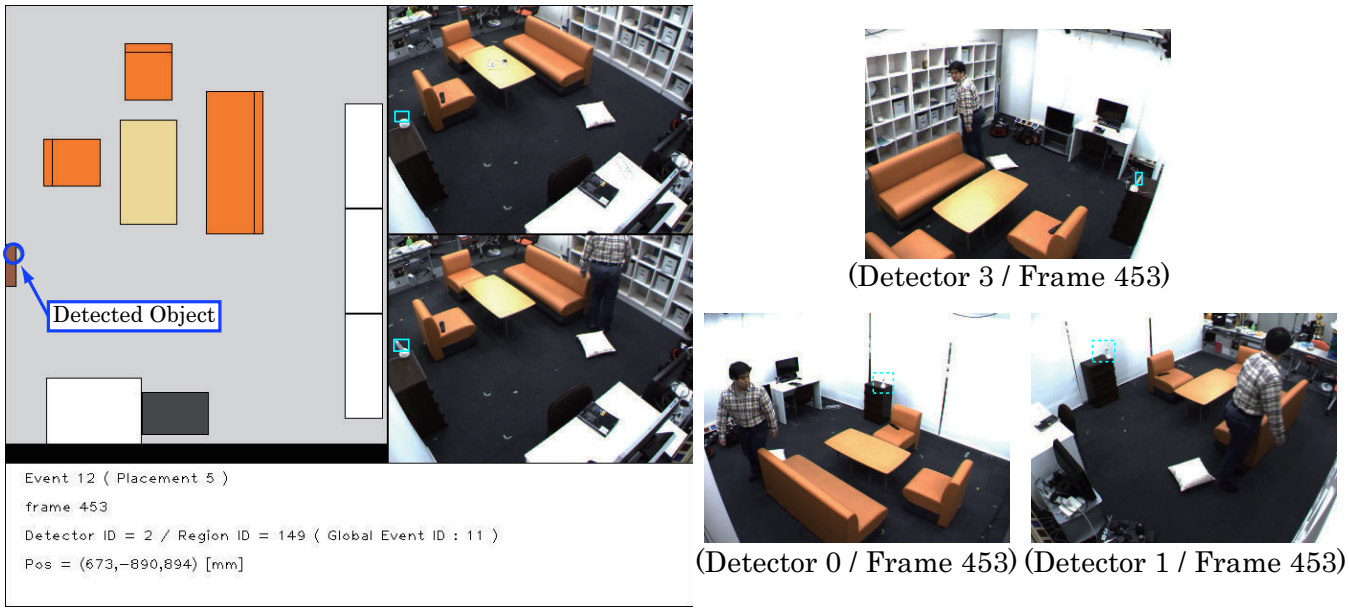


Fig. 8. An object placement detection result: when the object color is same to near object's

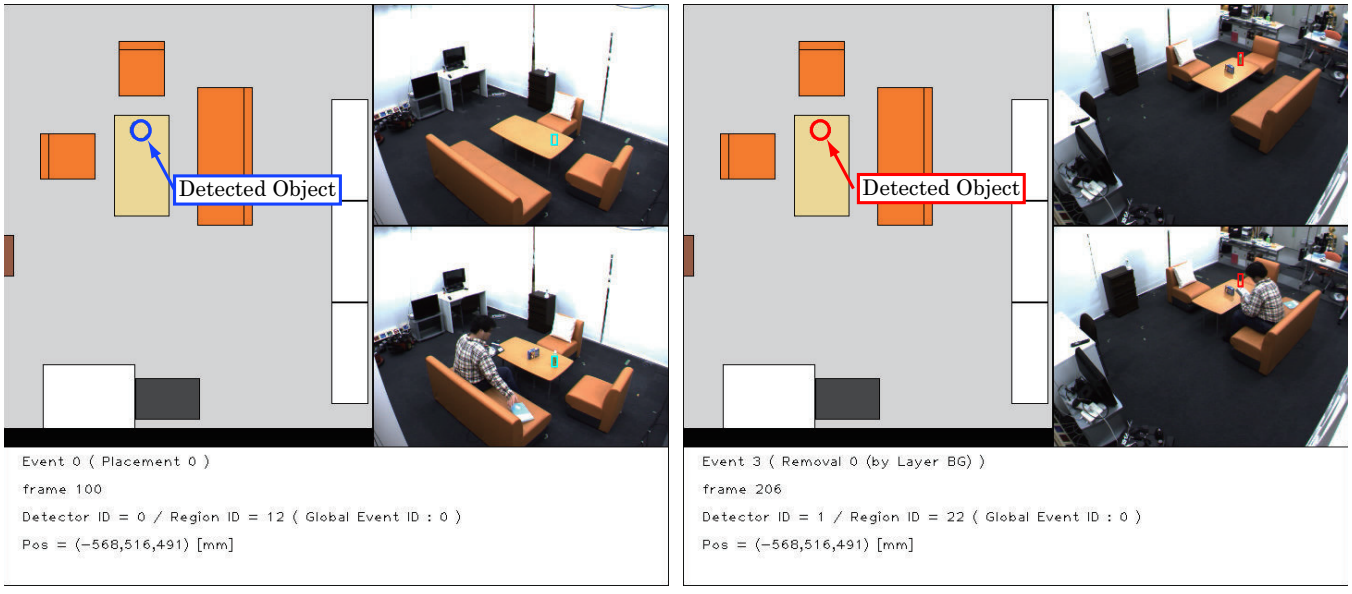


Fig. 9. An example of detection of placement and removal of one object