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

On-line Object Segmentation through Human-Robot Interaction

Soohwan Kim, Dong Hwan Kim, and Sung-Kee Park

Abstract— **In this paper we propose a new method for on-line object segmentation through human-robot interaction. Particularly, we define three types of human gestures for object learning by the size of target objects; holding small objects, pointing at medium ones and contacting two corners of large ones. The regions of interest where objects are likely to be located are interpreted from those gestures and represented as rectangles in captured images. For object segmentation, we suggest a marker-based watershed segmentation method which segregates an object within a region of interest in real-time performance. Experimental results show that the segmentation quality of our method is as good as that of the GrabCut algorithm, but the computational time of ours is so much faster that it is appropriate for practical applications.**

I. INTRODUCTION

In order to navigate and perform service tasks in natural human environments, robots need to learn and recognize objects. For example, objects like wall frames or sofas can be used as visual landmarks for localization [1][2] or a user may wants his or her robot to search for an object which was taught during human-augmented mapping [3].

However, most researches on object recognition [4][5][6] assume that objects are already annotated or segmented in training images. This assumption is not valid in robotic applications because it is inconvenient for common users to label objects in captured images by hand. Therefore, it is required for robots to be able to automatically recognize what their users want them to learn in the environments.

Technically, this kind of interactive object learning can be divided into two procedures: human-robot interaction and object segmentation. Then, the scenario is normally as follows. While a user naturally interacts with a robot, he or she attracts the robot's attention and gives a hint about the object position in an environment. The robot, on the other hand, recognizes the region of interest through interpreting the user's gesture and segments the target object in the captured image.

This work was supported by the Intelligent Robotics Development Project of the 21st Century Frontier R&D Programs of the Ministry of Knowledge and Economy, Republic of Korea.

Soohwan Kim is a research scientist of Korea Institute of Science and Technology, 39-1 Hawolgok-Dong, Wolsong-Gil 5, Seongbuk-Gu, Seoul, Republic of Korea, kimsoohwan@kist.re.kr

Dong Hwan Kim is a research scientist of Korea Institute of Science and Technology, 39-1 Hawolgok-Dong, Wolsong-Gil 5, Seongbuk-Gu, Seoul, Republic of Korea, gregorykim@kist.re.kr

Sung-Kee Park is a principal research scientist of Korea Institute of Science and Technology, 39-1 Hawolgok-Dong, Wolsong-Gil 5, Seongbuk-Gu, Seoul, Republic of Korea, skee@kist.re.kr

II. RELATED WORK

A. Human-Robot Interaction for Object Learning

In general, multi-modal interaction systems which utilize speeches and gestures together are employed for interactive object learning. Ghidary et al. [7] developed a home robot positioning system(HRPS) for robot navigation. Particularly, in order to generate an object-based environmental map, they proposed a multi-modal interaction using natural language and hand gestures like saying "This is a TV", while pointing at that TV. Haasch et al. [8] also proposed a multi-modal object attention system(OAS) for a mobile robot. When a user is pointing at an object and tells what color it is, the robot recognizes the context area and segments the object. In addition, they employed a finite state machine to detect unknown objects.

In the meantime, pointing or hand gestures are usually used for human-robot interaction for object learning. Kahn el al. [9] proposed the Perseus system for interpreting pointing gestures. To do that, they specially defined five object representations: person, background, floor, light and small-isolated-object. Roth et al. [10] applied a Maximally Stable Extremal Region(MSER) tracker to detect changes in scenes and segment hand-held objects. In addition, Arsenio [11] proposed an active and interactive object system which detects an event when a user is waving his or her finger on an object, showing an object, or slapping an object in front of the robot.

B. Object Segmentation

Traditionally, there have been many attempts to the feature-space based segmentation which maps each pixel of an image into a color space and clusters those point clouds. Pappas [12] generalized the k-means algorithm [13] by including spatial constraints and accounting for local intensity variations in an image. Comaniciu and Meer [14], on the other hand, applied the mean-shift algorithm [15] to image segmentation. It is a non-parametric procedure for detecting modes while a kernel is moving toward the direction of maximum increase in the density. However, those clustering algorithms suffer from difficulties that the number of clusters should be determined a priori and the clustering results depend on the initial set of clusters, which fails to gain desirable results.

Recently, many researches have been devoted on the graph-based image segmentation method which represents an image as a graph, where each node corresponds to a pixel in an image and the weight associated with each edge is proportional to the pixel affinity. Shi and Malik [16] viewed the

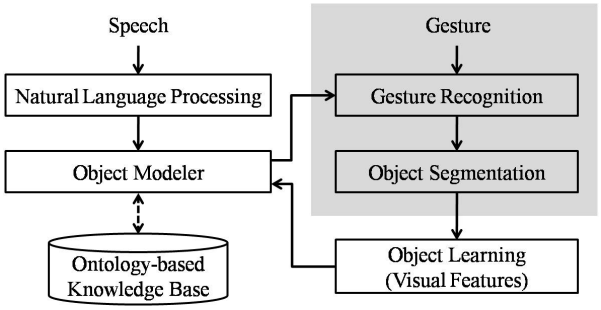


Fig. 1. Our multi-modal interaction framework for object learning

image segmentation problem as a graph partitioning problem and proposed the normalized cut with an eigenvector-based approximation method. Boykov and Jolly [17] introduced user interactions to image segmentation which designate object and background parts of an image. They solved the max-flow/min-cut problem between source and sink nodes in directed graphs. Rother et al. [18] expanded this approach by replacing the histograms of a gray image with Gaussian Mixture Models(GMM) for colors and adapting an iterative procedure for energy minimization. However, those graph-based approaches are usually so time-consuming that it is not appropriate for many practical applications.

Finally, watershed segmentation algorithm [19] has an analogy to punch holes in local minimums of a basin and immerse it under water to find watershed lines. Since it normally leads to over-segmentation of an image, additional pre-/post-processings are needed. Meyer [20] proposed a marker-based watershed segmentation for color images which prevents over-segmentation and reduces computational time.

C. Our Approach

Fig. 1 shows the flowchart of our multi-modal interaction framework for object learning. When a user speaks a command, the robot pays an attention to him or her and interprets the order. For an object learning case, the object modeler ignites the interactive object learning process. Then, the robot recognizes the user's gesture and segments an object from captured images. Next, visual features are extracted and learned for object recognition. Finally, the object modeler anchors the symbol from the command to the visual features from the training images and stores them to the ontology-based knowledge base.

Particularly, since it is too much to cover the whole framework in this paper, we only deal with the interactive object learning parts of the framework which are marked gray in Fig. 1. For the first part, human-robot interaction, we expand our previous work [21] and propose three types of gestures according to the size of target objects; holding small objects, pointing at medium ones and contacting two corners of large ones. And for the second part, object segmentation, we suggest a marker-based watershed segmentation method which is appropriate for interactive applications. In addition, we demonstrate our method and compare it with the GrabCut algorithm on the same conditions.

TABLE I
THREE TYPES OF HUMAN GESTURES FOR OBJECT LEARNING

Size	Objects	Gesture	# of Views
Small	book, pencil sharpener, table clock, mug cup, doll, etc.	holding	1-view, multi-views
Medium	wall clock, wall frame, TV, juice dispenser, dishwasher, etc.	pointing	1-view, 2-views
Large	desk, table, bed, sofa, refrigerator, etc.	contacting	multi-views

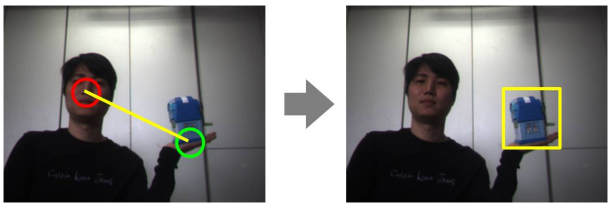


Fig. 2. Holding a pencil sharpener (small object) on the user's hand

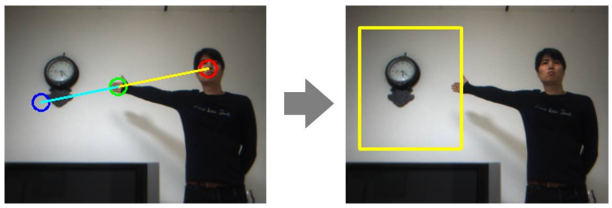
III. HUMAN-ROBOT INTERACTION FOR OBJECT LEARNING

In this section, we define three human gestures according to the size of learning objects, as shown in Table I. You might think that one unified gesture would look simple and consistent, but it is advantageous to divide cases and specify appropriate gestures for better segmentation performance. This is also because the purposes of learning objects differ according to the size; small objects will be used for find-and-fetch, while medium and large objects for vision-based localization and obstacle avoidance.

When a user selects which gesture will be applied for human-robot interaction, face and hand detection is performed first. Here, we apply a cascade of boosted classifiers with Haar-like features [22][23] for face detection. With the color histogram of the detected face, the user's hand is extracted from the back projection image. Those face and hand detection is an initialization process for all three types of gestures in this paper. Note that the result of human-robot interaction for object learning is a rectangular region of interest, and it will be given to the object segmentation process with captured images.

A. Holding Small Objects

Small objects like books, pencil sharpeners, table clocks, mug cups and dolls are easy to lift up. That is why we decided to hold a small object on one's hand and show it to a robot, saying "this is the object you have to learn." Pointing an object on a desk is not suitable for our case, because we assume that a robot is not fixed but moving around the environment, and there can be no desks to put objects on. Also, note that we decide not to wave a small object in front of a robot nor to shake a hand to indicate it since the scene may be dynamic.



(a) 1-view: the user and the wall clock in one view

(b) 2-views: the user in one view and the wall frame in another view

Fig. 3. Pointing at medium objects

Fig. 4. Contacting two corner points of a sofa(large object) for multiple views; the robot captures four images and merges them into one

Fig. 2 shows the result of interpreting a holding gesture. The region of interest is determined by those pixels which have similar disparity values with the user's hand. One thing to be noticed here is that the views of small objects can vary dramatically rather than medium or large objects. Thus, we turn the object on the hand and capture several images for better recognition results.

B. Pointing at Medium Objects

Medium objects like wall clocks, wall frames, TVs, juice dispensers and dishwashers are usually attached on the wall or placed on something flat. Thus, we decide to employ pointing gestures for medium objects, saying "that is the object you have to learn."

Pointing is the most common and intuitive gesture for humans to refer an object in environments. But for a robot it is not that easy; the only thing a robot can be informed by a pointing gesture is a direction (a vector from the face point to the hand point in 3D space). In order to exactly identify what the user is pointing at, another information is required: the distance along the direction.

By the way, sometimes the user and the object can be seen in one view as in Fig. 3(a), but other times the robot needs to pan its head to see the object as in Fig. 3(b). Here, we apply different methods for each case to estimate the distance along the pointing direction. For the 1-view case, we build a 3D virtual cube along the pointing direction and find where the most of point clouds are located from the disparity image. After estimating where the object is located, the 3D virtual cube is projected on the image and its bounding box becomes the region of interest. The details is explained in our previous work [21].

For the 2-view case, it is realistically impossible to pan the camera along the direction continuously and estimate the distance in every captured image like in the 1-view case. But, in most of real situations the distance was between 0.8m and 1.3m, and so we set the distance to 1m. If it fails to find a region of interest in 1m, the robot is supposed to look at another view of 2m. How to find a region of interest in a disparity image is the same for two cases.

C. Contacting Two Corners of Large Objects

The main problem of this case is that a large object itself is too big to be seen in one view. Thus, we decide to capture multiple images and combine them into one single image to apply the same object segmentation method as the previous cases. Particularly, since large objects like desks, tables, beds, sofas and refrigerators are usually placed on the ground, we define contacting two upper corners of a large object to designate how big it is, saying "the object you have to learn is from here to there."

As shown in Fig. 4, first of all, the robot considers the position of the user's hand as the top-left corner point in the world coordinates. Then, the bottom-left corner point is calculated by projecting the top-left corner point to the ground. After tracking the user while he or she moves to the other side of the object, same thing happens to the top-right and bottom-right corners. Given the intrinsic and extrinsic parameters of a camera, a robot can compute how much to pan and tilt and how many images to capture to cover those four corners. Finally, the region of interest is constructed by projecting the four corner points from the world coordinates to the image coordinates.

IV. OBJECT SEGMENTATION

Now, we have an image and a rectangle on it where an object is expected to exist through human-robot interaction. In this section, we explain how to segment an object in the given image with the region of interest.

Of course, you can stop the process here and consider the whole rectangle as an object because for localization, for example, it still works as a visual landmark. However, since the region of interest is a rough approximation, refined segmentation is better for object recognition.

As mentioned earlier, we adopt a watershed segmentation approach for a real-time performance, since it is critical in human-robot interaction. Fig. 5 describes the overall flowchart of our object segmentation method. Note that the original image of Fig. 5(a) is composed with four views as you can see the break lines on the border of each image.

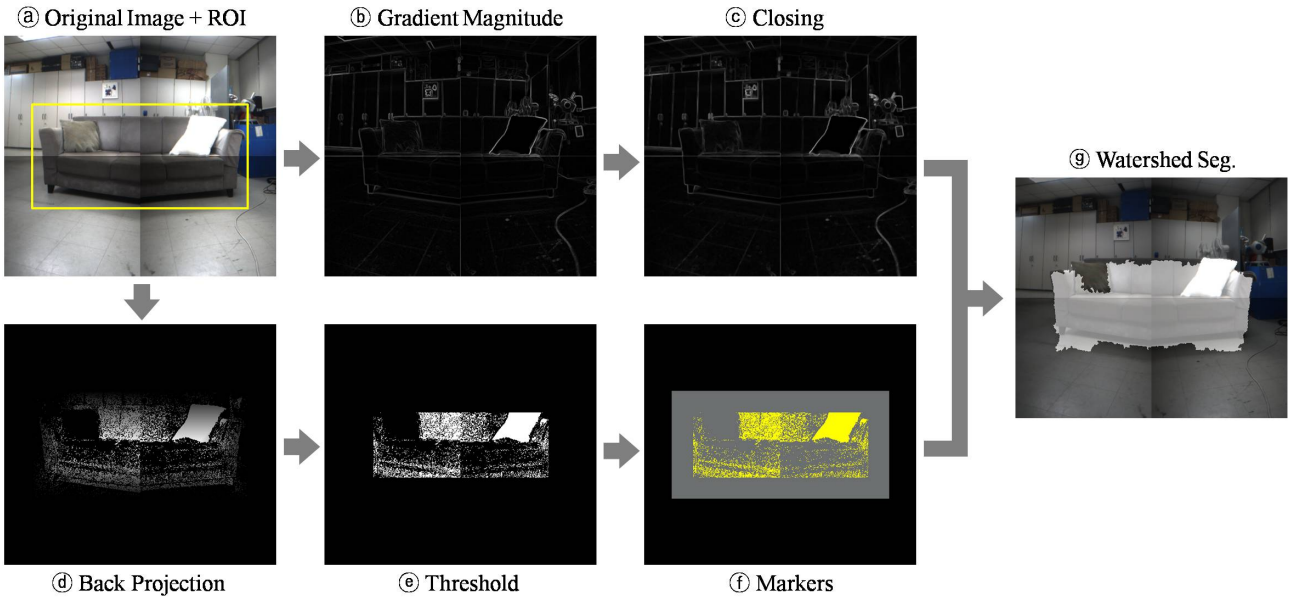


Fig. 5. The Overall Flowchart for Our Object Segmentation Method

A. Gradient Magnitude

As shown in Fig. 5(b), we compute the gradient magnitude, G to detect meaningful discontinuities in an image:

$$G(x,y) = \sqrt{\left(\frac{\partial I(x,y)}{\partial x}\right)^2 + \left(\frac{\partial I(x,y)}{\partial y}\right)^2}, \quad (1)$$

where $I(x,y)$ is the intensity of each pixel.

Here, we employ the CIELAB color space which is a perceptually uniform color space that has proven to perform better than RGB for color texture analysis [24]. Thus, we calculate the gradient magnitudes in L*, a* and b* spaces and merge them into one. In addition, since a noisy gradient introduces over-segmentation, we take the maximum gradient over all color spaces:

$$\widehat{G} = \max\{\omega_L G_L, \omega_a G_a, \omega_b G_b\}, \quad (2)$$

where $\widehat{G}$ is the final gradient magnitude, while G_L, G_a, and G_b correspond to the gradient magnitudes, and ω_L, ω_a and ω_b are the weight coefficients for each color space.

B. Closing

But, there is still noise in the gradient image; some boundaries are double-lined or broken. Thus, we perform a morphological transformation, closing which is a combination of the dilation and erosion to remove noise further in the gradient image. Fig. 5(c) shows the result of closing operation to the gradient image.

The result grayscale image is considered as a topographic relief of which each pixel stands for the elevation at that point. Of course, you can perform watershed transformation with this result image. But we make a major enhancement here to prevent over-segmentation and to specify which parts should belong to the foreground; control markers. How to generate markers automatically will be explained in the following subsections.

C. Back Projection

The watershed transformation partitions the image into two different sets, catchment basins and watershed lines. As a result, you have to group some catchment basins to designate an object in postprocessing. Instead, you can set different markers for different regions which play roles of starting points for flooding.

Fortunately, we have a clue to segregating the foreground from the background; the region of interest. We assume that the outside of the region of interest completely belong to the the background. Thus, we build two 2D color histograms from the inside and outside of the region of interest in the CIELAB color space and estimate the foreground histogram by subtracting the latter from the former:

$$h_f(i,j) = \max\big(h_in(i,j) - h_out(i,j), 0\big), \quad (3)$$

where h_f, h_in, and h_out represent the estimated foreground, inside and outside color histograms, respectively, and i and j denote the indices of the a* and b* bins, respectively.

Another assumption is that the object lie in the middle of the inside. Thus, we create a back projection image with the estimated foreground histogram and apply a distance transform as a weighting factor:

$$D(x,y) = \min\big(|x-x_1|, |x-x_2|, |y-y_1|, |y-y_2|\big), \quad (4)$$

where $D(x,y)$ represents the distance map at each pixel

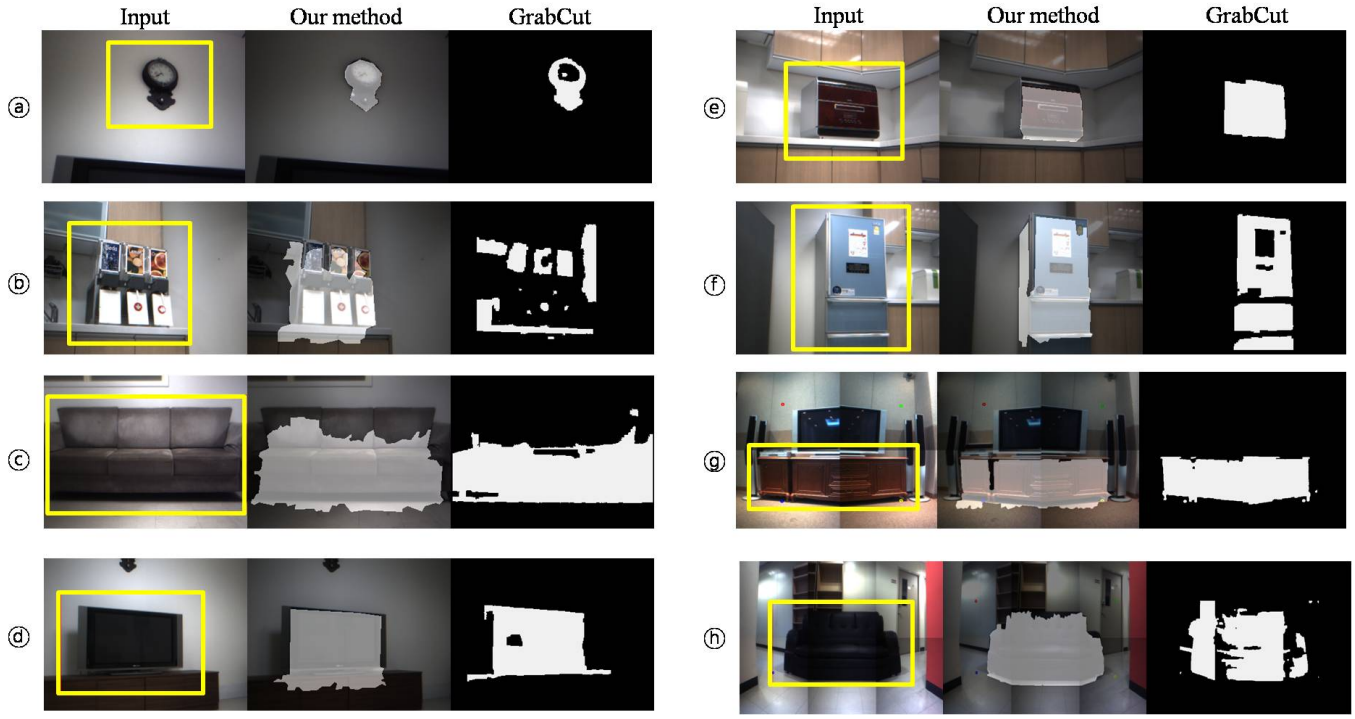


Fig. 6. Experimental Results of Our and GrabCut Algorithms Given the Same Images and Regions of Interest

(x, y) in the region of interest, and (x_1, y_1) and (x_2, y_2) are the top-left and bottom-right points of the region of interest.

Fig. 5(d) shows the back projection image multiplied by the distance map. As you can see, the outside of the region of interest is completely black, but some parts of the sofa look bright. Note that the intensity goes higher from the border to the center of the region of interest in general, which is caused by the distance transform.

D. Threshold and Markers

In order to make a decision for the foreground in the region of interest, we reject those pixels whose intensity values are below a threshold. The remained pixels in Fig. 5(e) become the foreground markers, while the outside of the region of interest is filled with the background markers in Fig. 5(f). In the next step, the watershed transformation will mark the unknown regions which are not labeled as foreground nor background.

E. Marker-driven Watershed Segmentation

Fig. 5(g) shows the result of the marker-driven watershed transformation [20] from the topographic relief, Fig. 5(c) and the foreground/background markers, Fig. 5(f). The highlighted region stands for the segmented object. As you can see, most of the sofa including a cushion is successfully segmented, but some parts around the top-left corner are missing.

V. EXPERIMENTAL RESULTS

Fig. 6 shows the experimental results of object segmentation, given captured images and their regions of interests. In order to evaluate the performance of our method, we also tested the GrabCut algorithm (number of GMM models: 3) in the same conditions.

As you can see, in most of cases like Fig. 6(a), (c), (d), (e), and (f) the results are almost similar. But in some cases like 6(b) and (h) our method shows better performances, while in the other one like Fig. 6(g) GraCut outperforms our method.

In general GrabCut was good at color-featured objects and produced elaborated segmentations, while our method was good at achromatic objects and showed robust results. However, in time complexity two algorithms shows big differences. Our method took about 30 milliseconds for 320×240 images and about 150 milliseconds for 960×480 images, while GrabCut algorithm took about 2 seconds for 320×240 images and about 5 seconds for 960×480 images on a desktop PC with a 2.4GHz dual-core CPU and two 2GB RAMs.

In conclusion, our method more suitable for interactive object learning since the user usually wants to check whether the robot has segmented well after referencing objects.

VI. CONCLUSIONS AND FUTURE WORKS

In this paper we proposed a new method for interactive object learning from the view point that it is an integration of human-robot interaction and object segmentation.

For human-robot interaction, we defined three types of human gestures by the size of the target object; holding, pointing and contacting two corners for small, medium and large objects, respectively. As a result, the region of interest where the object is likely to be located in the environments

was estimated and expressed as a rectangle in the captured image.

For object segmentation, on the other hand, we suggested a marker-based watershed segmentation of which uses a noise-filtered gradient image and forground/background markers estimated from the color histograms of the region of interest. Experimental results showed that the segmentation quality of our method is as good as that of the GrabCut algorithm, but the computational time of ours is so much faster that it is appropriate for practical applications.

In this paper, extracting visual features for object learning and recognizing learned objects in the environments is thought as a straightforward process and not included, but it is necessary to evaluate interactive object learning as a whole. In the meantime, registering learned objects to environmental maps and localizing with them is another issue to be solved. For example, large objects can be expressed as occupied areas on occupancy grid maps for localization and path generation. We think that our method can play a basis for those researches in the future.

VII. ACKNOWLEDGMENTS

The authors would like to thank prof. Kuk-Jin Yoon in Gwangju Institute of Science and Technology for implementing the GrabCut algorithm for benchmarks.

REFERENCES

[1] Soonyong Park, Soohwan Kim, Mignon Park, and Sung-Kee Park, "Vision-based Global Localization for Mobile Robots with Hybrid Maps of Objects and Spatial Layouts," *Information Sciences*, Vol. 179, No. 24. pp. 4174-4198, 2009.

[2] Soonyong Park, Howon Cheong, and Sung-Kee Park, "Coarse-to-Fine Global Localization for Mobile Robots with Hybrid Maps of Objects and Spatial Layouts," *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 1, pp. 3993-4000, 2009.

[3] Soohwan Kim, Howon Cheong, Ju-Hong Park, and Sung-Kee Park, "Human Augmented Mapping for Indoor Environments using a Stereo Camera," *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 1, pp. 5609-5614, 2009.

[4] David G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, Vol. 60, No. 2, pp. 91-110, 2004.

[5] Herbert Bay, Tinne Tuytelaars and Luc Van Gool, "SURF: Speeded Up Robust Features", *Lecture Notes in Computer Science*, Vol. 3951, pp. 404-417, 2006.

[6] Marius Leordeanu, Martial Hebert, and Rahul Sukthankar, "Beyond Local Appearance: Category Recognition from Pairwise Interactions of Simple Features," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2007.

[7] S.S. Ghidary, Y. Nakata, H. Saito, M. Hattori, and T. Takamori, "Multi-modal Iinteraction of Human and Home Robot in the Context of Room Map Generation," *Autonomous Robots*, Vol. 13, No. 2, pp. 169-184, 2002.

[8] A. Haasch, N. Hofemann,J. Fritsch, and G. Sagerer, "A Multi-modal Object Attention System for a Mobile Robot," *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 1, pp. 1499-1504, 2005.

[9] Roger E. Kahn, Michael J. Swain, Peter N. Prokopowicz, and R. James Firby, "Gesture Recognition Using the Perseus Architecture," *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 734-741, 1996.

[10] P. M. Roth, M. Donoser, H. Bischof, "On-line Learning of Unknown Hand Held Objects via Tracking," *Proceedings of Internationl Conference on Computer Vision Systems*, 2006.

[11] Artur M. Arsenio, "Figure/Ground Segregation from Human Cues," *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 4, pp. 3244-3250, 2004.

[12] Thrasyvoulos N. Pappas, "Adaptive Clustering Algorithm for Image Segmentation," *IEEE Transactions on Signal Processing*, Vol. 40, No. 4, pp. 901-914, 1992.

[13] J. B. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations," *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, Vol. 1, pp. 281-296, 1967.

[14] D. Comaniciu and P. Meer., "Mean shift: A Robust Approach Toward Feature Space Analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, pp. 603-619, 2002.

[15] Keinosuke Fukunaga, Larry D. Hostetler, "The Estimation of the Gradient of a Density Function with Applications in Pattern Recognition," *IEEE Transactions on Information Theory*, Vol. 21, No. 1, pp. 32-40, 1975.

[16] Jianbo Shi and Jitendra Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 8, pp. 888-905, 2000.

[17] Yuri Y. Boykov and Marie-Pierre Jolly, "Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in N-D Images," *Proceedings of International Conference on Computer Vision*, Vol. 1, pp. 105-112, 2001.

[18] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake, "GrabCut – Interactive Foreground Extraction using Iterated Graph Cuts," *ACM Transactions on Graphics (SIGGRAPH)*, Vol. 23, pp. 309-314, 2004.

[19] S. Beucher and F. Meyer, "The Morphological Approach to Segmentation: the Watershed Transformation," *Optical Engineering*, Vol. 34, pp. 433, 1992.

[20] F. Meyer, "Color image segmentation," *Proceedings of the International Conference on Image Processing and its Applications*, Vol. 1, pp. 303-306, 1992.

[21] Hyung-O Kim, Soohwan Kim, and Sung-Kee Park, "Pointing Gesture-based Unknown Object Extraction for Learning Objects with Robot," *Proceedings of International Conference on Control, Automation and Systems*, Vol. 1, pp 2156-2161, 2008.

[22] Paul Viola and Michael J. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp. 511-518, 2001.

[23] Rainer Lienhart and Jochen Maydt, "An Extended Set of Haar-like Features for Rapid Object Detection," *Proceedings of IEEE Interantional Conference on Image Processing*, Vol. 1, pp. 900-903, 2002.

[24] George Paschos, "Perceptually Uniform Color Spaces for Color Texture Analysis: An Empirical Vvaluation," *IEEE Transactions on Image Processing*, Vol. 10, pp. 932-936, 2001.