# Detection of a Specific Moving Object from Head-mounted Camera Images

Katsuma Ishitobi, Joo Kooi Tan, *Member, IEEE*, Hyoungseop Kim, Seiji Ishikawa, *Member, IEEE*

***Abstract*— In this paper, a method is proposed for detecting and tracking a specific moving object (e.g., a bus) on the road from images of a camera attached to the head of a user, aiming at developing a system to support daily lives of visually impaired people. The proposed method traces feature points on the images, extracts a moving object region, and detects a bus by applying Haar-like feature and random trees to the region. The effectiveness of the proposed method is shown experimentally.**

## I. INTRODUCTION

According to the statistics issued by The Ministry of Health, Labor and Welfare [1] in 2011, the number of visually impaired people throughout Japan is about 320,000, among which approximately 69% is elderly people aged 65 years or older. It is predicted that the number of visually impaired elderly people will further increase from now on.

Difficulties in the daily lives of visually impaired people include outdoor activities. Particularly, when going out to a distant place, it is necessary for them to use transportation facilities such as a taxi, a bus or a railroad. However, it is not very easy for visually impaired people to use these vehicles at present.

In the questionnaire conducted to 29 people with visual impairment [2], 18 people answered that they use a bus almost every day, but at the same time 7 people responded that a bus is the most difficult vehicle to use. The reason for this is that it is often difficult to find the entrance.

Therefore, it is necessary to develop a support system for visually impaired to make it easier to use public transportation facilities. Studies on detecting vehicles on the road [3][4][5] are actively conducted, but they are intended for the investigation of traffic volume and automatic driving, and they neither aim at detecting specific vehicles nor support visually impaired people. There is a study aiming at detecting a bus from a head-mounted camera images for the support of a visually impaired [6], but its performance is still insufficient for a practical use.

In this paper, we propose a method to detect and track specific moving objects traveling on the road. First, feature points are detected on a frame in a video using Harris corner detector [7] and tracked in the video to yield optical flows. From these optical flows, we remove the optical flows of camera motion using projective transformation and RANSAC [8]. A moving object region is estimated from the remaining optical flows. Haar-like feature [9] and random trees [10] are applied to this estimated moving object region to find a specific moving object. When it is found, it is tracked using the CamShift [11] algorithm from the next frame. We also detect a specific moving object in the tracking window and update the histogram used in CamShift to adapt to changes in its appearance.

Finally, in order to show the effectiveness of the proposed method, experiments are conducted using the images taken under different weather conditions, at different places, and with different appearances of a specific moving object.

Kastuma Ishitobi is with Graduate School of Engineering, Kyushu Institute of Technology, Japan, (e-mail: ishitobi@ss10.cntl.kyutech.ac.jp).

Joo Kooi Tan is with Faculty of Engineering, Kyushu Institute of Technology, Japan (phone: 81-93-884-3191, fax: 81-93-884-3191, e-mail: etheltan@cntl.kyutech.ac.jp)

Hyoungseop Kim is with Faculty of Engineering, Kyushu Institute of Technology, Japan

Seiji Ishikawa was with Faculty of Engineering, Kyushu Institute of Technology, Japan, (e-mail:ishikawa@ss10.cntl.kyutech.ac.jp).

## II. OUTLINE OF THE PROPOSED METHOD

The proposed method detects and tracks a specific moving object from the images provided by a camera fixed at the head of a user.

The method contains two parts; a training part and an identification part. The flowchart of the training part is shown in Fig.1. The identification part is further divided into two stages, a detection stage of a specific moving object and a tracking stage after the detection. These flowcharts are given in Fig. 2.

In the training part, training is performed by combining Haar-like feature and random trees. The used training images are separated into three classes; a bus image class which is a positive image class, and a class of other vehicles and a class of background images which are negative image classes.

In the identification part, on the other hand, moving object areas are detected from camera images in the first place, and then a specific moving object is searched within these areas. If the specific moving object is detected, the procedure moves to the tracking stage to track it using CamShift.

## III. ESTIMATION OF A MOVING OBJECT AREA

In order to detect a specific moving object, an area where a traveling vehicle exists is estimated in an image. It aims at reducing the identification time and erroneous detection by limiting the area to be identified.

### *A. Feature point detection*

In order to estimate a moving object region, we first detect feature points on an image. In the proposed method, Harris corner detector is used for this purpose.

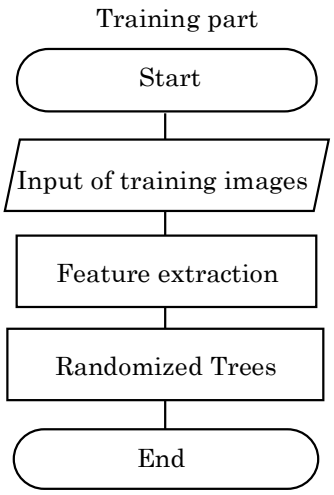



Figure 1. Flowchart of the training part.

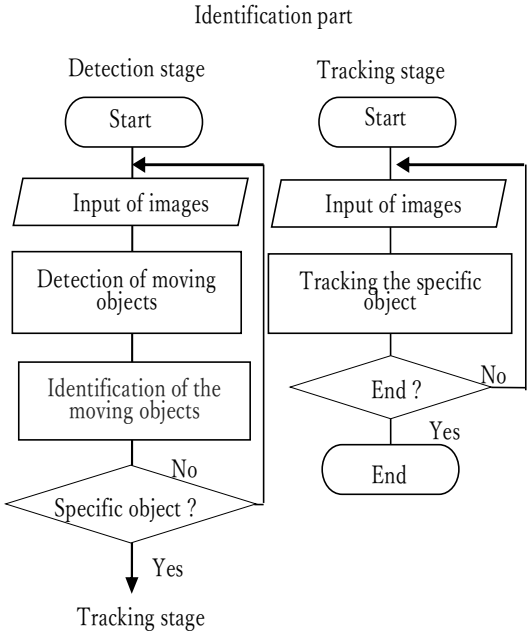



Figure 2. Flowchart of the identification part.

### B. *Feature point tracking*

The detected feature points are tracked by the pyramid LK tracker [12].

### C. *Removal of outliers*

Camera movement is included in the detected optical flows. Therefore, by estimating the camera motion, it is possible to extract only the optical flows of a moving vehicle traveling on the road. For the camera motion model, the following projective transformation model is used.

$$m\begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} = \begin{pmatrix} h_0 & h_1 & h_2 \\ h_3 & h_4 & h_5 \\ h_6 & h_7 & h_8 \end{pmatrix}\begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} \tag{1}$$

where $m$ is the unknown scale, $(x_1, y_1, 1)^T$ is the simultaneous coordinate of a point $(x_1, y_1)$ on the image before the transformation, $(x_2, y_2, 1)^T$ is the same point after the transformation, and $h_i$ $(i = 0,\cdots,7)$ are unknown parameters. From (1), the equation of projective transformation is obtained as follows;

$$\begin{cases} x_2 = \dfrac{h_0 x_1 + h_1 y_1 + h_2}{h_6 x_1 + h_7 y_1 + 1} \\ y_2 = \dfrac{h_3 x_1 + h_4 y_1 + h_5}{h_6 x_1 + h_7 y_1 + 1} \end{cases} \tag{2}$$

The unknown parameters can be obtained if there are four corresponding points. These parameters are estimated by RANSAC, and only the optical flows judged as inliers are extracted. The procedure for RANSAC is explained below.

1) Four optical flows are chosen at random from all the detected optical flows.
2) The model parameters $h_i$ $(i = 0, \cdots, 7)$ in (2) are calculated using the four optical flows. With respect to all the optical flows except for the chosen four optical flows denoted by $F^{A/4}$, their start points $(x_1, y_1)$s are substituted into (2) to get new corresponding points, $(x_2', y_2')$s. Then the difference is calculated between the points $(x_2, y_2)$ derived from the LK tracker and $(x_2', y_2')$ obtained from (2) by

$$E = \sqrt{\left(x_2 - x_2^{'}\right)^2 + \left(y_2 - y_2^{'}\right)^2} \tag{3}$$

If, for a certain threshold $th$, $E<th$ holds with the optical flow in the set $F^{A/4}$, it is counted as an inlier.

3) Repeat steps 1) to 2) $N$ times.
4) Select the parameters having the largest number of inliers.

### D. *Finding a moving object area*

The feature points of inliers extracted in C are clustered by the k-means method according to their locations. As the result, a rectangular area is created around the center of each cluster. If some regions overlap, they are integrated, and, if there are clusters having few feature points, they are not used. The area having the largest number of the feature points is taken chosen as a moving object area to be used for identification. The estimated moving object region is shown in Fig.3. The color of the window is changed with each cluster. Cluster 1 is shown by a green window and cluster 2 is given by a yellow window.

## IV. DETECTION OF A SPECIFIC MOVING OBJECT

Identification is performed in the moving object region estimated by the method described in the previous section. Haar-like features and random trees are used to detect a specific moving object. Three image classes, namely, buses,

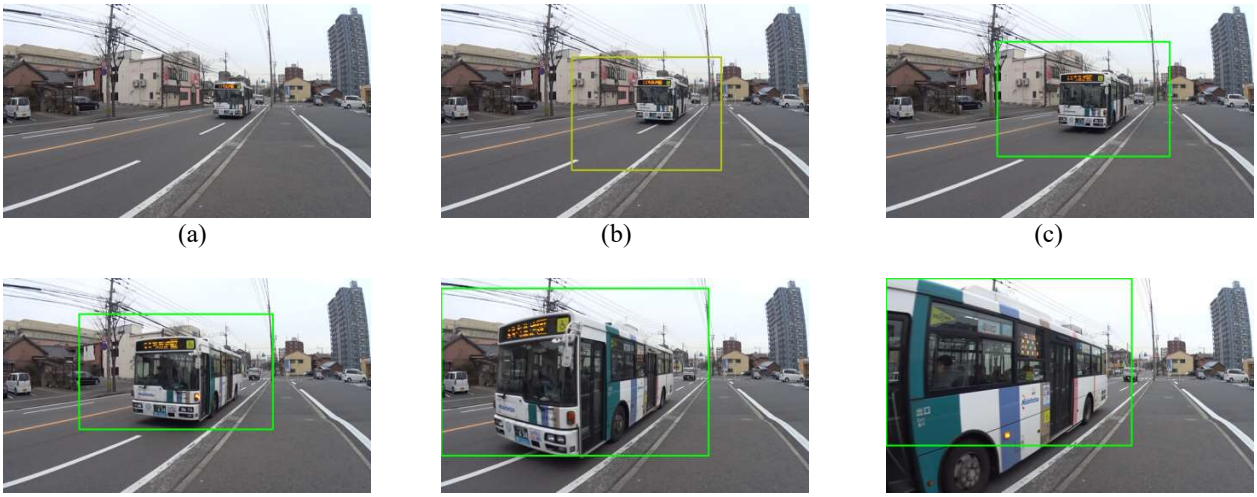


Figure 3. Estimation of a moving object area.

other vehicles, and background images are used as training images as shown in Fig.4.

### A. *Haar-like feature*

Haar-like feature uses local brightness difference focusing on the brightness difference between two adjacent regions. Haar-like feature is often used for face detection and uses the difference in brightness of eyes, nose, mouth, etc., of a face. It is a feature robust to noise and illumination change in an image. We use this feature in considering that the difference in brightness between the windows and the door of the bus and the lower part of the car is effective for its detection. Haar-like feature is calculated by (4) as the difference of the average brightness of the rectangular area of A and B.

$$H(A,B) = f(A) - f(B) \tag{4}$$

Here, $f(A)$ and $f(B)$ are average luminance values of the areas $A$ and $B$, respectively.

### B. *Random trees*

We use random trees that can identify multi-class objects. In addition to being high-speed in the training and the identification resultant from random training, it has the feature that it is robust to the noise included in the training images. The flow of the random trees training is shown below.

1) Subsets are randomly generated from the training images. We create a branching function to determine the branching of the samples in the subset as follows.

$$S_l = \{i \in S_n \mid f(v_i) < t\} \tag{5}$$

$$S_r = S_n \setminus S_l \tag{6}$$

where $S_n$ is a set of all samples in the subset, $S_l$ is a sample set that branches to the left, $S_r$ is a sample set that branches to the right, $f(v_i)$ is a feature amount, and $t$ is a threshold value.


Figure 4. Trees-class training images.

2) The information gain $\Delta E$ is evaluated using the following evaluation function;

$$\Delta E = -\frac{|S_l|}{|S_n|} E(S_l) - \frac{|S_r|}{|S_n|} E(S_r) \tag{7}$$

where $E(S_l)$ and $E(S_r)$ indicate information entropy for each class of samples branched to the left and right, respectively. Information entropy is the appearance probability of samples and is expressed by the following equation.

$$E(S) = -\sum_{i=1}^{n} P_i \log_2 P_i \tag{8}$$

where $P_i$ is the class probability obtained from the teacher signal in the subset, and $n$ is the class number.

3) Repeat steps 1) to 2), and let the combination of feature quantity and the threshold when the information entropy becomes the maximum be the parameters of the branch function.

$$P(c_j \mid l) = \frac{\left|I_{c_j}\right|}{|I|} \tag{9}$$

Here $l$ is the terminal node, $P(c_j \mid l)$ is the class distribution, $|I|$ is the number of samples in all classes, $|I_{cj}|$is the number of samples in class $cj$.

### C. Identification

Identification is performed within the moving object region found by the procedure stated in 3. A specific moving object, a bus in this particular study, is searched in a raster scan manner in the region by using random trees.

When the identification result is the class of a bus and the degree of the match is larger than a threshold and at the same time the maximum, the window containing a bus is displayed as the result of the detection. If a bus class is detected in 5 successive frames, the procedure moves to Tracking Stage as shown in Fig. 2.

## V. Tracking of the Specific Moving Object

A specific moving object is tracked after its detection. For the tracking, CamShift is used. It is an improved version of MeanShift. Since CamShift depends on the initial color histogram of the tracked object, it cannot fully cope with the change in the appearance of the tracked object. Therefore, the histogram is updated repeatedly by detecting the specific moving object even in the process of tracking, so that the tracking may become more accurate. Precise description of the tracking is shown in Fig.5.

### A. CamShift

MeanShift focuses its attention on the histogram of the initially set tracking target area and shifts the window to the area closest to this histogram distribution in the next frame. However, since MeanShift uses a RGB histogram of the tracking target, it cannot deal with illumination change sufficiently. Also, since the size of the window is constant, there is a problem that tracking of an approaching object is difficult. CamShift is the one that improves these difficulties. CamShift uses a hue value histogram. This allows robust tracking under illumination change.

Another idea of CamShift is that it runs MeanShift based on a hue histogram and, when it converges to a certain position, it updates the size of the window by the following equation.

$$s = 2 * \sqrt{\frac{M_{00}}{256}} \tag{10}$$

Here $M_{00}$ is the 0th moment of a tracked object. MeanShift is again applied using the size-updated window from the former converged position. By repeating these procedures, the algorithm finally converges to a preferable position. This is the algorithm of CamShift.

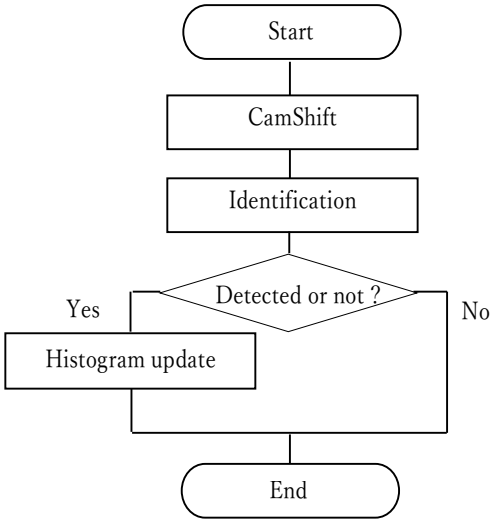



Figure 5. Procedure for tracking.

### B. Tracking algorithm

In the proposed method, tracking is performed by CamShift. Although CamShift can deal with changes in the appearance of a tracked object by updating the window size, the hue value histogram of the object is set initially and the histogram of the approaching object may vary, since, if one observes a tracked vehicle from the sidewalk, one can see initially only the front of the vehicle at a distant location, but as it approaches one can see its side. This suggests the necessity of update of the histogram. To realize this, a larger window including the window for tracking is set and a specific moving object is searched within the new window. Once it is found, the hue histogram of the specific moving object is updated. Thus CamShift performs the tracking not influenced by the change of object appearance. When the tracking window has reached the left end of an image, the tracking ends as the object has come in front of a user.

## VI. Experimental Results

In the experiment, a bus was chosen as a specific moving object and its detection and tracking were performed by applying the proposed method to the videos containing roads and vehicles. Videos were taken in different conditions: Some were taken from the location close to the road, whereas others from the location a little distant from the road. Some videos contain electric poles, whereas others do not. As shown in Fig. 4, the training images are separated into three classes; buses, other vehicles, and background images. The bus class contains 1,800 images, whereas other classes have 1,400 images in all. The image size is 90 × 48 pixels. The filters of the Haar-like feature used in the experiments are depicted in Fig.6.

The specifications of the personal computer and the software used in the experiments are as follows;

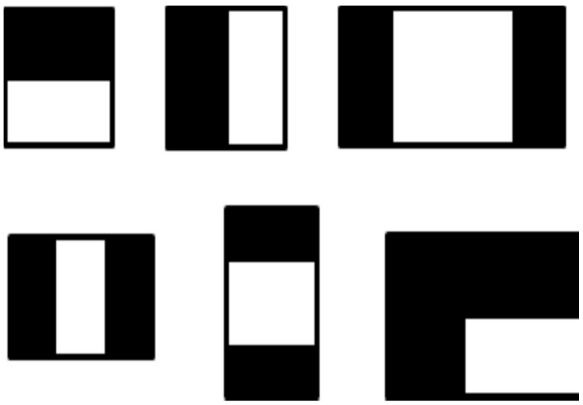
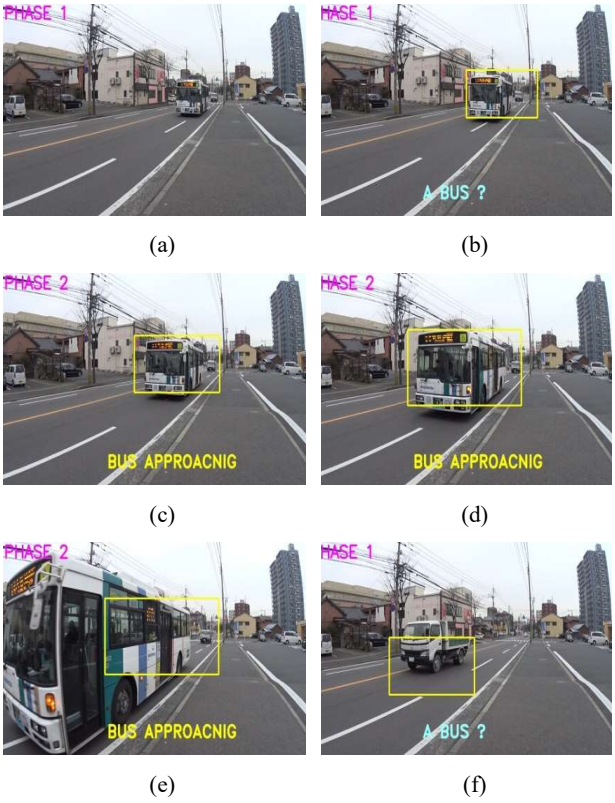


Figure 6. Haar-like feature filters employed in the experiment.



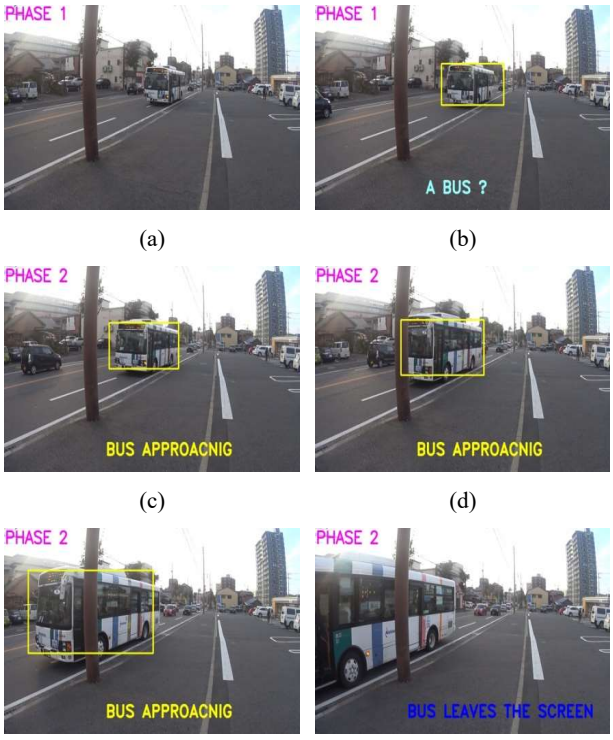

(a) (b) (c) (d) (e) (f)

Figure 7. Experimental result: The case a user stands near the road.

OS: Windows 10 Home, CPU: Intel Core i7-3770@3.4GHz, RAM: 8.00GB, Programing language: Visual C++.

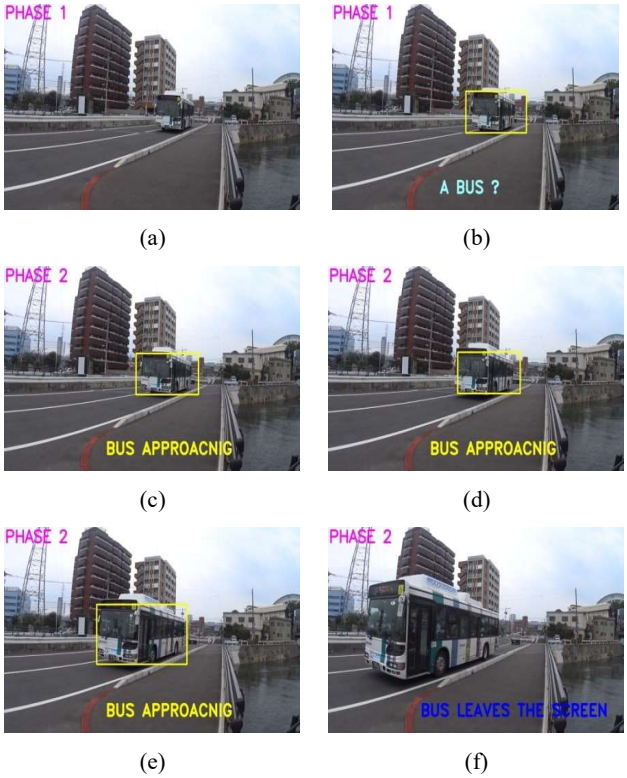
Experimental results are given in Fig. 7 to Fig. 9. Figure 7 is the result using a video taken by standing near the road. In (a), the bus is not detected, since it is far and smaller than the minimum size window. In (b), the bus is detected and, since, in (c) and after, it was detected in five consecutive frames, the procedure has shifted to the tracking stage.

During the tracking, the display 'BUS APPROACHING' appears on the image as well as a yellow window indicating the detected bus. However, in (e), since the program didn't recognize that the tracking window has come to the left-most end of the image, the tracking continued even after the head of




(a) (b) (c) (d) (e) (f)

Figure 8. Experimental result: The case a bus is partly hidden by the electric pole.




(a) (b) (c) (d) (e) (f)

Figure 9. Experimental result: The case a user stands a little away from the road.

the bus passed in front of the user. In (f), erroneous detection occurred in the area including the lower side of the track and the road. But, as it was not detected in five successive frames, the track was not detected as a bus.

In the video of Fig. 8, a utility pole stands along the road and occludes the bus as seen in (d) and (e), but the detection window manages to detect the bus correctly. In (f), the bus is exactly judged as passed in front of the user and the program displays 'BUS LEAVES THE SCREEN' on the image.

Figure 9 is a picture taken from a position away from the road. In this case, the feature of the bus is enhanced by a better view and the detection is successful.

The average processing time per frame in the experiment is 1.34 [s]. Evaluation on the moving object area is performed only with the RECALL value, and its average value is 92.82 [%].

## VII. Conclusion and Discussion

In this paper, we proposed a method of detecting and tracking a specific moving object, in particular a public bus, aiming at supporting daily lives of visually impaired people. For the detection, Haar-like feature quantity and randomized trees which can do multi-class identification were used, and CamShift was used for the tracking of a bus. Experiments were conducted using actual images and the effectiveness of the proposed method was demonstrated.

The present method performs detection and tracking of a bus. However, in order to support a visually impaired person practically, it is necessary to improve the method so that it detects the stop position of a bus and indicates its entrance to the user. This remains for further study.

In this paper, bus detection is dealt with. Although any vehicles or pedestrians can be detected by the proposed method only if their respective shapes are learned, this paper focuses its attention on bus detection at the moment. People with visual impairment use taxis in addition to buses. A taxi detection method is already proposed in [13]. Inclusion of the method in the present study will result in providing a more useful method of finding convenient means of transportation, buses and taxis, for visually impaired.

In addition, there are studies on the detection of traffic lights and signs [14], and the detection of pedestrians and their attributes (height, walking direction, adult/child, etc.) [15] for visually impaired people to walk safely on the road. The goal of the present research is to develop a comprehensive safety support system for visually impaired people when they go out by integrating these research results with the method proposed in this paper.

In this study, random trees is employed for recognizing objects. The use of deep learning in recognition problems is another way of achieving better results. However, deep learning has some problems. The largest problem is that why and how deep learning provides high performance remains unknown. Another problem is that enormous training data is required for training. When one or a few objects need to be detected, a huge amount of data is not necessary. The use of random trees makes it possible to analyze the factors affecting recognition results, and it may allow to use less amount of training data compared to deep learning.

A method of removing the influence of camera motion from images is described III. It refers to the literatures [16], [17] which deal with the compensation of camera motion when detecting and tracking a moving object by a hand-held camera.

## Acknowledgment

This work was supported by JSPS KAKENHI Grant Number 16K01554, which is greatly acknowledged.

## References

[1] The Ministry of Health, Labor and Welfare, 2011 Report on the Survey on the Difficulties in the Living of Homebound Impaired Children/Adults, p.13, 2013.

[2] A. Higashiyama, On the Matter Visually Impaired People Feel Inconvenience, http://www.ritsumeihuman.com/uploads/publication

[3] S. Sivaraman, M. M. Trivedi, "Real-time vehicle detection using parts at intersections", Proc. 2012 15th IEEE Int. Conf. on Intelligent Transportation Systems, Anchorage, Sept., 16-19, 2012.

[4] G. Jun, J. K. Aggarwal, M. Gokmen, "Tracking and segmentation of highway vehicles in cluttered and crowded scenes", in *Proc*. IEEE Workshop on Applications of Computer Vision (WACV 2008), pp. 1-6, 2008.

[5] N. K. Kanhere, S. J. Pundlik, S. T. Birchfield, "Vehicle segmentation and tracking from a low-angle off-axis camera", in *Proc. CVPR*, pp. 1152-1157, 2005.

[6] S. Hatano, J. K. Tan, H. Kim, S. Ishikawa, "Specific moving objects detection based on co-occurrence features considering the occlusion from self-wearable camera images", in *Proc*.The 33rd SICE Kyushu Branch Annual Conference, 2pages, 2014. (in Japanense)

[7] C. Harris, M. Stephens, "A combined corner and edge detector", in *Proc*. of the 4th Alvey Vision Conference, pp.147-151, 1988.

[8] M. A. Fischler, R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography", *Communication of the ACM*, Vol. 24, pp.381-395, 1981.

[9] P. Viola, M. Jones, "Rapid object detection using a boosted cascade of simple features", in *Proc. CVPR*, Vol. 1, pp.511-518, 2001.

[10] F. Breiman: "Random forests", *Machine Learning*, Vol. 45, No. 1, pp.5-32, 2001.

[11] G. R. Bradsky, "Computer vision face tracking for use in a perceptual user interface", *Intel Technology Journal*, Vol. 2, pp.1-15, 1998.

[12] J. Y. Bouguet, "Pyramidal implementation of the Lucas Kanade feature tracker description of the algorithm", Intel Corporation, Microprocessor Research Labs, OpenCV Documents, 2000.R. W. Lucky, "Automatic equalization for digital communication," *Bell Syst. Tech. J.*, vol. 44, no. 4, pp. 547–588, Apr. 1965.

[13] A. Nishimura, J. K. Tan, H. Kim, S. Ishikawa, "Detecting a taxi from a video for visually handicapped people", Proceedings of SICE Annual Conference 2015, pp. 89-92, 2015

[14] T. Kumano, J. K. Tan, H. Kim, S. Ishikawa,"Traffic signs and signals detection employing the MY VISION system for a visually impaired person", *ICIC Express Letters, Part B: Applications*, Vol. 7, No. 2, pp. 385-391, 2016.

[15] R. Sakai, J. K. Tan, H. Kim, S. Ishikawa,"Detecting a pedestrian and extracting their attributes from self-mounted camera views", *ICIC Express Letters, Part B: Applications*, Vol. 7, No. 2, pp. 279-286, 2016.

[16] J. K. Tan, S. Ishikawa, S. Sonoda, M. Miyoshi, T. Morie,: "Moving objects segmentation at a traffic junction from vehicular vision", *ECTI Transactions on Computer and Information Technology*, Vol. 5, No. 2, pp. 73-88, 2011.

[17] F. X. A. Setyawan, J. K. Tan, H. Kim, S. Ishikawa,: "Moving objects detection employing iterative update of the background" , *Artificial Life and Robotics*, Vol. 22, No. 2, pp. 168-174, 2017.