PAPER

Utilizing Attributed Graph Representation in Object Detection and Tracking for Indoor Range Sensor Surveillance Cameras

Houari SABIRIN[†a)], **Hiroshi SANKOH**[†], ***Members***, ***and*** **Sei NAITO**[†], ***Senior Member***

SUMMARY The problem of identifying moving objects in a video recording produced by a range sensor camera is due to the limited information available for classifying different objects. On the other hand, the infrared signal from a range sensor camera is more robust for extreme luminance intensity when the monitored area has light conditions that are too bright or too dark. This paper proposes a method of detection and tracking moving objects in image sequences captured by stationary range sensor cameras. Here, the depth information is utilized to correctly identify each of detected objects. Firstly, camera calibration and background subtraction are performed to separate the background from the moving objects. Next, a 2D projection mapping is performed to obtain the location and contour of the objects in the 2D plane. Based on this information, graph matching is performed based on features extracted from the 2D data, namely object position, size and the behavior of the objects. By observing the changes in the number of objects and the objects' position relative to each other, similarity matching is performed to track the objects in the temporal domain. Experimental results show that by using similarity matching, object identification can be correctly achieved even during occlusion.

key words: *3D sensor, infrared camera, surveillance, automatic object tracking*

1. Introduction

Surveillance monitoring systems are intended to gather information on the behavior of moving objects in the monitored area. For indoor surveillance of a shop, a video recording from a surveillance camera can be used by the shop's manager to identify the movements of customers, shop assistants, their interactions with each other, and to maintain the security of the shop. On the other hand, customers might feel uncomfortable with the presence of surveillance cameras, due to the possibility of privacy infringement [1].

To achieve detection and tracking of human objects without requiring detailed information (i.e., texture or pixels) of their faces, a range sensor camera can be used. A range sensor camera may utilize an array of sensors that measure the time a light signal takes between the monitored area and the camera. As a result, the data acquired from such a camera would be the value of the distance between the captured scene (the moving objects and the background) and the camera. This information, the depth data, is sufficient to indicate the presence of moving objects.

In this paper, an automatic object detection and tracking method is proposed that uses the depth data acquired from range sensor cameras. Depth data obviously provides less information from the captured scene compared to that of a conventional color camera. Therefore, the main challenge of object detection and tracking based on depth data is to use the limited information to match the tracking performance achieved with color information. Color information is commonly regarded as an important feature in object detection and tracking, as shown by Joshi and Thakore [2]. Particularly when handling object occlusions, color information can provide sufficient information to correctly identify the objects. While the proposed method may not be able to achieve 100% accuracy compared to color-based object detection, and tracking cannot be easily achieved, in the proposed method the ultimate goal is to detect all moving human objects and label them so that the detected objects can be distinguished, even during occlusion.

The method proposed in this paper aims for indoor object tracking for a medium to large coverage area. It consists of data preparation and construction followed by automatic object tracking. The depth information acquired from a range sensor camera is measured in millimeters per pixel. Each pixel can be presented as a gray-scale value after normalization to provide a glance of how the scene visually appears. This preparation step is followed by projecting the depth information onto the 2D plane for further processing. The structure of the detected objects is then constructed by extracting the features from the captured frame: width and height of the projected data and its centroid. Finally, automatic object tracking is performed upon the 2D-projected data, using the extracted features to identify different objects. The proposed method utilizes attributed graphs that are constructed in the spatio-temporal domain to represent the moving objects. Object tracking is then performed by observing the number of objects detected in the 2D plane as well as the interactions between objects, such as a merge or separation. Selection between similarity matching between two consecutive frames, or between the current frame and a selected reference, is performed based on the observation. The tracking result is then shown as a 2D representation of the objects, showing uniquely assigned identities and their trajectory along the frames in the sequence.

This paper is structured as follows: related work is first reviewed in Sect. 2; Section 3 provides the detail of the proposed method; the experimental results are presented in Sect. 4 and Sect. 5 concludes this paper.

Manuscript received March 26, 2015.
Manuscript revised August 7, 2015.
Manuscript publicized September 10, 2015.

[†]The authors are with KDDI R&D Laboratories, Inc., Fujimino-shi, 356–8502 Japan.
a) E-mail: ho-sabirin@kddilabs.jp

DOI: 10.1587/transinf.2015EDP7108

2. Related Work

Object detection and tracking using range sensor cameras has existed for many years, especially targeting the area where sufficient light conditions cannot be obtained. Range sensor cameras are also commonly used in robotic-based object detection to estimate the distance between the robot and its environment [3], [4], which is not closely related to the work presented in this paper.

It is not uncommon that depth information is utilized as a supporting feature in color-based object detection and tracking to produce better results in accuracy and to handle complex occlusion which cannot be easily achieved by using only color information. Depth information is used by Tran and Harada [5] to handle occlusion by projecting an occlusion region detected in a color frame into a depth estimation frame. Liu et al. [6] use a combination of RGB information and depth information to find the upper part of human body, which is assumed to be less prone to occlusion. In the proposed method by Jafari et al. [7] the depth information is utilized via an RGB-D camera mounted on the person's head. Fu et al. [8] and Bondi et al. [9] use depth information in a people counting system to estimate the position of head and shoulders of the detected objects to accurately locate the human objects to be counted.

Wang et al. [10] proposed a method based on a Support Vector Machine (SVM) classifier to detect a moving object from an infrared image. By training the component classifier via SVM, the moving object can be segmented from the background. The depth stream from a Kinect sensor is utilized by Tian et al. [13] to detect a moving human body based on the head and shoulders. Here, SVM is also used to isolate the head and shoulders of the human body for real-time object tracking. A range sensor mounted on a moving vehicle is used by Liu et al. [11] to track multiple objects in a traffic scene. The Kalman Filter is used in this method to track the detected object. Li and Gong [12] conducted pedestrian detection and tracking over thermal infrared imagery. The histogram of intensity of the detected object from the infrared image is then fed into a particle filter for robust tracking. Similarly, Ikemura and Fujiyoshi [15] use a histogram from depth information to detect and track multiple human objects, especially to find the relationship between two local regions, to accurately detect different objects. A Kinect is employed by Xia et al. [14] to produce a depth stream to detect and track moving humans. Here, the 2D contour model and 3D surface model of a human head is used as the reference to determine the region of the object so it can be segmented from the background and from other objects. Hansen et al. [16] utilized depth information by constructing clusters of moving objects. To track these moving clusters, an EM algorithm is used to estimate the parameters of those clusters. Multiple range sensors are used by Jia and Radke [17] to produce depth information of objects in a room. Object tracking is performed upon the depth information of the moving objects by observing the weighted centroid of each detected object. The people counting method proposed by Hsieh et al. [18] provides a good guide to the method proposed in this paper to handle merged objects in depth data by first projecting the depth data onto the 2D plane.

Projection of depth data onto the 2D plane has also been used in depth-based object tracking in Zhou et al. [19]. Here, a Kinect mounted in a mobile robot is used to obtain depth data. The method utilizes the depth in the form of binary and depth data, called Ground Plane Projections (GPPs). The method provides reliable object tracking results in relatively small areas of indoor observations, which may not be sufficient for large room object tracking as proposed in this paper.

While the results of object detection and tracking based on depth information have been shown to be robust, the methods lack unique labeling for the detected objects. The information to determine the correct identity of the detected objects seems to be neglected, thus cases such as object overlapping are also not presented. The goal of the work presented in this paper is closely related to the work by Latecki et al. [20], where the detected objects from an IR camera with a large coverage area are uniquely identified using identification numbers. Their method tracks the objects by finding the similarity of the positions of the objects (represented by a bounding box) and their velocity from frame to frame. In cases where the sizes of the detected objects are small compared to the image size and there are no occlusions among them, the accuracy of object tracking will be relatively high. However, when many occlusions occur, for example due to an indoor environment where the camera position is relatively closer to the objects, the method by Latecki et al. would not be sufficient. Therefore in the method proposed in this paper, more information to identify moving objects is utilized to handle complex movements and occlusion problems.

3. Proposed Method

3.1 Data Preparation

The main issue in detecting and tracking an object from depth data is the limitation of information that is available to correctly and consistently identify the moving objects. To achieve an accuracy similar to that of color-based object tracking, a complex process involving manual operation may be necessary. Therefore, some restrictions in depth-based object tracking are usually applied as previously mentioned in Sect. 2.

Maintaining a less complex object detection and tracking process requires that the indoor range sensor in a room is statically mounted at a high position close to the ceiling and the positions of the furniture in the room are rarely altered. To cover a large room that exceeds the coverage range of the sensor, additional sensors would be mounted at the opposite end of the room. Therefore, it is possible to obtain an initial homographic projection of the scene captured by the sensor

Fig. 1 (a) depth data with highlighted moving objects; (b) segmented moving objects; (c) contours of objects in (b) as 2D projection with bounding rectangle indicating the region of interest (ROI).

to determine the transformation matrix for 2D projection as well as the initial background frame (i.e., the captured frame in which only the empty room and the furniture are present). In addition, background segmentation can be performed by simple pixel subtraction between the current frame and the initial background frame.

The values of the depth data acquired from a range sensor camera are in the form of the distance of the captured objects (background or moving objects) from the camera in centimeters per pixel. To visualize the depth scene, the depth data is normalized into gray-scale values (0-255) so that the images of actual scene can be reconstructed. From these images, the position of the detected moving objects and their environment can be observed.

The depth data is then projected onto a 2D plane from the normalized depth data by calculating the transformation matrix provided by the camera specification. In the proposed method, the 2D plane is a processing plane onto which the depth data acquired from more than one range camera is projected, as illustrated by Fig. 4. The main advantage of the projection is to reduce the complexity of processing object tracking with more than one sensor, as in the experimental setup that will be described in Sect. 4. In some cases, the projection may also reduce the number of overlapping pixels between objects (thus occlusions may also be reduced). The resulting projected 2D data is a binary image where nonzero pixels represent the projected moving object and zero pixels represent the background. Throughout the paper the projected data is defined as "2D data" and a group of pixels that represents one object in the 2D data is denoted as "contour."

Figure 1 summarizes the processing steps for the depth data. Firstly, the depth information of the room without any moving objects is captured as the initial background frame. As a result, any moving objects, as shown in the highlighted depth in Fig. 1 (a), would produce dissimilarity of depth data when background subtraction is performed. Next, the pixels of the data in this highlighted area is extracted as in Fig. 1 (b) and then projected onto a 2D plane as in Fig. 1 (c). Finally, a tightly encapsulating rectangle box that represents the region of interest (ROI) for each object's contours is constructed. At this point, the ROIs are initially labeled with an integer value in the order of their relative position to the top-left of the frame.

3.2 Graph-Based Automatic Object Tracking

Object data construction extracts the features of the moving objects from the contours in the 2D data. These features are then utilized in the automatic object detection and tracking. Basically, the width, height, and the centroid of each contour can be extracted and a unique identity assigned to these features as one set of object data. These features are represented in graph structure as described as follows.

Let $G_f = (V_f, E_f, A_f)$ be an attributed graph representation of the f-th frames in 2D data, where V_f is the vertex set, E_f is the edge set and A_f is the set of attributes correlated to vertices in set V_f. The vertex set represents detected objects in frame f and is denoted as $V_f = \left\{v_1^f, v_2^f, \ldots, v_J^f\right\}$ where J is the number of detected objects in the 2D data. Its set attribute $A_f = \left\{a_1^f, a_2^f, \ldots, a_J^f\right\}$ contains sets of attributes for each vertex element v_j^f, $j = 1, 2, \ldots, J$. Each attribute element is defined as

$$a_j^f = \left\{w_j^f, h_j^f, \mathbf{p}_j^f\right\} \tag{1}$$

which represents the width and height of the ROI of a contour, and its centroid $\mathbf{p}_j^f = \left\{p_x, p_y\right\}_j^f$ in the x and y coordinate of the segmented object in the frame. The centroid is calculated as $(p_x)_j^f = x_j^f + w_j^f/2$ and $(p_y)_j^f = y_j^f + h_j^f/2$ respectively.

The edge set is defined as the spatio-temporal relation between the vertices in G_f and G_{f-1}, i.e., two graphs in two consecutive frames. Their relationship is determined as

$$E = \left\{d\left(a_1^f, a_1^{f-1}\right), d\left(a_1^f, a_2^{f-1}\right), \ldots, d\left(a_2^f, a_1^{f-1}\right), \ldots, d\left(a_J^f, a_K^{f-1}\right)\right\} \tag{2}$$

where K is the number of detected objects in frame f-1 and

$$d\left(a_j^f, a_k^{f-1}\right) = \left\|\mathbf{p}_j^f - \mathbf{p}_k^{f-1}\right\| \tag{3}$$

is defined as similarity matching between the attributes of the j-th and the k-th vertices in frame f and frame f-1, respectively, with $k = 1, 2, \ldots, K$. The similarity matching is then performed to find the smallest edge value.

By performing similarity matching from frame to frame, it is possible to correctly identify different objects with unique identities. However, in many cases overlapping objects would still occur, especially for confined spaces where the objects move very close to each other and create occlusion. Therefore, correct identification of an object before and after occlusion is performed by matching the two objects between the current frame and a selected reference frame instead of the previous frame. In this paper, such matching is called a conditional graph matching.

The case of occlusion is determined by observing the position and size of the detected objects in 2D data as illustrated in Fig. 2. When two objects are moving towards each other in frame f-1 and their contours are merged in frame f, the overlapping area of v_1^f with the area of v_1^{f-1} (area R) and v_2^{f-1} (area S) is observed. Assuming that the depth video

has a frame rate of no less than 28fps, if each of those areas are larger than 2/3 of the area of their corresponding ROI, an occlusion is assumed to have occurred. In this case, frame f would be determined as the first frame of an occlusion period and selected as the reference frame for the conditional graph matching. Otherwise, no occlusion is detected and similarity matching based on Eq. (3) is performed.

When occlusion is detected, the attributes of the occluded objects and the timestamp (i.e., frame number) just prior to the occlusion event are stored in an "occlusion list." Conversely, disocclusion is signaled when the merged objects are separated in a frame, and at this point the attribute information will be removed from the occlusion list. Thus that frame will be assigned as the last frame of the occlusion period.

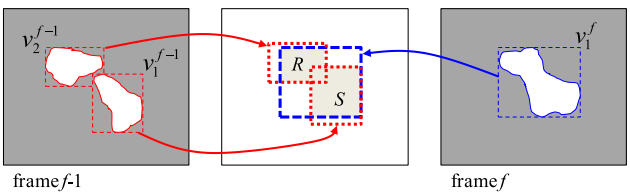


Fig. 2 Illustration of a merge condition between two objects that determines an occlusion.

To handle occlusion, conditional graph matching is performed. If the occlusion period occurs from frame f_N to frame f_M, the similarity matching is calculated as

$$d\left(a_j^{f_M}, a_k^{f_N}\right) = \left\| \mathbf{p}_j^{f_M} - \mathbf{p}_k^{f_N} \right\| \tag{4}$$

where frame f_N is the reference as described by the occlusion list. Since the values of object's attributes as the result of tracking are stored in memory, it is possible to recall the identity of objects for conditional matching. Therefore, conditional matching is also performed to check duplicate identities by assigning the reference frame in which the objects are previously correctly identified. A correct identification is determined when no duplicate identification is detected. Based on the number of occlusions detected in a frame (i.e., merge conditions detected for more than two objects), conditional graph matching is performed sequentially based on the spatial position of the merged objects in the frame.

The overall process of automatic object tracking to handle the behavior of contours in 2D data is as shown in Fig. 3, including handling various occlusion problems. Basically the process checks for two states: the changes of the number of objects in two consecutive frames and whether any overlapped areas as illustrated in Fig. 2 exist. As cate-

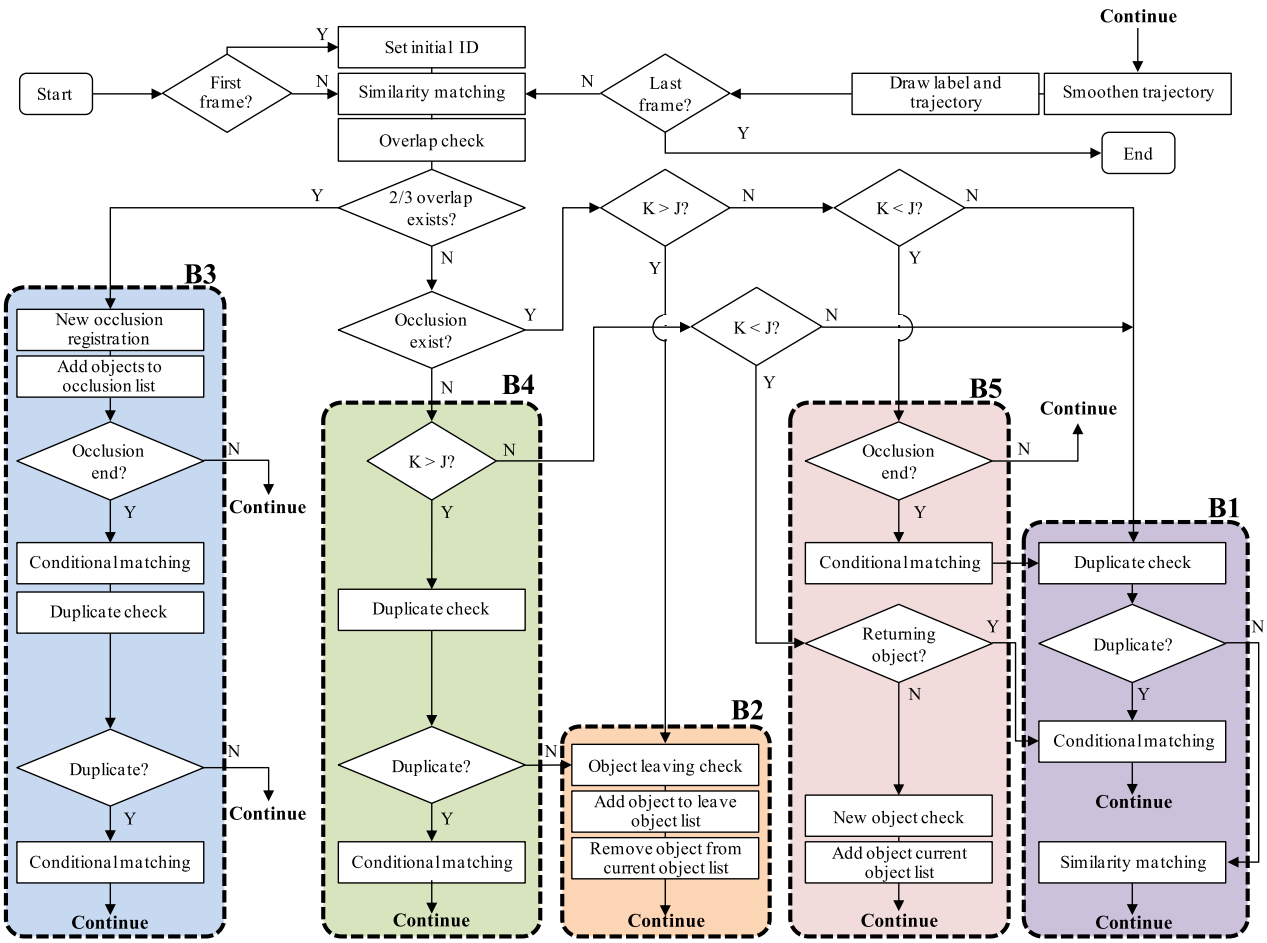


Fig. 3 Overall process of automatic object tracking.

gorized by blocks *B1* to *B5* in Fig. 3, these main procedures are performed based on the checked state in a frame:

- Block *B1*
  In a normal and simple case with no changes in the number of objects, block *B1* will be performed where similarity matching (3) is employed to track the object. In some cases, however, it is possible that the identities of the objects are misassigned, which may cause a duplicate identity or identity switch (e.g., object 1 is assigned as object 2 and vice versa). In this case, conditional matching (4) will be performed with the latest frame in which the correct identification is stored as the reference.
- Block *B2*
  The procedures in block *B2* will be conducted whenever the number of objects is decreasing. In this case, an object that cannot be detected in the next frame will be designated as a "left object" and its attributes will be registered into a "left object list." Depend on its latest location, an object can be also designated as "leaving the frame" (if its last position is near the frame's border) or it is just possibly overshadowed by another object.
- Block *B3*
  When occlusion conditions are met as aforementioned, occlusion handling block *B3* is performed. This block will also check whether the occlusion has ended or any duplicate identification is present. In both cases, a similar operation to block *B1* will be performed. However, the reference of the conditional matching will be adjusted; the last frame before occlusion is assigned as the reference. Furthermore, a duplicate objects check will be performed as in block *B1*.
- Block *B4*
  Technically, procedures in block *B4* are similar to that of block *B1*. However, when duplicate identities are not found, *B4* will check for leaving objects and perform the procedures in block *B2*.
- Block *B5*
  In the case when the number of objects is increasing and no occlusion is occurring, block *B5* will check for returning objects by examining the left object list that was updated in block *B2*. A returned object will be determined by conditional matching between the current frame and the frame in the left object list as the reference. On the other hand, if occlusion has occurred, it will perform conditional matching and furthermore check for duplicate objects as in block *B3*

While not all objects' behavior can be accurately handled, which is demonstrated in Sect. 4, the procedures in blocks *B1* to *B5* ensure relatively reliable object detection and tracking accuracy.

4. Experimental Results and Analysis

The proposed method is tested with test sequences that record the movements of several people in a room using two range sensor cameras to cover the entire room, where each camera covers half of the room as illustrated in Fig. 4. Each camera is specified as described in Table 1. Both cameras are located at a height of 2.7 meters and there is an intersection region located roughly in the center of the room, as illustrated in Fig. 4. This intersection region provides guidance for each camera whether an object is moving from one side of the room to another. If an object is detected in the intersection region, its contour will be detected as the same object based on its relative position to the projected room area.

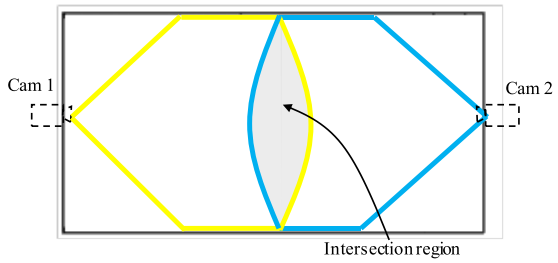


Fig. 4 The setup of two range sensor cameras to cover the entire room used in the experiments.

Table 1 Specification of the cameras used in the experiments

Parameter	Value
Resolution	126×126
Field of view	72°×72°
Coverage distance	0.3m – 4m
Distance accuracy	±180mm
Ambient light performance	100,000Lx

It should be noted that automatic object tracking is not performed independently for each camera. Instead, the 2D projection from camera 1 and camera 2 are first combined into one processing frame. Next, the graph construction and automatic object tracking procedures as shown in Fig. 3 are performed over the combined area, with a resolution of 252×126 pixels.

The proposed method is applied in five test sequences, each of which acquired data from the two range sensor cameras; their frames have been synchronized with each other. In overall, the test sequences contain a room furnished with static objects where people are the only moving objects. Each test sequence has interaction events between two or more people with overlapping trajectories that cause occlusions. The scenario of each test sequence is summarized in Table 2.

Figure 5 shows the depth data acquired from camera 1 and camera 2 in the 184^{th} frame of *Test* 2 sequence and the 657^{th} frame of *Test* 5 sequence, respectively. Note that the highlighted area shown in Fig. 5 denotes the source of the range sensor, not the identification of the detected objects, as at this step the tracking has not been performed. The results of the proposed method in those frames are as shown in Fig. 6. The circles represent the detected moving object identified with different colors and numbers with the

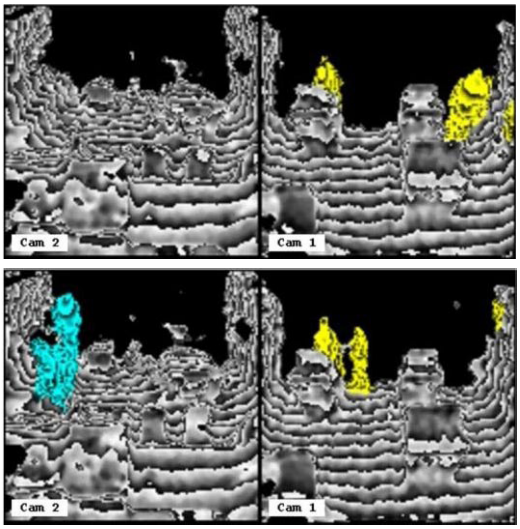


Fig. 5 Depiction of depth data for the objects detected from the 184th frame of Test 2 (above) and the 657th frame of Test 5 (bottom) sequences.

Table 2 Use case scenario of the experiments

Test 1	Three objects with two short duration occlusion events, between each of two objects
Test 2	Four objects with short duration occlusion events between two objects
Test 3	Four objects with long duration occlusion events between three objects
Test 4	Three objects with short duration occlusion events between three objects
Test 5	Four objects with two short duration occlusion events, between each of two objects

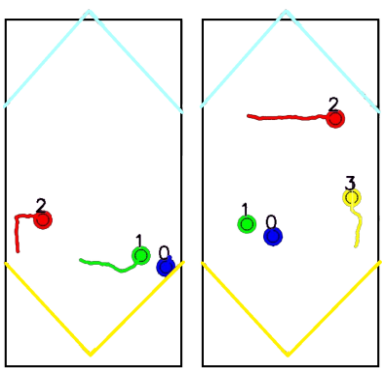


Fig. 6 Object tracking results from the depth data of *Test* 2 (left) and *Test* 5 (right) in Fig. 5.

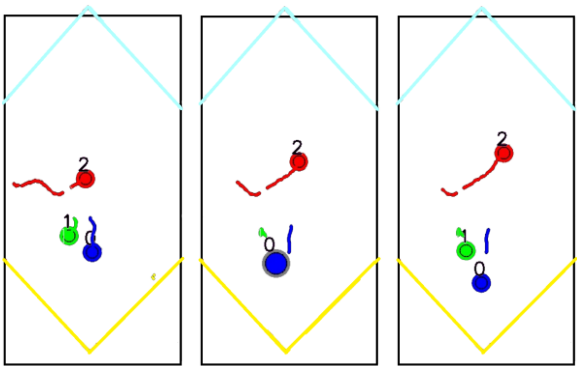


Fig. 7 Occlusion event (left: before occlusion, center: during occlusion, right: after occlusion) from *Test* 5 sequence.

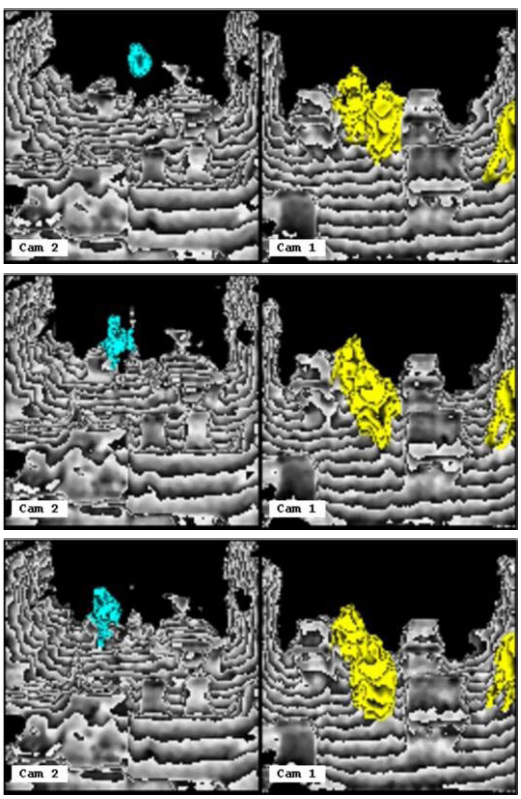


Fig. 8 Depiction of depth data for the objects detected and tracked in Fig. 7 before occlusion (top), during occlusion (middle), and after occlusion (bottom) from *Test* 5 sequence.

embedded lines showing the path the objects had taken from the past 50 frames.

In the event of occlusion, the contours of the merged objects cannot be separated. However, the contour indicates two or more occluded objects are indicated with a dark-colored outer circle. Figure 7 shows the result of object tracking from the 1706th, 1720th and 1734th frame of the *Test* 5 sequence illustrating the tracking of the objects before, during and after occlusion. The corresponding depth data for the same frames are shown in Fig. 8. Here, when *Object* 1 and *Object* 0 are occluded, their contours are merged with each other and indicated as one object (*Object* 0) with a gray-colored outer circle.

Due to its closely related method and similar object tracking results in uniquely identifying each of the detected objects, the proposed method is compared with the centroid-based object tracking algorithm for an infrared sequence proposed by Latecki et al. [20]. The accuracy of the trajectories of tracked objects and its comparison against the trajectories observed by the ground truth is presented. Figure 9 shows the graphs of selected trajectories of objects from the

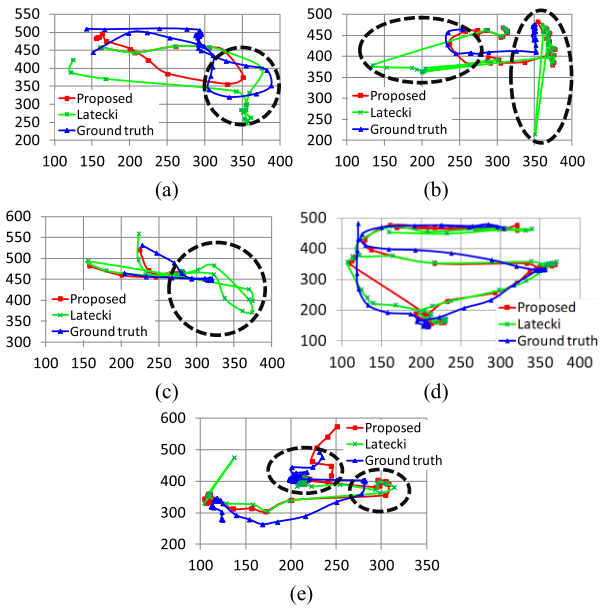


Fig. 9 Trajectories of (a) Object 2 in *Test* 1 sequence, (b) Object 1 in *Test* 2 sequence, (c) Object 3 in *Test* 3 sequence, (d) Object 1 in *Test* 4 sequence, and (e) Object 1 in *Test* 5 sequence, produced by the proposed method and [20] compared against the ground truth.

five test sequences. The horizontal and vertical axes depict the x and y coordinates of the frame respectively.

The performance of the proposed method is measured by how similar are the positions of the detected objects compared to the ground truth. Overall, the produced trajectories from the proposed method and the centroid-based method are relatively in line with the ground truth. The ground truth is obtained by observing the position of the feet of the detected objects. The position of the feet is approximated from the bottom-center part of the ROI of the objects. The x and y coordinates of the ground truth position are then compared with the x and y coordinates of the detected objects using the proposed method. Figure 9 shows the graphs depicting the trajectories of a selected object in sequences *Test* 1 to *Test* 5. The horizontal and vertical axes of the graphs shown in Fig. 9 are the x and y coordinates of the frames, respectively.

The calculation of the position of the objects in the 2D plane in the proposed method is affected by the size of the contours of the objects. Since the position is calculated based on the centroid of the contours, the positions are somewhat deviated from the ground truth. Nevertheless, as can be seen in the graphs, the displacements of the objects during the sequence are consistent with the trajectories of the ground truth. Compared to the centroid-based method, the proposed method has better trajectory accuracy, especially when the objects are involved in occlusion, as shown in Fig. 9 (a), (b), (c) and (e). Here the trajectories produced by the centroid-based method move away from the ground truth. On the other hand, when an object is not involved in occlusion, as in Fig. 9 (d), the trajectories tend to follow the ground truth. The cases of occlusion for the selected objects in Fig. 9 are indicated with dashed circles.

Table 3 Bhattacharyya distance of the produced trajectories compared against the ground truth

Sequence	Frames	Method	Object 1	Object 2	Object 3	Object 4
Test 1	934	[20]	**0.797**	0.050	0.030	n/a
		Proposed	2.255	**0.004**	**0.000016**	
Test 2	1024	[20]	0.086	0.027	0.069	0.434
		Proposed	**0.000014**	**0.016**	**0.044**	**0.129**
Test 3	800	[20]	0.361	0.034	0.067	**0.524**
		Proposed	**0.003**	**0.002**	**0.002**	2.327
Test 4	1500	[20]	**0.00008**	**0.0206**	0.0003	n/a
		Proposed	0.00073	0.0214	**0.0001**	
Test 5	1800	[20]	0.0042	**0.205**	0.0006	**0.192**
		Proposed	**0.0038**	0.308	**0.000056**	0.203

Table 4 Identification accuracy

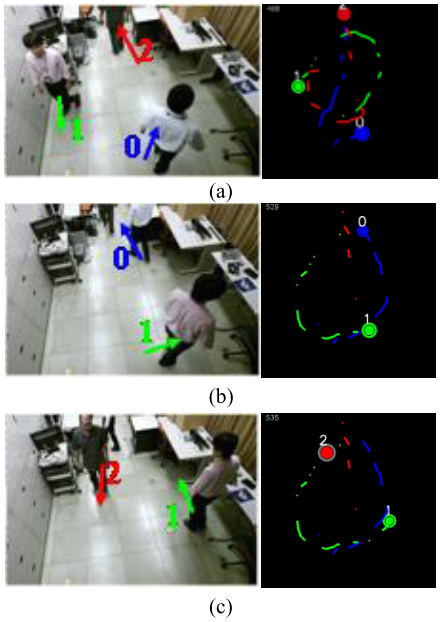
Method	*Test* 1	*Test* 2	*Test* 3	*Test* 4	*Test* 5
[20]	88.9%	72.8%	45.9%	**98.1%**	59.7%
Proposed	**94.5%**	**98.3%**	**87.4%**	84.1%	**92.4%**
△	5.6%	25.5%	41.5%	*-14.0%*	32.7%

The overall trajectory accuracy is shown in Table 3. The distribution of the positions for all objects in all sequences is calculated. Then the Bhattacharyya distance from the positions of objects in the proposed method and the centroid-based method are compared against the ground truth according to the measurement evaluation proposed by Coleman and Andrew [21]. Here, smaller distance values denote a trajectory more similar with the ground truth and bold values indicate better accuracy. As shown in Table 3, the proposed method would produce more accurate trajectories compared to the centroid-based method.

The quantitative measurement of the tracking accuracy of the proposed method is computed by finding how many frames in which the detected objects can be correctly identified out of total frames in the sequence. Table 4 shows the tracking rate of the proposed attributed graph method compared with the result of using the centroid-based method [20]. Since the centroid-based method provides no feature to handle occlusion between two or more objects, its accuracy is relatively lower than the proposed method. In the *Test* 4 sequence, *Object* 1 appeared in more frames while involved in short duration occlusion than the other objects. Thus the accuracy in *Test* 4 sequence for centroid-based method is seen to be higher than the proposed method.

The robustness of occlusion handling in the proposed method depends on how much area of overlapping pixels exists among the occluded objects. While the 2D projection can reduce the overlapping pixels, there are many cases where noise produced by interference in a sensor increase the overlapping pixels among objects. The robustness of the occlusion handling may suffer from such problems.

Some limitations are also present in the proposed method, especially involving ambiguous object identity. Ambiguity in the changes of the number of objects can be produced by a complex situation. For example, when an occlusion has just started and a new object appears at the same time, then there is no change in the number of objects from one frame to the next and the proposed method will assume



(a)

(b)

(c)

Fig. 10 Snapshots of actual video (with manually annotated labels and moving directions) and the corresponding automatic depth-based tracking results show the situation of ambiguous identification of the returning object after two objects leave the frame successively at the same position.

neither occlusion nor new object appearance has occurred.

In another case, consider when two objects are successively leaving the frame at almost the same position, as in the example shown in Fig. 10 (a) and Fig. 10 (b). From the actual video it can be easily seen that a person wearing a dark colored shirt is leaving the frame followed by another person wearing a white shirt. After several frames one of the people returns to the scene at the position where both people previously left as shown in Fig. 10 (c). While both people can be easily tracked using depth information prior to leaving the frame, when one of the people returns to the frame, the features extracted from 2D data become unclear. The position of both people when leaving the frame is almost the same as the position of the one person who returns to the frame.

In such cases, the proposed method may fail to produce correct object identification. While object tracking using a conventional camera can rely on color information to handle such cases, in depth-based object tracking, further exploration of additional features is needed to differentiate objects by utilizing the actual depth values in form of depth histograms [12], [15] or in finding upper body structure [13], [14].

5. Conclusions and Future Work

This paper introduced a method that can automatically detect moving objects and track the objects from depth data, where each object is uniquely identified. Attributed graph structure is employed to represent the features of the objects extracted from projected depth data and utilize the attributes to perform object tracking. The proposed method makes use of the changes of number of objects and observes the interaction between objects to determine which graph similarity matching is the most appropriate to identify objects. Experimental results show that the performance of the proposed method against a comparable method in depth-based object tracking performs better during occlusion.

Future work includes the extension to the current method to use actual depth values to handle more complex occlusions as well as to handle the objects' ambiguity by using actual depth information. Additionally, the proposed method currently can only handle up to four objects, which should also be further improved.

Acknowledgements

This work was supported by "R&D on Ultra-Realistic Communication Technology with Innovative 3D Video Technology", the Commissioned Research of National Institute of Information and Communications Technology (NICT), Japan.

References

[1] American Civil Liberties Union, "What's Wrong With Public Video Surveillance?," https://www.aclu.org/technology-and-liberty/whats-wrong-public-video-surveillance

[2] K.A. Joshi and D.G. Thakore, "A Survey on Moving Object Detection and Tracking in Video Surveillance System," Int. J. of Soft Computing and Engineering, vol.2, no.3, pp.2231–2307, 2012.

[3] H. Blume and B. Heimann, "A Laser Range Scanner Simulation for Probabilistic Object Tracking," ICRA Workshop, 2007

[4] J. Bobruk and D. Austin, "Laser Motion Detection and Hypothesis Tracking from a Mobile Platform," Proc. Australian Conf. Robot. Autom., 2004.

[5] A.T. Tran and K. Harada, "Depth-Aided Tracking Multiple Objects under Occlusion," J. of Signal and Information Processing, vol.4, no.3, pp.299–307, 2013.

[6] J. Liu, Y. Liu, Y. Cui, and Y.Q. Chen, "Real-time Human Detection and Tracking in Complex Environments using Single RGBD Camera," IEEE Conf. Image Processing, pp.3088–3092, 2013.

[7] O.H. Jafari, D. Mitzel, and B. Leibe, "Real-time RGB-D based People Detection and Tracking for Mobile Robots and Head-Worn Cameras," IEEE Conf. on Robotics and Automation, pp.5636–5643, 2014.

[8] H. Fu, H. Ma, and H. Xiao, "Real-time Accurate Crowd Counting based on RGB-D Information," IEEE Int. Conf. on Image Processing, pp.2685–2688, 2012.

[9] E. Bondi, L. Seidenari, A.D. Bagdanov, and A.D. Bimbo, "Real-time People Counting from Depth Imagery of Crowded Environments," IEEE Conf. on Advanced Video and Signal based Surveillance, pp.337–342, 2014.

[10] Z. Wang, Y. Wu, J. Wang, and H. Lu, "Target Tracking in Infrared Image Sequences Using Diverse AdaBoostSVM," Int. Conf. on Innovative Computing, Information and Control, pp.233–236, 2006.

[11] B. Liu, O. Jesorsky, and R. Kompe, "Robust Real-Time Multiple Object Tracking in Traffic Scenes Using an Optical Matrix Range Sensor," IEEE Intelligent Transportation System Conf., pp.742–747, 2007.

[12] J. Li and W. Gong, "Real Time Pedestrian Tracking using Thermal Infrared Imagery," Journal of Computers, vol.5, no.10, pp.1606–1614, 2010.

[13] Q. Tian, B. Zhou, W.-H. Zhao, Y. Wei, and W.-W. Fei, "Human Detection using HOG Features of Head and Shoulder Based on Depth

Map," Journal of Software, vol.8, no.9, pp.2223–2230, 2013.
[14] L. Xia, C.-C. Chen, and J.K. Aggarwal, "Human Detection using Depth Information by Kinect," IEEE Conf. on CVPR Workshops, pp.15–22, 2011.
[15] S. Ikemura and H. Fujiyoshi, "Real-time Human Detection Using Relational Depth Similarity Features," Proc. 10th Asian Conference on Computer Vision, 2010.
[16] D.W. Hansen, M.S. Hansen, M. Kirschmeyer, R. Larsen, and D. Silvestre, "Cluster Tracking with Time-of-Flight Camera," IEEE Conf. on Computer Vision and Pattern Recognition, pp.1–6, 2008.
[17] L. Jia and R.J. Radke, "Using Time-of-Flight Measurement for Privacy-Preserving Tracking in a Smart Room," IEEE Trans. on Industrial Informatics, vol.10, no.1, pp.689–696, Feb. 2014.
[18] C.-T. Hsieh, H.-C. Wang, Y.-K. Wu, L.-C. Chang, and T.-K. Kuo, "A Kinect-based People-flow Counting System," Int. Symp. On Intelligent Signal Processing and Communications Systems, pp.146–150, 2012.
[19] Y. Zhou, Y. Yang, M. Yi, X. Bai, W. Liu, and L.J. Latecki, "Online multiple targets detection and tracking from mobile robot in cluttered indoor environments with depth camera," Int. Journal of Pattern Recognition and Artificial Intelligence, vol.28, no.1, p.1455001, 2014.
[20] L.J. Latecki, R. Miezianko, and D. Pokrajac, "Tracking Motion Objects in Infrared Videos," IEEE Conf. on Advanced Video and Signal Based Surveillance, pp.99–104, 2005.
[21] G.B. Coleman and H.C. Andrews, "Image Segmentation by Clustering," Proceedings of The IEEE, vol.67, no.5, pp.773–785, May 1979.

Houari Sabirin received his Ph.D. in Information and Communications Engineering from Korea Advanced Institute of Technology, Daejeon, Korea in 2012. From 2012-2013 he was a postdoctoral researcher with Information & Electronics Research Institute, in KAIST, Korea. He is now with KDDI R&D Laboratories, Inc.

Hiroshi Sankoh received his B.E. in information science and M.E. in intelligence science and technology, both from Kyoto University in 2006 and 2008, respectively. Since 2008 he has been with Ultra-Realistic Communications Laboratory in KDDI R&D Laboratories, Inc.

Sei Naito received his B.E., M.E., and Ph.D. degrees from Waseda University in 1994, 1996, and 2006 respectively. He joined Kokusai Densin Denwa Corporation (currently KDDI) in 1996. He is currently a senior manager of Ultra-Realistic Communications Laboratory in KDDI R&D Laboratories, Inc.