ROBOMECH Journal

RESEARCH ARTICLE

Open Access

Object segmentation in cluttered environment based on gaze tracing and gaze blinking

Photchara Ratsamee[1*], Yasushi Mae[2], Kazuto Kamiyama[3], Mitsuhiro Horade[4], Masaru Kojima[5] and Tatsuo Arai[6]

Abstract

People with disabilities, such as patients with motor paralysis conditions, lack independence and cannot move most parts of their bodies except for their eyes. Supportive robot technology is highly beneficial in supporting these types of patients. We propose a gaze-informed location-based (or gaze-based) object segmentation, which is a core module of successful patient-robot interaction in an object-search task (i.e., a situation when a robot has to search for and deliver a target object to the patient). We have introduced the concepts of gaze tracing (GT) and gaze blinking (GB), which are integrated into our proposed object segmentation technique, to yield the benefit of an accurate visual segmentation of unknown objects in a complex scene. Gaze tracing information can be used as a clue as to where the target object is located in a scene. Then, gaze blinking can be used to confirm the position of the target object. The effectiveness of our proposed method has been demonstrated using a humanoid robot in experiments with different types of highly cluttered scenes. Based on the limited gaze guidance from the user, we achieved an 85% F-score of unknown object segmentation in an unknown environment.

Keywords: Gaze interface, Human–robot interaction, Object segmentation

Introduction

There are many patients who suffer from devastating conditions, such as amyotrophic lateral sclerosis (ALS) [1], brain stroke, and muscular dystrophy [2]. These patients usually retain full consciousness but can only blink or move their eyebrows. As it is assumed that humanoid robots will coexist in human environments in the near future, the ability of a humanoid robot to assist such patients is in high demand. The most common situation in which patients need assistance from a robot is an object search application, where the robot is expected to deliver a specific object in the environment according to the user's needs. Even the most common application of object segmentation is a very challenging problem.

To autonomously segment objects in a cluttered environment, many techniques, such as active contour model or snake [3], level sets [4], and the graph cuts [5] method have been proposed in the computer vision field. In extending the graph cut method, many researchers have tried to use predefined information, such as shape [6], or using kernel methods [7]. Recently, many learning-based approaches [8], in which the robot has previously learned object categories, have been introduced. However, it is still hard to achieve high accuracy from passive observation. Learning-based methods are hard to apply in practical situations, in which new objects are introduced every day. It is not practical to register all new objects in a database.

Interaction with a user makes object segmentation feasible for practical situations. Based on the interactive

*Correspondence: photcyber@gmail.com
[1] Graduate School of Information Science and Technology, Osaka University, Osaka, Japan
Full list of author information is available at the end of the article

capability of a patient, we can use facial engagement, head pose commands, and Brain Machine Interface (BMI) to interact with the robots [9]. Recently, BMI played an important role in helping patients control electronic appliances and the movements of robotic or prosthetic limbs [10]. However, to give a command about an object's position and set boundaries in a cluttered environment is difficult, as transmitting a precise command via these interaction methods is complicated.

To interact with a robot in an object segmentation task, we introduced a Gaze Tracking Device (GTD) to disabled patients. This gaze-interaction task was primarily divided into three steps. First, the patient gazes at the target object using their own vision or visual feedback from the robot's vision and then give a command by blinking to select the target object. Next, the robot navigates to the target object, sending visual feedback to the user's monitor, which allows the user to select and confirm the target object with another gaze. Finally, the robot grasps the object and brings it back to the user (Fig. 1).

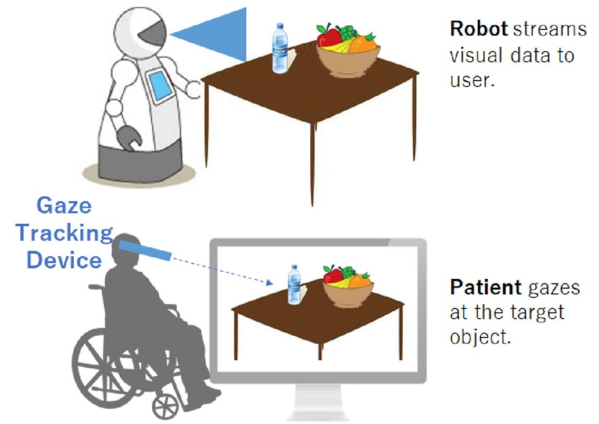
Robot streams visual data to user.
Gaze Tracking Device
Patient gazes at the target object.

Fig. 1 The overall concept of an object search using a Gaze Tracking Device (GTD). The patient gazes at the target object and then the robot autonomously navigates the environment to search for the object while providing visual feedback to the patient

In this paper, we proposed a gaze-based object segmentation method, based on gaze interaction from users, to optimize image labels and segment mixed, multicolored, and occluded objects in a cluttered environment. As the objects in an image are multicolored with noise and low resolution, we investigated a transformation of the original image to a higher dimensional kernel space using an iterative image segmentation as well as proposed a method to filter target object label from the image based on gaze information. Specifically, we proposed two types of gazes interaction for object segmentation:

- Gaze Tracing (GT): in which the user passively gazes into the area surrounding the object in an image.
- Gaze Blinking (GB): in which the user blinks at the center of the object for confirmation.

These two types of gazes can be integrated with visual-based object segmentation to achieve accurate object segmentation in cluttered environment.

System architecture

This section briefly explains the object-search system architecture (Fig. 2) [11], which allows a robot to navigate to the target location according to the user's gaze command. The calibration of the GTD is initially conducted with the method described in [12]. Then, a Kalman filter [13] is applied for gaze position (g_x^p, g_y^p) smoothing. Finally, our proposed gaze-based object segmentation is applied for target object segmentation.

Blink detection

We can also detect a user's eye blinks using [14]. Therefore, voluntary long blinks and involuntary short blinks can be classified. Different types of blinks instruct different robot commands as follows:

- Location command: This command specifies a target's location. This can be done by performing two

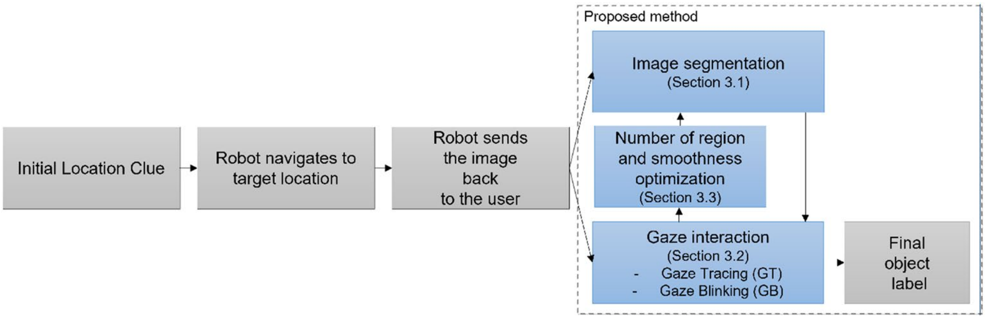
Proposed method
Initial Location Clue
Robot navigates to target location
Robot sends the image back to the user
Image segmentation (Section 3.1)
Number of region and smoothness optimization (Section 3.3)
Gaze interaction (Section 3.2) - Gaze Tracing (GT) - Gaze Blinking (GB)
Final object label

Fig. 2 Flowchart of object search by service robot and our proposed method

consecutive blinks at the target's location in the real environment.

- Object confirmation command: After gaze tracing, we can confirm an object's location by gaze blinking, in which the user performs three consecutive blinks.

Each blink interval must be around 300 ms. This blink pattern is clearly not natural human eye motion and clearly separated from involuntary short blinks, so it does not increase cognitive load and does not disturb the user's natural gaze pattern.

Robot teleoperation using gaze

The user can teleoperate the robot by performing a location command in which the user gazes directly at the goal location in the environment. Our robot platform can independently navigate to that target location based on prior map and indirect search algorithms [15]. Once the robot arrives at the target location, it streams visual data of the target object to the user for gaze interaction. Since the robot has a limited field of view, it can adapt its observation point to provide different perspectives of the object which is based on the approach of the author's previous work [11].

Methods

This section presents our proposed gaze-based object segmentation. Our strategy is to use gaze-interaction to enhance vision-based target-object segmentation. The gaze interaction was designed so that object segmentation can be performed as few gaze interactions as possible. Firstly, the user applies only a few examples of gaze tracing (GT) and gaze blinking (GB) to the target object. Afterward, we apply image segmentation to segment an image into different labels with parameter optimization from obtained gaze information. Finally, we filter target object label based on gaze tracing (GT) and gaze blinking (GB) approach.

Image segmentation

Label creation

The goal is to segment Image I into different K regions $(r_{l=1}, r_{l=2} \ldots, r_{l=K})$ that are smooth and consistent for the user to perform gaze interaction. This problem is a labeling problem, in which we utilize the graph cut method [7] to find a label f that minimizes the energy.

$$E(f) = \sum_{p \in \mathrm{P}} D_p(f) \; + \alpha \sum_{p,q \in N} S_{p,q}(f) \tag{1}$$

where α is a resolution constant and $D_p(f)$ and $S_{p,q}(f)$ are data term and smoothness term, respectively. The data term is defined as:

$$D_p(f) = \sum_{l \in L} \sum_{p \in R_l} (c_l - I_p)^2 \tag{2}$$

where c_l is the piecewise constant model parameter of region R_l. The data term is derived from the observed data that measures the cost of assigning label f_p to pixel p. The smoothness term is defined as:

$$S_{p,q}(f) = \sum_{\{p,q\} \in N} s(f(p), f(q)) \tag{3}$$

where $s(f(p), f(q))$ measures the cost of assigning the labels f_p, f_q to neighbor pixels p, q. We define $s(f(p), f(q))$ as the truncated squared absolute distance as:

$$s(f(p), f(q)) = \min\left(ct^2, \left|c_{f(p)} - c_{f(q)}\right|^2\right) \tag{4}$$

where ct is a constant of the truncated squared absolute distance.

Kernel mapping

A data term $D_p(f)$ of an image data is converted via a kernel function so that the system is suitable to segment nonlinearly separable data. Therefore, Eq. 1 is transformed to:

$$E_K(\{c_l\}, f) = \sum_{l \in L} \sum_{p \in R_l} (\phi(c_l) - \phi(I_p))^2 \; + \alpha \sum_{p,q \in N} S_{p,q}(f) \tag{5}$$

where $\phi(.)$ is a nonlinear mapping from image space to a higher dimensional feature space depending on the number of segmentation regions N_R. Based on a kernel trick [7], we can derive the kernel function as:

$$(\phi(c_l) - \phi(I_p))^2 = K(I_p, I_p) + K(c_l, c_l) - 2K(I_p, c_l) \tag{6}$$

where we use the radial basis function (RBF) kernel, which is suited for pattern data clustering. The RBF kernel is defined as:

$$K(x_1, x_2) = \exp(-\|x_1 - x_2\|^2 / \sigma^2) \tag{7}$$

Optimization

To achieve an optimized image label, Eq. 5 was optimized with an iterative two-step optimization method. The first step is performed by updating the centroid data of each label based on the following condition:

$$c_k - z_{R_k}(c_k) = 0 \tag{8}$$

where

$$z_{R_k}(c_k) = \frac{\sum\limits_{p\in R_k} I_p K\left(I_p, c_k\right)}{\sum\limits_{p,q\in N} K\left(I_p, c_k\right)} \tag{9}$$

The second step consists of finding the optimal label of the image from label centroid data provided by the first step. Each step updates the centroid data and creates a new optimal label. The algorithm iterates these two steps until the energy converges to a local minima.

Target object selection by gaze interaction

Even though the image was segmented into different labels, the target-object label needs to be defined by the user and we aim to use gaze interaction to assist in object selection. There are several existing gazed based user interfaces. Rivu et al. [16] propose to use gaze from the user to gradually reveal information on demand. Also, in Augmented Reality and Virtual Reality, instead of using mouse or gestures, we can confirm targets using gaze selection [17]. Gaze is also used to select objects in 3D environments based on hybrid gaze and controller techniques [18]. Recently, a combination of gaze and gestures is an active field with several applications such as object manipulation [19] or gaze-enhanced menu interfaces [20].

In this work, we focus on gaze-only as a potentially implicit and effortless method for selecting object from cluttered environment. The user can provide the robot with a clue about the target object's boundaries by gaze tracing and blinking at the target object on a computer screen. The gaze interaction process is divided into:

Gaze tracing (GT)

The user is asked to gaze at the target object on the label image (Fig. 3a). This step is defined as a passive gaze. Based on the position of the gaze tracing, we then build the heat map [21] which is an ellipsoid distribution centered at gaze position (g_x^p, g_y^p):

$$H(p) = w_{hm} e^{-\left(\left(x-g_x^p\right)^2+\left(y-g_y^p\right)^2\right)} \tag{10}$$

where w_{hm} is the weight of the heat map and x and y is the position of each pixel in an image. p indicates gaze points index which is arranged in chronological order. A single ellipsoid heat map from each gaze point will be

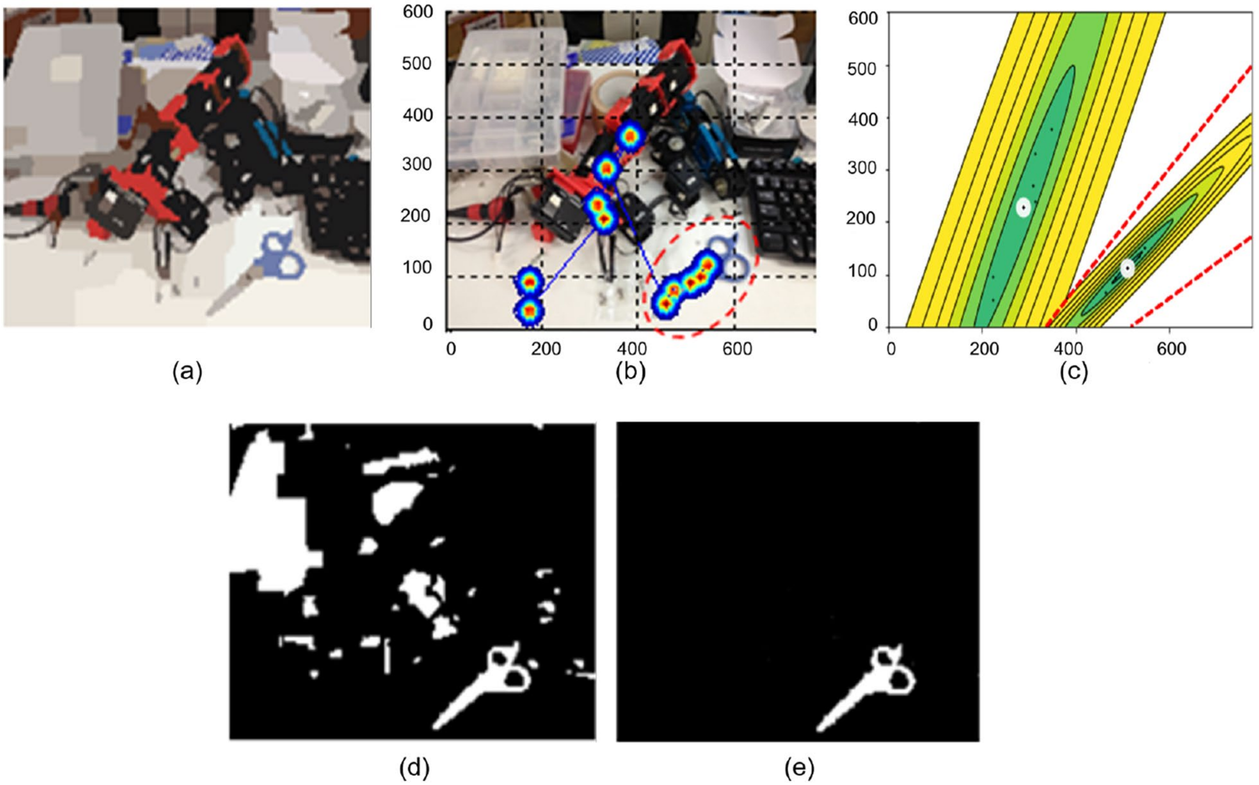

Fig. 3 **a** Label creation from kernel-based image segmentation, **b** heat map created by the user's GT and selected region by GB, **c** Gaussian mixture prediction of GT points where white dot determined the centroid of cluster and red dot line show boundary of cluster, **d** selected label with only GT interaction, **e** selected label with GT and GB interaction

superposed to generate the final heat map, presented in Fig. 3b. This model enables the fixation duration and the number of fixations to be integrated into one map.

For the gaze tracing, it is hard to conclude which labels the user gazed at that should be assembled as object labels, since the user can easily be distracted and look at other spots. As presented in Fig. 3b, we asked the user to look at the scissors, which was the target object. However, the user also unintentionally gazed at other points. Including all gaze-tracing points yields an error in the object-label selection.

We apply a Gaussian Mixture Prediction (GMP) [22] approach to fit Gaussian distribution to GT data as shown in Fig. 3c. The number of clusters has been set to 2 since we want to classify GT points at target location and GT points at non-target location. The co-variance of Gaussian distribution determines the contour of the cluster. GMP uses Expectation Maximization (EM) algorithm [23] to optimize the separation of soft clustering. In our case, we set a non-negative regularization of 10^{-6} added to the diagonal of co-variance and The convergence threshold to stop EM iterations is set to 10^{-3}.

Gaze blinking (GB)

Label selection based only on gaze tracing yields an object label with noise, as presented in Fig. 3d. To confirm the target object, the user is required to give three consecutive blinks at the center of the target object. GB will be used to confirm the cluster that belongs to the target object and we select the label within 2 standard deviations (the size of the ellipsoid) where a mask is centered on the confirmation location (presented as a red ellipsoid in Fig. 3b), which will be created to filter out only the local maximum points of the heat map and used to integrate the corresponding label to the object label as presented in Fig. 3e. The instances that are out of the ellipsoid will be considered outliers.

Smoothness and number of region optimization

We use gaze pattern information to choose the smoothing parameter (α) and a number of segmented region (K). From our observation, we found that the user tends to have more gazing points (GT) when they select complex or multicolored objects while having low number of gazing points on a simple object. As a result, we proposed to choose a number of segmented region parameter K as follow

$$K = w_K \left(\frac{P}{W_o H_o} \right) \tag{11}$$

where P is number of the local maximum points of the heat map from gaze tracing (GT) lies within ellipsoid defined by GB. W_o and H_o are lengths of major and minor axes of ellipsoid derived from GT and GB. w_K is the weight of the number of region adaptation. Furthermore, we update α which is the smoothing term of region by

$$\alpha = 1 - w_a \left(\frac{P}{W_o H_o} \right) \tag{12}$$

where w_a is the weight of the alpha adaptation. From observation, we found w_K to be 1.2 and w_a is set to 1 for best performance on our dataset and TOSD dataset. With K and α optimization, we can achieve high accuracy and low recall rate of object segmentation while reducing the segmentation time of a simple object.

Experiments and results

Experiment setup

We developed and tested the system on a humanoid robot called ENON [24]. The robot was equipped with two RGB-D sensors (Kinect V1 sensor), one for navigation and another on its head at a height of 1.8 m to stream visual data to the user. Users were asked to wear GTDs and accelerometers (to measure the user's head orientation), as presented in Fig. 4.

Our experiments were conducted in an office environment, as presented in Fig. 4. We asked five users which are master students from Osaka University, Japan. Their age ranged from 23 to 31 years (M = 27.20, SD = 3.35), 3 participants were male and 2 were female. All users have no prior experience using GTDs. Users use our system to guide the robot to a target location and perform gaze interaction with objects on a table.

Each user individually performed a calibration exercise to confirm gaze precision. The subject was required to look at different reference points from different ranges. Overall, an acceptable average error was found to be 1.51° with variance of 0.77°. The robot sent a visual stream with an image size of 640*480 pixels, which was then processed on MATLAB running on Windows 10 on a personal computer (PC) (E5-1620 3.50 GHz Xeon CPU, 16384 MB RAM, NVIDIA Quadro K2200 graphics card). The processing time varied depending on the number of iterations, segmentation regions, and resolution sets for kernel graph cuts segmentation.

Ten objects of different shape, size, and appearance were prepared. The experiment was conducted 10 times for each object, modifying the object's appearance from being strongly occluded with textured sides, to sparsely or partially textured, non-textured, multicolored, or with unicolored sides.

Evaluation

We evaluated our proposed segmentation algorithm on two datasets: our dataset and TOSD datasets [25]. We selected TOSD as a comparison dataset since it consists

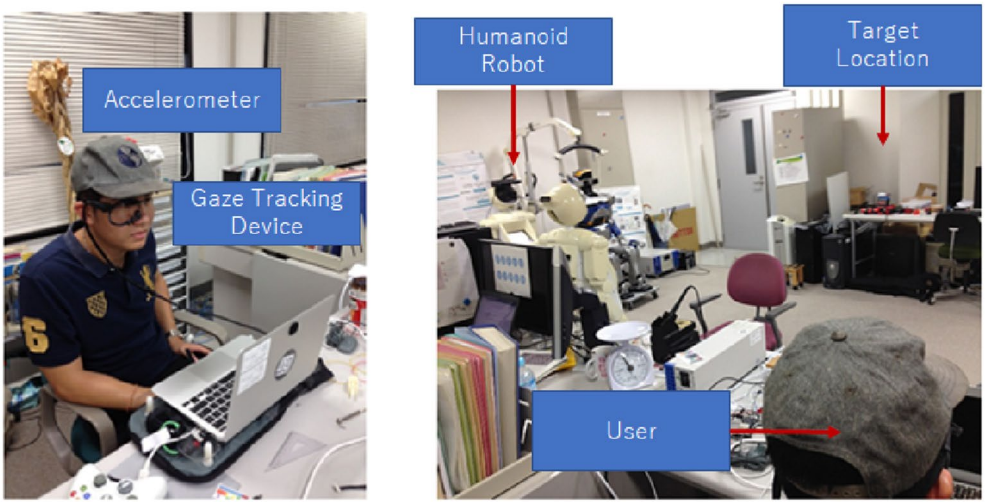

Fig. 4 The experiment setup

of scenes with varied object-configuration complexities. It is composed of images with complex and cluttered scenes, as well as scenes where only several boxes or other simple objects are presented. The TOSD dataset consists of 111 scenes for training and 131 scenes for testing.

For comparison, we also compare our object-segmentation method with the active [26] and the saliency segmentation [25] methods for GT, GB, as well as the combination of GT and GB for segmentation. The quality of segmentation was measured based on the recall and precision of segmentation, i.e., how many points in the final object label corresponded to the ground truth (the object being manually selected by the user). We compared by using the F-measure defined by

$$F = \frac{2PR}{(P+R)} \tag{13}$$

where calculated precision P is the fraction of the segmentation mask overlapping with the ground truth and recall R is the fraction of the ground truth overlapping with segmentation mask.

Result

Object label segmentation analysis

As a piecewise constant model, the system starts by segmenting an image into different regions based on K-mean clustering. The result is presented in Fig. 5, in which the initial label image is not smooth and consistent. The target object selected using the gaze interaction from this initial label will still result in an object label with noise from the areas both inside and outside the label. For example, consider an initial object label calculated from K-mean clustering in which the number of labels is set to four. The initial label is converted via RBF kernel function to a higher dimension of four different images. As a result, each label characteristic is represented in each kernel image.

Next, each kernel image is applied to an iterative graph cut algorithm. The algorithm interactively merges the small noise inside the label while preserving the minimal

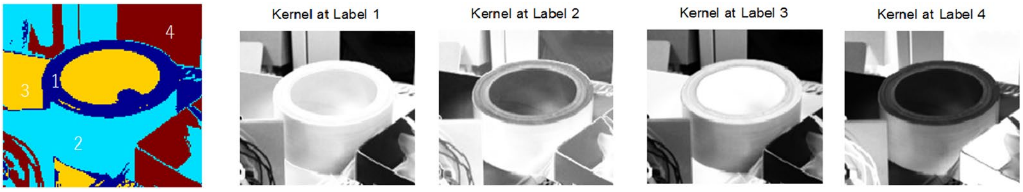

Fig. 5 Left: the example of the initial object label is based on K-mean clustering, in which the number of labels is set to 4. The initial label is not smooth and consistent, so it is converted via RBF kernel function to a higher dimension of four different images in which the distribution of each kernel is changed based on each label characteristic

energy of each label. The advantage of each kernel image is that it is resilient to noise from Gaussian assumption of the RBF kernel function. From our observation, the algorithm converges within five iterations, as presented in Fig. 6. With the label image, only one gaze point at each label is sufficient to include that label as part of the object label.

Gaze-based object segmentation result

Based on an optimized label obtained from the iterative graph cut, the user can select the region by GT point and confirm it with GB. Example results are presented in Fig. 7, in which the second and third columns show the optimized boundaries and labels, respectively. The user viewed the object label image and performed GT and GB interaction. The result of gaze tracing is presented in the fourth column of Fig. 7, in which the target label is only gazed at, but noise remains. The final object, in which the noise is removed by GB, is presented in the fifth column of Fig. 7. The result confirms that this system is applicable for the object segmentation of multicolored objects (as shown in the first row) and occluded objects (as shown in the second and third rows).

Figure 8 shows a comparison of the proposed segmentation performance with an active segmentation approach. For GB only, active segmentation is performed at an average of 38.7%, since the algorithm only works when the object has a linear color distribution. However, objects 1, 2, and 3 were multicolored and had noise, so the active segmentation algorithm performance dropped to an average of 19%. Furthermore, the kernel-based method handled noise and nonlinear data more robustly, with an average precision of 45%.

By integrating GT and GB to the object-segmentation algorithm, as presented in Fig. 8, the robot achieved better segmentation, and the performances of the kernel-based and active segmentation approaches improved to 86.9% and 80.3% precision, respectively. This was due to the GT and GB clues from the user, helping the system integrate multicolor labels into the same object.

Interaction analysis

Since each interaction possibly leads to a different segmentation for an object, we also analyzed the results of four different F-scores. *First* refers to the segmentation

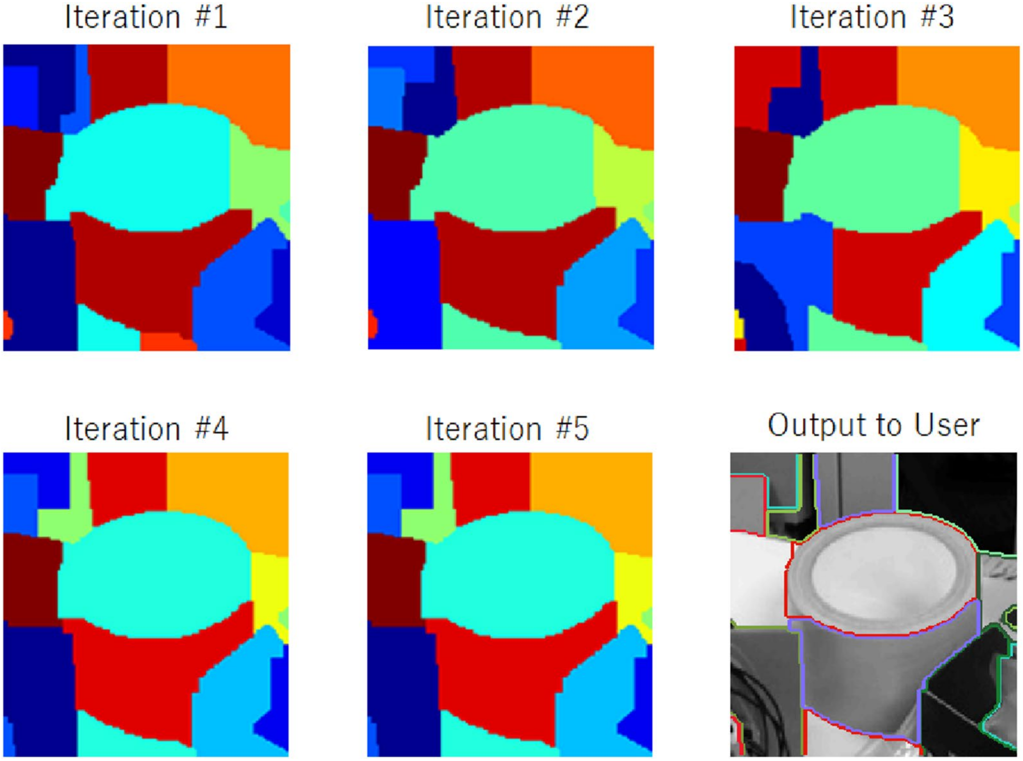

Fig. 6 The result of energy optimization of iterative kernel segmentation. At each iteration, the algorithm updates the kernel image and output label. Therefore, the final label at the last iteration has a smooth and consistent label with a minimum energy guarantee. This optimizes the label (where all the holes are filled), allowing the user to easily interact with the label

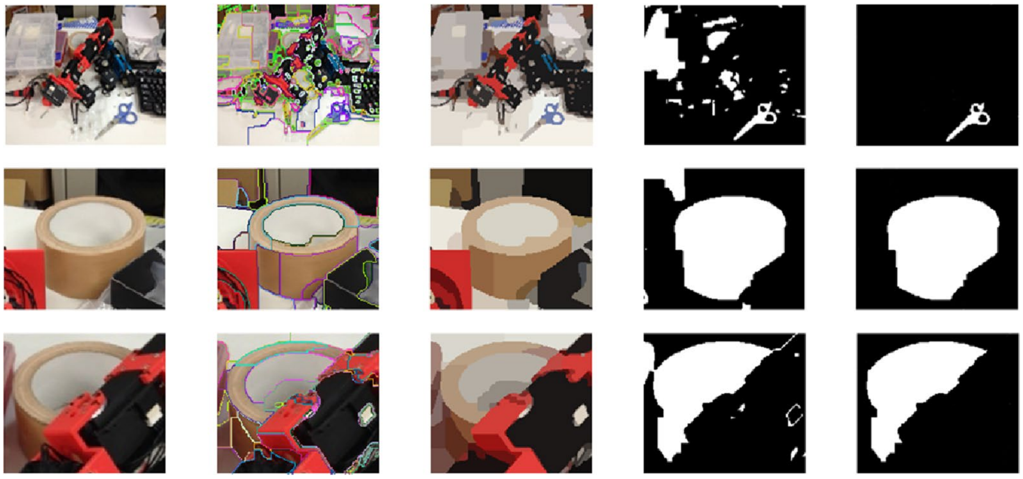

Fig. 7 Original image (first row), boundary of each label overlay on the original image (second row), example of object segmentation (third row), image label result from image segmentation (fourth row), object segmentation result from only gaze-tracing (GT) (fifth row), and the final object label from GT and GB

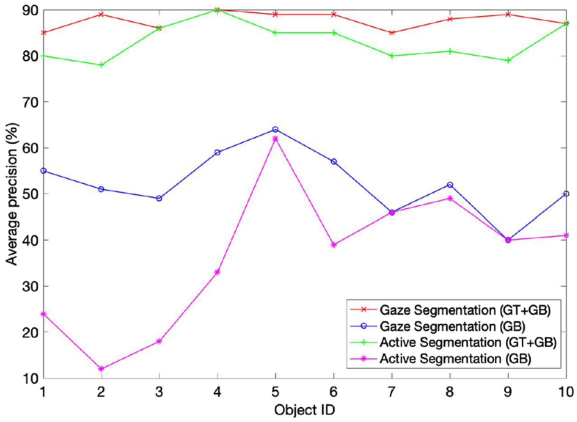

Fig. 8 The comparison of average precision of each object segmentation based on the proposed gaze-based method

from the first interaction, *Best* refers to the best segmentation WRT ground truth, *All* refers to the average score over all the segmentations for an object, and *Worst* refers to the worst score among all the segmentations for an object.

Ideally, we would like each interaction to yield the best segmentation. However, the segmentation algorithm depends a lot on the position of the GT and GB points. Therefore, an algorithm that is resilient to different interactions (i.e., different sets of GT and GB points) is expected. To show how GT and GB interaction can improve segmentation to other traditional approaches (Active and Saliency), we also implemented 3 types of interaction which are GT-only, GB-only, and GT + GB to conventional approaches. For GB-only interaction, without GT points, GMP will use the centroid of label that has a similar color (±5 of hue value) to be points for clustering. For example, if the GB point is located at the green label, all the centroids of the green labels will be used in the GMP process. If the target object has a single color or similar color, the GB-only can assist segmentation well. However, GB-only interaction is not robust to multi-color object segmentation. For GT-only interaction, since there is no object confirmation from the GB point, the system will select the cluster that has more GT points as a target object. This approach can fail when the user wants to select a tiny object since the GT points at a small object usually less than the distraction point.

We present all F-scores in Table 1 for all cases (All, First, Worst, and Best), for all objects, and all scenes. GT + GB interaction generally improve overall performance in all approaches and on average, our proposed gaze-based object segmentation outperformed Active segmentation, Saliency segmentation and Kernel segmentation by 24%, 15% and 10% respectively.

Comparison with state of the art methods

We also evaluated our segmentation algorithm with other state-of-the-art object segmentation approaches using publicly available datasets: the Table Object Scene Dataset (TOSD) [25]. Table 2 compares our results to state-of-the-art object segmentation.

With GT and GB interaction, our method achieves an F-score of 0.75, which is an over 10 % relative improvement from the previous best entry (SGN [27]) and is also

Table 1 F-score for different segmentation algorithms evaluated on our datasets

	Interaction	All		First		Worst		Best	
		Mean	SD	Mean	SD	Mean	SD	Mean	SD
Active segmentation [26]	GT only	0.41	0.02	0.21	0.02	0.18	0.02	0.75	0.02
	GB only	0.52	0.03	0.20	0.03	0.22	0.01	0.71	0.01
	GT+GB	0.61	0.04	0.42	0.02	0.33	0.01	0.83	0.01
Saliency segmentation [25]	GT only	0.50	0.03	0.46	0.02	0.34	0.02	0.79	0.02
	GB only	0.42	0.06	0.31	0.03	0.33	0.02	0.74	0.02
	GT+GB	0.71	0.02	0.55	0.04	0.40	0.01	0.86	0.03
Kernel segmentation [7]	GT only	0.50	0.07	0.36	0.03	0.21	0.04	0.88	0.03
	GB only	0.40	0.06	0.33	0.02	0.22	0.03	0.70	0.02
	GT+GB	0.75	0.03	0.61	0.02	0.32	0.04	0.81	0.02
Gaze segmentation	GT only	0.54	0.09	0.43	0.03	0.31	0.02	0.87	0.02
	GB only	0.44	0.02	0.34	0.02	0.34	0.01	0.72	0.04
	GT+GB	0.85	0.03	0.66	0.02	0.41	0.02	0.91	0.02

Table 2 F-score for different segmentation algorithms evaluated on TOSD [25] datasets

Method	All	Glass	Bottle	Plate	Book	Mug
Mask RCNN [28]	0.68	0.72	0.62	0.61	0.68	0.81
DWT [29]	0.74	0.63	0.82	0.72	0.75	0.82
InstanceCut [30]	0.69	0.52	0.84	0.62	0.64	0.85
SGN [31]	0.65	0.66	0.72	0.70	0.67	0.52
Kernel segmentation [7]	0.70	0.62	0.74	0.64	0.65	0.86
Gaze segmentation	0.75	0.72	0.75	0.68	0.68	0.87

better than the concurrent work from MASK R-CNN [28] with 7%. Compared to the best entry, using fine data only, we achieve 15% improvement. We also performed evaluation within each individual category. Our method shows massive improvement of each category over other approaches (relatively 15% improvement over Glass and Bottle and 20% improvement on Plate, Book and Mug).

Discussion

As only color information from a monocular camera was used, we analyzed and discussed the characteristics of object segmentation and how smoothness and a number of regions affect the accuracy of segmentation. Typically, the image segmentation algorithm [7] was robust enough to segment image into fine piece of image label and it can filter out noise in higher kernel space. However, as it is typical unsupervised learning, some parameter such as the number of regions and the smoothness must be predefined. A single-color object and multi-color required different sets of parameters to achieve high accuracy. In our study, we not only use GT and GB to directly segment target object label, but we also proposed to adapt smoothness and a number of regions based on gaze clues from the user which is another factor to achieve high accuracy (Fig. 9).

For a single-color object, the user usually performs only a few gaze tracing. If the label segmentation is too fine, the same color label can be separated into several labels. There is a chance some target object labels might not be selected by gaze tracing. As a result, the low number of the region and a high degree of smoothness should be set so that the object label is not too fine. On the other hand, the user usually performs more gaze tracing points on different regions of a multi-color object. To achieve high precision of multicolor objects, a high number of regions and a low degree of smoothness parameter should be set. As a result, object labels are more distinct from other object labels and can be matched with gaze tracing points.

Conclusion

There are many patients who suffer from locked-in syndrome and an inability to live independently. We proposed object segmentation based on gaze interaction for patients to interact with a robot for the application of an object search in a cluttered environment.

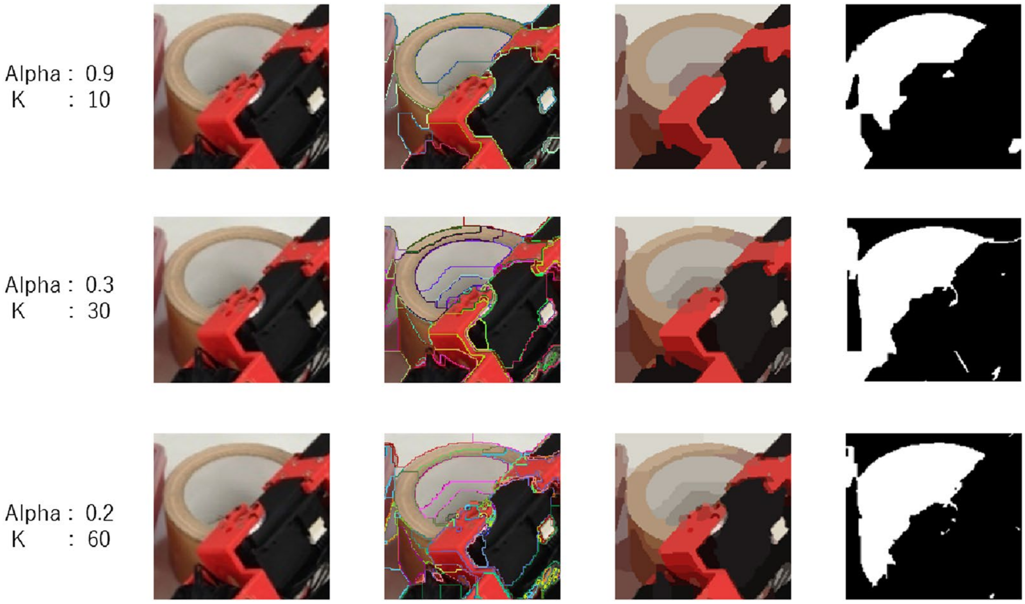

Fig. 9 The difference of segmented labels related to the number of regions (K) and smoothness constant (α)

To interact with the label in an image, we introduced the concepts of GT and GB to help the robot determine its target object. The patient can perform GT by freely gazing at the area around a target object. Later, the user provides the robot with the object location from GB, which involves three consecutive blinks. Afterwards, the kernel-based segmentation algorithm with parameter selection from gaze information was performed with the purpose of image labeling. The result of this interaction will be integrated with the image labeling to confirm the final object label.

Our experiment results show that the proposed gaze-based method overcomes the conventional method (with an F-score of 85% for a combination of GB and GT) for noisy multicolor and occluded object segmentation with an average precision of 54.8% for GT and 86.9% for a combination of GB and GT, respectively.

Our future work will focus on integrating this system with autonomous navigation for autonomous wheelchairs.

Acknowledgements
Thank you to all the users who joined the experiment and provided valuable comments on this research. We would also like to offer special thanks to Christian Penaloza and Krishneel Chaudhary for their valuable comments on this research.

Authors' contributions
Ratsamee devised the system's basic concept, technically constructed the system, and conducted the research and experiments. Mae led the research progress, assisted with the implementation, secured funding for the research, and revised and refined the manuscript. Arai assisted the research and revised the manuscript. All authors read and approved the final manuscript.

Funding
This research was supported in part by the Mohamed Bin Zayed International Robotics Challenge (MBZIRC) Grant. This work was also supported in part by JSPS KAKENHI Grant Number JP17H01801 and the Japanese Government's Monbukagakusho (MEXT) Scholarship.

Availability of data and materials
The datasets generated during the current study are available in the NextCloud repository https://bit.ly/33FldHN.

Declarations

Ethics approval and consent to participate
In subject experiments, the ethics committee for safety subjects it to an ethical review established by Cyber Media Center at Osaka University. From the Ethics Committee, we obtained approval for data use in this subject. Appropriate care was taken to avoid any psychological or physiological distress on the subject during the experiment.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Author details
[1] Graduate School of Information Science and Technology, Osaka University, Osaka, Japan. [2] Graduate School of Engineering, Kansai University, Osaka, Japan. [3] Takenaka Research & Development Institute, Takenaka Corporation, Tokyo, Japan. [4] Department of Mechanical Systems Engineering, School of Systems Engineering, National Defense Academy, Tokyo, Japan. [5] Graduate School of Engineering Science, Osaka University, Osaka, Japan. [6] The University of Electro-Communications, Tokyo, Japan.

Received: 27 July 2021 Accepted: 2 December 2021
Published online: 22 December 2021

References

1. Eisen A, Kim S, Pant B (1992) Amyotrophic lateral sclerosis (ALS): a phylogenetic disease of the corticomotoneuron? Muscle nerve 15(2):219–224
2. Penaloza C, Mae Y, Ohara K, Arai T (2013) BMI-based learning system for appliance control automation. In: 2013 IEEE international conference on robotics and automation (ICRA). IEEE, pp 3396–3402
3. Bresson X, Esedoḡlu S, Vandergheynst P, Thiran J-P, Osher S (2007) Fast global minimization of the active contour/snake model. J Math Imaging Vis 28(2):151–167
4. Cremers D, Rousson M, Deriche R (2007) A review of statistical approaches to level set segmentation: integrating color, texture, motion and shape. Int J Comput Vis 72(2):195–215
5. Boykov Y, Veksler O, Zabih R (2001) Fast approximate energy minimization via graph cuts. IEEE Trans Pattern Anal Mach Intell 23(11):1222–1239
6. Toshev A, Taskar B, Daniilidis K (2010) Object detection via boundary structure segmentation. In: 2010 IEEE conference on computer vision and pattern recognition (CVPR). IEEE, pp 950–957
7. Salah MB, Mitiche A, Ayed IB (2011) Multiregion image segmentation by parametric kernel graph cuts. IEEE Trans Image Process 20(2):545–557
8. Horbert E, García GM, Frintrop S, Leibe B (2015) Sequence-level object candidates based on saliency for generic object recognition on mobile systems. In: 2015 IEEE international conference on robotics and automation (ICRA). IEEE, pp 127–134
9. Pourmehr S, Monajjemi VM, Vaughan R, Mori G (2013) "You two! take off!": creating, modifying and commanding groups of robots using face engagement and indirect speech in voice commands. In: 2013 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, pp 137–142
10. Hochberg LR, Bacher D, Jarosiewicz B, Masse NY, Simeral JD, Vogel J, Haddadin S, Liu J, Cash SS, van der Smagt P et al (2012) Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. Nature 485(7398):372–375
11. Ratsamee P, Mae Y, Kamiyama K, Horade M, Kojima M, Kiyokawa K, Mashita T, Kuroda Y, Takemura H, Arai T (2015) Object search framework based on gaze interaction. In: 2015 IEEE international conference on robotics and biomimetics (ROBIO). IEEE, pp 1997–2002
12. Li D, Babcock J, Parkhurst DJ (2006) Openeyes: a low-cost head-mounted eye-tracking solution. In: Proceedings of the 2006 symposium on eye tracking research & applications. ACM, pp 95–100
13. Bar-Shalom Y, Li XR (1995) Multitarget-multisensor tracking: principles and techniques. University of Connecticut, Storrs
14. Grauman K, Betke M, Gips J, Bradski GR (2001) Communication via eye blinks-detection and duration analysis in real time. In: Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001, vol. 1. IEEE, p 1010
15. Chumtong P, Mae Y, Ohara K, Takubo T, Arai T (2014) Object search using object co-occurrence relations derived from web content mining. Intell Serv Robot 7(1):1–13
16. Rivu R, Abdrabou Y, Pfeuffer K, Esteves A, Meitner S, Alt F (2020) Stare: gaze-assisted face-to-face communication in augmented reality. In: ACM symposium on eye tracking research and applications. ETRA '20 Adjunct. Association for computing machinery, New York. https://doi.org/10.1145/3379157.3388930
17. Pfeuffer K, Abdrabou Y, Esteves A, Rivu R, Abdelrahman Y, Meitner S, Saadi A, Alt F (2021) Artention: a design space for gaze-adaptive user interfaces in augmented reality. Comput Gr 95:1–12. https://doi.org/10.1016/j.cag.2021.01.001
18. Piumsomboon T, Lee G, Lindeman RW, Billinghurst M (2017) Exploring natural eye-gaze-based interaction for immersive virtual reality. In: 2017 IEEE symposium on 3D user interfaces (3DUI). IEEE, pp 36–39
19. Pfeuffer K, Mayer B, Mardanbegi D, Gellersen H (2017) Gaze+ pinch interaction in virtual reality. In: Proceedings of the 5th symposium on spatial user interaction, pp 99–108
20. Pfeuffer K, Mecke L, Delgado Rodriguez S, Hassib M, Maier H, Alt F (2020) Empirical evaluation of gaze-enhanced menus in virtual reality. In: 26th ACM symposium on virtual reality software and technology, pp 1–11
21. Cerf M, Harel J, Einhäuser W, Koch C (2008) Predicting human gaze using low-level saliency combined with face detection. Advances in neural information processing systems. MIT, Cambridge, pp 241–248
22. Géron A (2019) Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: concepts, tools, and techniques to build intelligent systems. O'Reilly Media, Beijing
23. Moon TK (1996) The expectation–maximization algorithm. IEEE Signal Process Mag 13(6):47–60
24. Kanda S, Murase Y, Sawasaki N, Asada T (2006) Development of the service robot "enon." J Robot Soc Jpn 24(3):12
25. Potapova E, Varadarajan KM, Richtsfeld A, Zillich M, Vincze M (2014) Attention-driven object detection and segmentation of cluttered table scenes using 2.5 d symmetry. In: 2014 IEEE international conference on robotics and automation (ICRA). IEEE, pp 4946–4952
26. Mishra A, Aloimonos Y, Fah CL (2009) Active segmentation with fixation. In: 2009 IEEE 12th international conference on computer vision. IEEE, pp 468–475
27. Liu T, Chen Z, Wang X (2019) Automatic instructional pointing gesture recognition by machine learning in the intelligent learning environment. In: Proceedings of the 2019 4th international conference on distance education and learning, pp 153–157
28. He K, Gkioxari G, Dollár P, Girshick R (2017) Mask r-cnn. In: 2017 IEEE international conference on computer vision (ICCV). IEEE, pp 2980–2988
29. Bai M, Urtasun R (2017) Deep watershed transform for instance segmentation. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR). IEEE, pp 2858–2866
30. Kirillov A, Levinkov E, Andres B, Savchynskyy B, Rother C (2017) Instancecut: from edges to instances with multicut. In: CVPR, vol. 3, p 9
31. Liu S, Jia J, Fidler S, Urtasun R (2017) Sgn: sequential grouping networks for instance segmentation. In: The IEEE international conference on computer vision (ICCV)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.