Region Extraction of a Gaze Object using the Gaze Point and View Image Sequences

Norimichi Ukita, Tomohisa Ono, and Masatsugu Kidode
Graduate School of Information Science
Nara Institute of Science and Technology
ukita@is.naist.jp

ABSTRACT

Analysis of the human gaze is a basic way to investigate human attention. Similarly, the view image of a human being includes the visual information of what he/she pays attention to. This paper proposes an interface system for extracting the region of an object viewed by a human from a view image sequence by analyzing the history of gaze points. All the gaze points, each of which is recorded as a 2D point in a view image, are transfered to an image in which the object region is extracted. These points are then divided into several groups based on their colors and positions. The gaze points in each group compose an initial region. After all the regions are extended, outlier regions are removed by comparing the colors and optical flows in the extended regions. All the remaining regions are merged into one in order to compose a gaze region.

Categories and Subject Descriptors

I.4.6 [**Image Processing and Computer Vision**]: Segmentation—*Region growing, partitioning*

General Terms

Algorithms

Keywords

Region extraction, Gaze points, View image sequence, gaze object

1. INTRODUCTION

If a computer system recognizes which object a person is interested in, the system can give useful information about that object to him/her. In particular, such a system is very convenient for an online information service if it can work wherever he/she is, even while moving around. Several researches have proposed online interface systems that give useful information to a system user. For example, [1] developed a system that provides information about objects (e.g., buildings) that are in front of a user. In this system, (1) the information is superimposed on a view image taken by a camera set near his/her eyes and (2) this image is displayed in a head-mounted display. This system, however, needs a gyro sensor, a GPS receiver, a pedometer, and other kinds of geometric sensors in order to identify objects in front of a user. That is, the given information is determined only by geometric information. In fact, this system[1] is designed for providing information about large buildings/areas in a wide area. Similar systems have been proposed in [2, 3]. Although this scheme is effective for identifying large objects such as buildings, (i) annotated information becomes complicated in an image if there are many objects in front of a user, (ii) it is difficult to identify an object that a user is focusing on, and (iii) it is impossible to give the information of mobile objects such as people, robots, and livingware. To solve these problems, a view image should be analyzed to estimate which object in the view image is being gazed at by a user. The recognition performance can be increased by clearly extracting the region of the object and recognizing the extracted region.

A human can inform a computer system of his/her attention in several ways, e.g., by finger pointing, directing the gaze, and so on. Many systems that recognize where a human points his/her finger at have been proposed (see [4], for example). However, pointing with the finger has the following problems: (1) even if the system can obtain the exact 3D position of a user's fingertip, it is difficult to determine the 3D direction of the finger pointing because the beginning point of the direction is ambiguous or unknown[1] and (2) a user has to stop his/her task while performing a gesture. These problems can be avoided by employing gaze directions: (1) the gaze direction can be determined without ambiguity and (2) while a user performs another task, he/she can gaze at a 3D position. In addition, a human naturally gazes at an object that he/she is interested in. Accordingly, our objective is to acquire the image region of an arbitrary object gazed at by a user, with analysis of the gaze and view image sequences.

We will refer to the previous researches that estimate a gaze region in an observed view image by employing a device for acquiring gaze directions. In [8], the system for

[1] As a result of extensive experiments, we found that the pointing direction is determined not only by a fingertip and a left/right eye but also any other points in the finger. This direction differs in individuals and targets.

ICMI'05, October 4–6, 2005, Trento, Italy.
Copyright 2005 ACM 1-59593-028-0/05/0010 ...$5.00.

extracting a gaze region from an observed image has been reported. This system, however, reconstructs 3D depth information of the region around a gaze point and regards a continuous region as a gaze object. [7] has proposed the active vision system that looks toward an object gazed at by a user. Since the system has no assumption regarding the shape of the target object, it can gaze at an arbitrary object. However, region extraction is not implemented. Since these methods[8, 7] do not deal with image information, it is difficult to extract the region of an arbitrary object. To apply the extracted image of a gaze object to other vision systems that understand what and where the object is (e.g., object recognition and tracking), the image should be extracted completely and precisely.

Object extraction is one of the most important subjects in Pattern Recognition. The regions of all observed objects are segmented based on image information, e.g., edge lines, color distribution, and so on. If all object regions can be extracted using an object extraction method, a region gazed at by a user can be selected by pinpointing it with a device for acquiring gaze directions. Note that the method for our objective must be able to work in a mobile camera system. Although it is more complex than the method for a fixed camera system, for example), some algorithms have been reported. These algorithms, however, cannot be applied in our case due to the restrictions below:

- Simple object detection[9]: This algorithm detects only where objects are in an observed image but cannot extract their regions.

- Object extraction in the polar coordinate system[10]: If the motion of an object is similar to that of a camera, the extraction performance is quite low. In general, a person moves his/her view towards the gaze object. For our purpose, therefore, the disadvantage of this algorithm is critical.

- Background subtraction with a mobile camera[11]: This method first generates a wide background image by mosaicing multiple images observed while moving a camera. However, (1) there must be a long distance between the camera and the observed object and (2) motionless or slow-motion objects cannot be extracted because their regions are included in the mosaiced background image.

For daily use, these restrictions regarding distance, motion and other characteristics of gaze objects should be removed. We believe that this disadvantage of image analysis can be solved by integrating with the above mentioned gaze analysis.

In [12], image and gaze analyses have been proposed for extracting a gaze region. Although this method can extract the region of a gaze object, it can extract only a static object with a fixed camera system (i.e., only when the user's head is not moving). In this paper, therefore, we focus on how to integrate image and gaze analyses for arbitrary object extraction in a mobile camera system. It should be noted that we have to employ the characteristics of temporal/sequential gaze points effectively in order to extract a gaze region precisely. This is an essential difference between our problem and traditional region segmentation in Pattern Recognition.

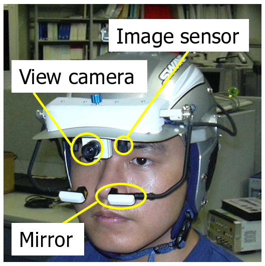


Figure 1: Eye-mark recorder EMR-8.

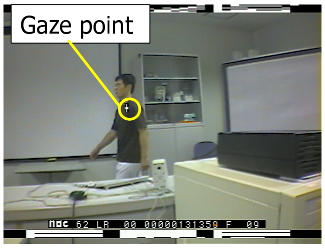


Figure 2: Output image of EMR-8.

2. SYSTEM OVERVIEW

To measure a user's view line, an eye-mark recorder is often used. We use the EMR-8 (Fig. 1) made by *NAC Inc.*[2] The EMR-8 consists of a view camera, an image sensors and a mirror. One of the user's eye is observed by the image sensor through the mirror, and its direction (we call it a *view direction*) is measured. The view camera is almost exactly between two eyes and observes the user's view images in real time. The view direction at each observation timing is represented as a 2D point on the image observed by the view camera and overlaid on it (Fig. 2). We call this point a *gaze point*. The above data is acquired at 15 fps.

The view camera captures sequential images from when a user starts gazing at an object to when he/she stops. In our system, we assume that a user intentionally moves his/her view directions so that (1) all gaze points exist within the gaze region wherever possible and (2) the region is covered by the gaze points as much as possible. In addition, he/she notifies the system when he/she starts and finishes gazing at an object in any way (e.g., pressing a button). During the gazing period, a 2D gaze point of one eye in each observed image is obtained. Let $f_0, \cdots, f_{Ex}, \cdots, f_N$ be the observed images taken at intervals. After the gazing finishes, the system extracts the gaze region. This extraction process is divided into two steps. First, all the gaze points are transfered to one of the images f_{Ex} in which the gaze region is extracted (this will be described in Sec. 3). Second, the gaze region is extracted in f_{Ex} by analyzing image information and the distribution of the transfered gaze points (this will be described in Sec. 4).

The outline of the extraction process is described as follows (Fig. 3):

1. Obtain the history of gaze points

2. Transfer all the gaze points to f_{Ex} (Sec. 3.1)

3. Remove obvious outlier points: (Sec. 3.2)

4. Generate initial regions (Sec. 4.1)

5. Detect edge lines (Sec. 4.2)

6. Extend the regions (Sec. 4.3)

7. Remove the outlier regions (Sec. 4.4)

[2] In our experiments, the EMR-8 was used. For a system which does not use a wearable device, a camera system for estimating gaze directions (e.g., [5, 6]) can be employed.

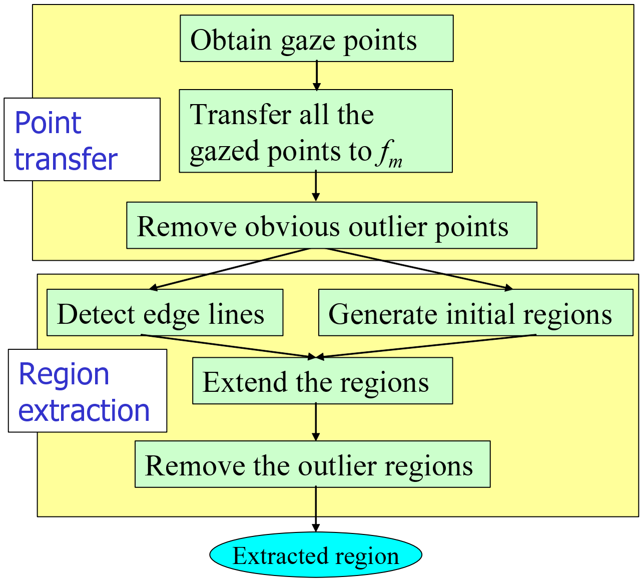


Figure 3: Process flow of the extraction process

3. TRANSFER OF GAZE POINTS TO AN IMAGE

3.1 Transfer of Gaze Points

In our prototype system, a user manually selects an image frame f_{Ex} in which the region of a gaze object is extracted. Gaze points in all observed images are then transfered to f_{Ex}. For this transfer, we integrate the SSDA[13] and the feature tracking and verification method proposed in [14]:

1. A gaze point in frame f_i is transfered to f_{i+1} (if $i < Ex$) or f_{i-1} (if $Ex < i$) towards f_{Ex} frame by frame. With the SSDA, each point is transfered to the point having the least discrepancy calculated by Eq. (1).

$$e(m,n) = \sum_j \sum_i |S(i,j) - W(i-m, j-n)|\,, \qquad (1)$$

   where (m,n) denotes the displacement of a point between two sequential frames and $S(i,j)$ and $W(i,j)$ denote the pixel color at (i,j) in frames f_i and f_{i+1}/f_{i-1}, respectively.

2. The verification method validates whether or not each gaze point successfully corresponds to a point in f_{Ex}.

In general feature point tracking/correspondence problems (e.g., for stereo vision and other 3D reconstruction algorithms), discriminative feature points are first detected and then tracked/corresponded. In our system, on the other hand, since tracking points are given as gaze points, they may have no discriminative features around them (e.g., points in a plain-color area). Such featureless points are mistakenly corresponded to a similar point. For example, a point in a plain-color area can be corresponded to other points in the same area. To cope with this problem, we give weighted variables w_x and w_y to Eq. (1) in order to keep the geometric configuration between two sequential gaze points.

$$\begin{aligned} w_x &= 1 + (x_{p'}^{org} - x_p^{org}) - (x_{p'}^{trn} - x_p^{trn}) && (2) \\ w_y &= 1 + (y_{p'}^{org} - y_p^{org}) - (y_{p'}^{trn} - y_p^{trn}) && (3) \\ & \quad p' = p + 1 \quad \text{if} \quad p < Ex && \\ & \quad p' = p - 1 \quad \text{if} \quad p > Ex && \\ e(m,n) &= \sum_j \sum_i |S(i,j) - W(i-m, j-n)| \cdot w_x w_y && (4) \end{aligned}$$

where (x_p^{org}, y_p^{org}) denotes the coordinate of a gaze point P_p in the original frame f_p and (x_p^{trn}, y_p^{trn}) denotes the coordinate of the same point transfered from f_p to a certain frame. In Eq. (2) and (3), $(x_{p'}^{trn}, y_{p'}^{trn})$ and (x_p^{trn}, y_p^{trn}) are the coordinates transfered to the same frame. Since gaze points are obtained at very short intervals, the weighted variables w_x and w_y work well even if the posture of a gaze object changes while a user gazes at it. It should be noted that any other algorithm for tracking feature points (e.g., [15]) can be employed instead of the SSDA[13] if the algorithm can incorporate the criteria about the distance between consecutive gaze points (i.e., Eq (2)(3)).

For our system, point transfer by the above mentioned process does not cause a serious problem even if it establishes an incorrect correspondence between similar points in a featureless area. This is because both of the original and transfered points are in the same area that should be entirely included in the region of a gaze object, except the case that this area is next to a background region and they have the same colors. Therefore, the transfered point can be used for the following process, namely initial region generation.

In what follows, a gaze point and (x_i, y_i) mean a gaze point transfered to frame f_{Ex} and its coordinate in f_{Ex}.

3.2 Removing Obvious Outlier Points

A user intentionally moves his/her view directions so that all gaze points exist within the gaze region. Several gaze points in f_{Ex}, however, might be outside the region of the gaze object because (1) the view direction of a human being inevitably tends to sway, (2) the estimated result of an eye-mark recorder includes errors, and (3) a gaze point observed in the gaze region is transfered over its boundary due to an incorrect correspondence.

We remove the outlier points based on the density of the valid gaze points; points distant from the densest concentration are removed. The distance is evaluated by the following equation:

$$w_i = \left| \sum_{j=1}^{N} (x_i - x_j) \right| + \left| \sum_{j=1}^{N} (y_i - y_j) \right|, \qquad (5)$$

where N denotes the number of gaze points. $w_1, \cdots, w_N$ are calculated and then outlier detection based on least median squares[18] is executed for them. The gaze points, each of which has one of the outlier distance, are considered to be outlier points and removed from the set of the gaze points.

With this procedure, the outlier points near the region of the gaze object cannot be eliminated. These points are eliminated based on analyzing the variation of geometric configurations of gaze points (will be described in Sec. 4.1.2) and removing outlier regions (will be described in Sec. 4.4).

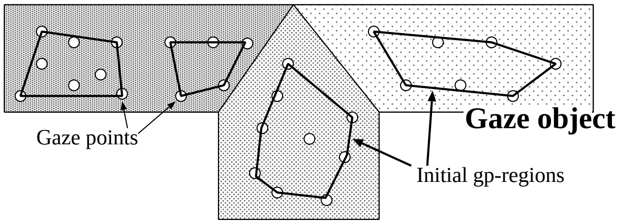


Figure 4: Generated initial regions.

4. GAZE REGION EXTRACTION FOR ESTIMATING THE GAZE REGION

4.1 Initial Regions

4.1.1 Gaze Point Grouping

In our system, some regions are generated from a set of the gaze points and extended to determine the total region of a gaze object. In the extension procedure, edge lines observed in the gaze region suppress successful extension. This problem is avoidable by making bigger initial regions. The bigger regions, however, can be generated over the boundary line of the gaze region. The region beyond the boundary results in fatal errors in extraction. To cope with this problem, we generate multiple initial regions as illustrated in Fig. 4.

Multiple initial regions are generated as follows. First of all, all gaze points in f_{Ex} are divided into several groups. We call these groups *gp-groups* (Gaze-Point groups). A convex hull of gaze points in each gp-group is regarded as an initial region. To divide all the gaze points into gp-groups taking into account the above mentioned problem, the following three points of information are considered:

Color information RGB color information is represented as a 3D vector in the RGB space. The angle θ between the RGB vectors around gaze points is calculated to determine whether or not these gaze points are classified into the same gp-group.

Spatial distribution The distance d_e between the centroid of the gaze points in a gp-group and another gaze point P_{new} is evaluated to determine whether or not P_{new} is classified into this gp-group.

$$d_e = \sqrt{(C_x - x_m)^2 + (C_y - y_m)^2}, \qquad (6)$$

where (C_x, C_y) and (x_m, y_m) denote the centroid of a gp-group and the position of a gaze point m.

Temporal trajectory The temporal information of two gaze points is represented by the length of the trajectory between them, represented by d_b:

$$d_b = \sum_{i=l}^{m-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}, \qquad (7)$$

where (x_l, y_l) denotes the coordinate of the gaze point that generates a gp-group and (x_m, y_m) denotes the coordinate of the gaze point that is evaluated whether or not classified into this gp-group.

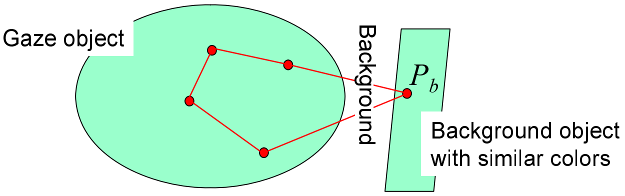


Figure 5: Error in the grouping procedure.

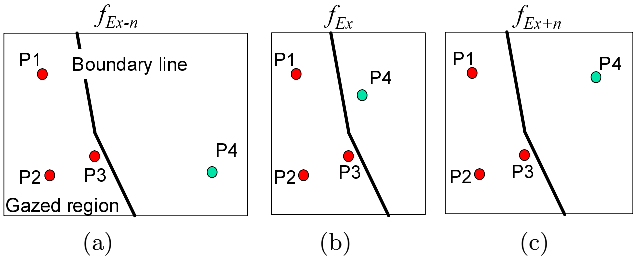


Figure 6: Outlier detection based on the change in the distance between points.

Based on these criteria, we define a correlation between a gp-group and a gaze point as follows:

$$C = \cos\theta \cdot d_e \cdot d_b \qquad (8)$$

With this function, all gaze points are segmented as follows:

Step 1 The gaze point (x_0, y_0), observed when a user starts gazing at an object, forms an initial gp-group.

Step 2 Suppose that N gp-groups have been generated. The correlations $C_{1,\cdots,N}$ between these N gp-groups and a gaze point (x_i, y_i) are calculated.

Step 3 (x_i, y_i) is then segmented to the gp-group that has the highest correlation C_{high} $(high \in \{1, \cdots, N\})$ if C_{high} is above a predefined threshold.

Step 4 If C_{high} is below the predefined threshold, on the other hand, a new gp-group is generated and (x_i, y_i) is segmented to this newly generated gp-group.

Step 5 Steps 2, 3 and 4 are applied to all gaze points.

4.1.2 Discrimination between Proper and Outlier Points in a Gp-group using Time Sequence Images

As a result of the grouping procedure described in Sec. 4.1.1, a gaze point P_b, which has a similar color to the gaze points in a gp-group that is in close proximity to P_b, might be integrated into the gp-group as illustrated in Fig. 5. Since this incorrect gp-group includes background regions, the gp-group must be corrected so that the gaze point in a background (i.e., P_b) is eliminated as an outlier point.

The targets for elimination in this section are in proximity to the gaze region whereas obvious outlier points removed in Sec. 3.2 are distant from the gaze region. These outlier points are obtained mainly when a user gazes at a moving object because it is difficult to accurately gaze at the inside of the moving object. Therefore, it is important to eliminate the outlier points and correct the gp-groups in the case of gazing at the moving object.

This correction procedure can be easily implemented if a gaze object moves under observation. The geometric configuration of the gaze points in the moving object and a background changes depending on where the object is in the image, while that only in the object it self is almost fixed as illustrated in Fig. 6. In this example, $P1$, $P2$, and $P3$ are in the object region and $P4$ is in a background. Figures (a), (b), and (c) show the gaze points transfered to different frames: Fig. (b) shows frame f_{Ex} and Fig. (a) and (c) show frames f_{Ex-n} and f_{Ex+n}, respectively. In our experiments, $n = 10$ was given in advance. In this example, $P4$ is an outlier point. Outlier elimination in this section is practically implemented as follows:

Step 1 All gaze points are transfered to frames f_{Ex-n} and f_{Ex+n} by the procedure described in Sec. 3.1.

Step 2 The distances between all pairs of gaze points in a gp-group are calculated. In an example shown in Fig. 6, $|\overrightarrow{P1 \cdot P2}|$, $|\overrightarrow{P1 \cdot P3}|$, $|\overrightarrow{P1 \cdot P4}|$, $|\overrightarrow{P2 \cdot P3}|$, $|\overrightarrow{P2 \cdot P4}|$, and $|\overrightarrow{P3 \cdot P4}|$ are calculated.

Step 3 Similar to Step 2, the distances between the gaze points in frames f_{Ex-n} and f_{Ex+n} are also calculated.

Step 4 Let $d_{max} = |\overrightarrow{(x_a, y_a) \cdot (x_b, y_b)}|$ be the maximum value in all the distances in f_{Ex-n}, f_{Ex}, and f_{Ex+n}. If d_{max} is larger than a predefined threshold, $P_a = (x_a, y_a)$ or $P_b = (x_b, y_b)$ is regarded as an outlier point.

Step 5 The summation of the distances between P_a and all other points is calculated in frames f_{Ex-n}, f_{Ex}, and f_{Ex+n}. Let S_a be the sum of three summations. Similarly, S_b for P_b is also calculated.

Step 6 If $S_a > S_b$, P_a is considered to be an outlier point and eliminated. Otherwise, P_b is eliminated.

This procedure allows the system to infallibly eliminate the outlier points that are wrongly included in the group within the gaze object. Some outlier points may remain near the region of the gaze object and compose outlier regions. These regions are removed by the process described in Sec. 4.4.

4.1.3 Generating Initial Regions

After the gaze points in each gp-group are determined, the region of each gp-group should be generated. We call this region a *gp-region*. In our method, the convex hull that consists of the gaze points in each gp-group is regarded as the initial state of a gp-region. This convex hull is generated by employing the Quick Hull algorithm proposed in [19]:

Step 1 If the number of the gaze points in a gp-group is less than three, the gaze points and their vicinities are regarded as the initial state of a gp-region.

Step 2 Otherwise, three arbitrary gaze points are selected for generating a triangle that is regarded as an initial hull.

Step 3 If there exists any gaze point outside the convex hull, the point that is the most distant from a side of the convex hull is selected. This point and side are denoted by P_d and L_d, respectively.

Step 4 The two line-segments between P_d and two vertices of L_d are inserted into the convex hull instead of L_d.

Step 5 Steps 3 and 4 are continued until no gaze point is outside the convex hull.

4.2 Edge Detection

For image segmentation, the boundary line between observed objects provides useful information. The boundary line can be detected by edge detection. In our system, the method proposed in [16] is employed for color edge detection. After edge detection, edge thinning and short edge elimination are executed so that the boundary line is indicated as clearly as possible.

4.3 Region Extension

The boundary line of a gaze object is estimated by extending the initial gp-regions. For the extension, taking into account the following factors, we employ Snakes[17]. Let $\vec{v}(s)$ denote s-th point located along the boundary of each gp-region. In our system, Eq. (9) is utilized to adjust the extension so that (1) the boundary of each gp-region is smooth, (2) the extension is suppressed by edge lines, and (3) the extension gets weaker as the boundary of the gp-region is more distant from its centroid:

$$E_{snake} = \int_{s=0}^{1} \{E_{int}(\vec{v}(s)) + E_{img}(\vec{v}(s)) + E_{grv}(\vec{v}(s))\} ds, \quad (9)$$

$$\begin{aligned} E_{int}(\vec{v}(s)) &= \alpha|\vec{v}_s(s)| + \beta|\vec{v}_{ss}(s)|, & (10)\\ E_{img}(\vec{v}(s)) &= \gamma(-|\nabla I(\vec{v}(s))|), & (11)\\ E_{grv}(\vec{v}(s)) &= \delta(-|\vec{v}(s) - g(\vec{v}(s))|), & (12) \end{aligned}$$

where

- $\vec{v}_s = d\vec{v}/ds$,
- $\vec{v}_{ss} = d^2\vec{v}/ds^2$,
- $I(\vec{v}(s))$ and $g(\vec{v}(s))$ denote the color value in $\vec{v}(s)$ and the centroid of all the points along the boundary, respectively, and
- α, β, γ, and δ are weighted constants.

Extension of each point on the extending gp-region finishes when the gp-region touches another gp-region.

4.4 Removing Outlier Regions

The extended gp-regions are integrated in order to compose the region of a gaze object. This integration procedure consists of the following processes; detection of a reliable gp-region and integration of neighboring gp-regions based on similarity of color and motion information.

First, the number of gaze points included in each extended gp-region is counted. Then, the gp-region having the maximum number is regarded as the gp-region that can be reliably considered to be included in the gaze region.

Next, the gp-regions, which touch the reliable gp-region and have a similar color to it, are merged. Note that this process integrates only the gp-regions that have similar colors, but the gp-regions that have different colors may be included in the gaze region. These gp-regions will be merged with the estimated gaze region by the following integration process.

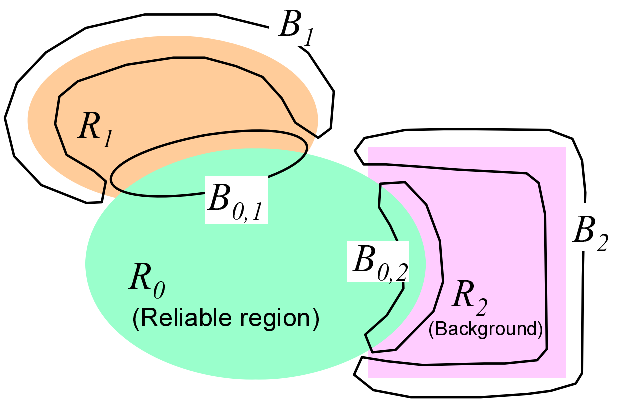


Figure 7: Outlier region elimination based on motion vector analysis.

In the integration process, we should notice that outlier points remaining near the gaze object compose outlier gp-regions as mentioned in the last paragraph of Sec. 4.1.2. If an extended outlier gp-region does not touch the gaze region, this outlier gp-region can be easily distinguished from the extended gp-regions included in the gaze region. All extended gp-regions are, however, close to each other because obvious outlier gaze points are eliminated in the procedure described in Sec. 3.2. We have to, therefore, detect those outlier gp-regions based on another criterion. Similar to outlier point elimination described in Sec. 4.1.2, motion information of the gaze object is useful to discriminate between proper and outlier gp-regions. That is, motion vectors of a moving gaze object and other regions are different from each other. The motion vectors are analyzed to find out whether they are merged with the reliable gp-region as follows.

Let R_0 be the region that is currently regarded as the gaze region by the above mentioned procedure. The system then examines whether neighboring gp-regions (R_1 and R_2 in an example shown in Fig. 7) are merged with the gaze region. For the examination, two kinds of typical motion vectors in the boundary of each neighboring gp-region are compared; one is in the overlapping area between the current gaze region and each neighboring gp-region ($B_{0,1}$, for example) and the other one is in the rest of the boundary line of the neighboring gp-region (B_1, for example). The median of the motion vectors is regarded as the typical motion vector. If the two vectors are close to each other, the neighboring gp-region is merged with the gaze region. Otherwise, the neighboring gp-region is is considered to an outlier gp-region. This integration procedure is continued until all neighboring gp-regions are examined whether or not they are merged. In an example shown in Fig. 7, A_1 is merged but R_2 is not merged.

Finally, the merged region is considered to be the target gaze region.

5. EXPERIMENTS

All experiments were conducted under the following conditions:

- The image size is 320 x 240 pixel.
- RGB images and gaze points were observed at 15 fps.
- Each observation was for two seconds.

Under this condition, it took about 2∼3 sec to obtain each extraction result through the whole processes described in this paper by using a Pentium4 3.2GHz PC[3].

The experimental results are shown in Fig. 8 (observing a static object) and 9 (observing a moving object):

Fig (a) Partial image and gaze point sequence observed while a user moves his/her head

Fig (b) Image in which extraction is executed and the gaze points which are transfered to this image

Fig (c) Initial gp-regions (described in Sec. 4.1)

Fig (d) Edge image (described in Sec. 4.2)

Fig (e) The result of region extraction (described in Sec. 4.3)

Fig (f) The result of removing outlier gp-regions (described in Sec. 4.4)

Fig (g) Extraction result

When gazing at a static object (Fig. 8), the extracted result was very good because most of the gaze points were within the region of the target and their number was sufficient to determine the region. In Fig. 9 (c), on the other hand, it can be seen that a number of outlier gaze points were observed when a user gazed at a moving object. Note that obvious outlier points (indicated by ***a*** in Fig. 9 (b)) were eliminated in Fig. 9 (c) but the outlier points near the moving object (indicated by ***b*** in Fig. 9 (b)) were not eliminated. These outlier points generate a gp-region and the region was extended as shown in Fig. 9 (e) (indicated by ***c***). This region was, however, eliminated taking into account the difference of motion vectors as shown in Fig. 9 (f) and (g).

We also conducted other experiments: observing a mobile robot (Fig. 10), observing one of moving people (Fig. 11), and observing a walking person in a complicated background (Fig. 12). It can be confirmed that our proposed system can work for various target objects.

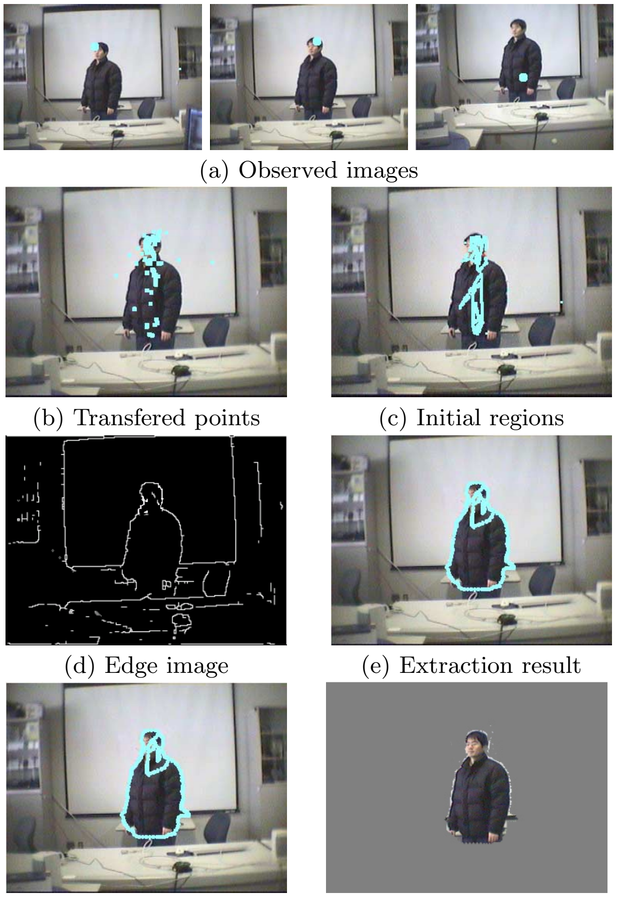
6. CONCLUDING REMARKS

This paper presented the system that analyzes the spatio-temporal information of observed images and gaze points to extract the region of an object gazed at by a user. The proposed system can extract both static and moving objects from images observed while the user moves his/her head. Since the extracted region shows what the user gazes at, this information is useful for analyzing human attention.

We are improving the system in terms of the following aspects:

- With the current implementation, a user must wait for an extraction result for about 3 sec. This response time is not suitable for an online interface system. Since

[3] The speed performance of the current system is not satisfying but applicable to providing simple information of what he/she is interested in (e.g., the owner/price of an object). However, some of expansive applications (e.g., teaching how to operate a device at each moment[20]) require a real-time processing.



(a) Observed images

(b) Transfered points

(c) Initial regions

(d) Edge image

(e) Extraction result

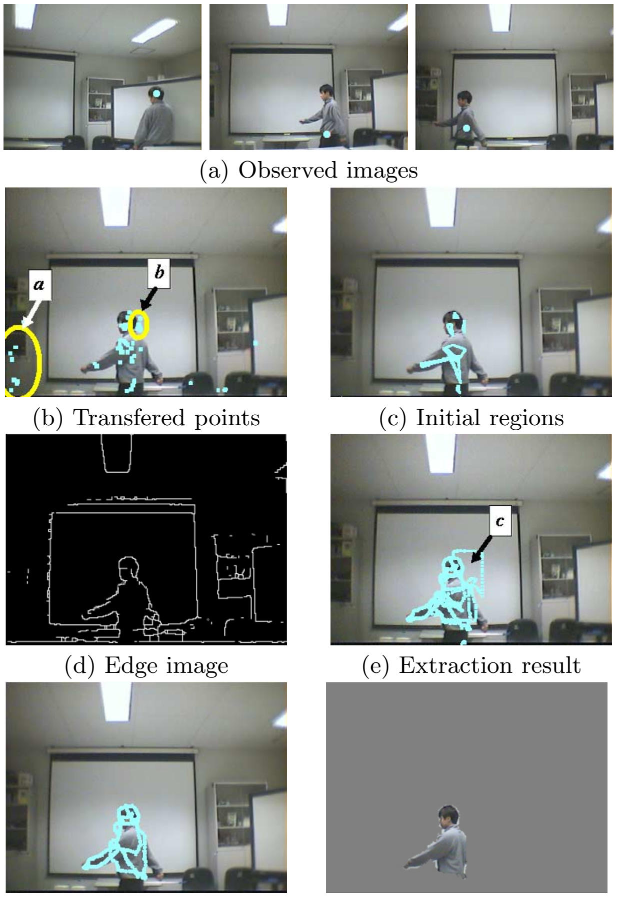
(f) Outlier elimination

(g) Extraction result

Figure 8: Extraction result of a static person.

most of the computational time is spent on transferring gaze points, we estimate that the response time can be within 1 sec by shortening the computational cost for transferring gaze points.

- The result of our system depends on an estimated edge image while the edge information in sequential images may differ from each other due to several factors such as image noise, change in shade and background colors near the boundary of a gaze object). The performance of the system, therefore, can be improved by integrating the extraction results in multiple images.
- A sophisticated active contour method can improve the performance of region extraction (see [21], for example) and can be used also for tracking a gaze object after extraction.
- We have developed a system for estimating the 3D position gazed at by a user by employing the history of binocular view lines[22]. In this system, the characteristic of the human gaze, in which binocular view lines intersect at one point when he/she gazes at something, is employed. With this characteristic, the system proposed in this paper can be augmented so that a user can gaze at multiple objects simultaneously. That is, a set of continuously crossing view lines give the gaze points when he/she gazes at each object.



(a) Observed images

(b) Transfered points

(c) Initial regions

(d) Edge image

(e) Extraction result

(f) Outlier elimination

(g) Extraction result

Figure 9: Extraction result of a moving person.

This research is supported by CREST Program “Advanced Media Technology for Everyday Living” of JST and the Grant-in-Aid for Scientific Research, No.15700157.

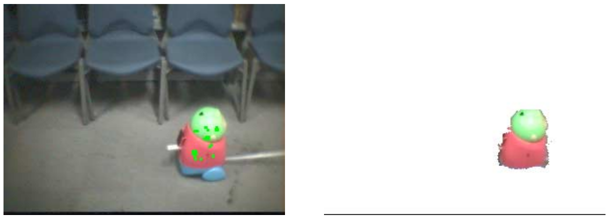
7. REFERENCES

[1] R. Tenmoku, M. Kanbara, and N. Yokoya: “A wearable augmented reality system for navigation using positioning infrastructures and a pedometer,” in Proc. of IEEE and ACM International Symposium on Mixed Augmented Reality, pp.344–345, 2003.

[2] P. Daehne and J. Karigiannis: “Aecheoguide: System Architecture of a Mobile Outdoor Augmented Reality System,” in Proc. of International Symposium on Mixed Augmented Reality, pp.263–264, 2002.

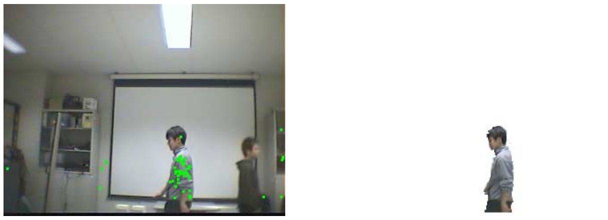
[3] S. Feiner, B. MacIntyre, T. Holler, and A. Webster: “A Touring Machine: Prototyping 3D Mobile Augmented Reality Systems for Exploring the Urban Environment,” in Proc. of International Symposium on Wearable Computers, pp.74–81, 1997.

[4] R. Cipolla and H. J. Hollinghurst, “Human-robot interface by pointing with uncalibrated stereo vision”, *Image and Vision Computing*, Vol.14, No.3, pp.178–178, 1996.



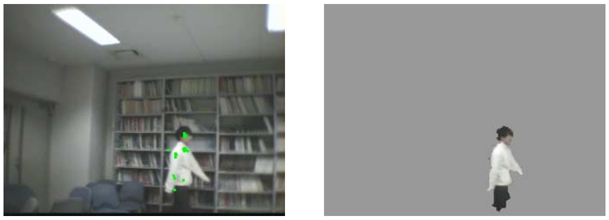
(a) Observation result (b) Extraction result

Figure 10: Extraction result of a moving robot.



(a) Observation result (b) Extraction result

Figure 11: Extraction result of a moving person.



(a) Observation result (b) Extraction result

Figure 12: Extraction result of a moving person in a complicated background.

[5] T. Ohno and N. Mukawa: "Gaze-Based Interaction for Anyone, Anytime", in Proc. of HCI International 2003, Vol.4, pp.1452–1456, 2003.

[6] Y. Matsumoto, T. Ino, and T. Ogasawara: "Development of Intelligent Wheelchair System with Face and Gaze Based Interface", in Proc. of 10th IEEE Int. Workshop on Robot and Human Communication (ROMAN 2001), pp.262–267, 2001.

[7] R. Atienza, A. Zelinsky: "Interactive skills using active gaze tracking", in Proc. of International Conference on Multimodal Interfaces, pp.188–195, 2003.

[8] A. Sugimoto, A. Nakayama, and T. Matsuyama: "Detecting Gazing Region by Visual Direction and Stereo Cameras", in Proc. of International Conference on Pattern Recognition 2002, Vol.3, pp.278–282, 2002.

[9] Y. Rosenberg and M. Werman: "Real-Time Object Tracking from a Moving Video Camera: A Software Approach on a PC," in Proc. of IEEE Workshop on Applications of Computer Vision, pp.238–239, 1998.

[10] J. Frazier and R. Nevatia, "Detecting Moving Objects from a Moving Platform," in Proc. of International Conference on Robotics and Automation, pp.1627–1633, 1992.

[11] Y. Sugaya and K. Kanatani: "Extracting moving objects from a moving camera video sequence," in Proc. of the 10th Symposium on Sensing via Imaging Information, pp.279–284, 2004.

[12] N. Ukita, A. Sakakihara, and M. Kidode: "Extracting a Gaze Region with the History of View Directions," in Proc. of 17th International Conference on Pattern Recognition, Vol.4, pp.957–960, 2004.

[13] D. I. Barnea and H. F. Silverman: "A class of algorithms for fast digital image registration," IEEE Trans. Computers, Vol.21, pp.179–186, 1972.

[14] Q. Zheng, R. Chellappa: "Automatic Feature Point Exaction and Tracking in image Sequences for Arbitrary Camera Motion," International Journal of Computer Vision, Vol.9, No.2, pp.31–76, 1995.

[15] J. Shi and C. Tomasi: "Good Features to Track," in Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pp.593–600, 1994.

[16] S. Wesolkowski and E. Jernigan: "Color Edge Detection in RGB Using Jointly Euclidean Distance and Vector Angle," in Proc. of Vision interface '99, pp.19–21, 1999.

[17] M. Kass, A. Witkin, and D. Terzopoulos: "Snakes: Active Contour Models," *International Journal of Computer Vision*, Vol.1, No.4, pp.321–331, 1988.

[18] R. J. Rousseeuw and A. M. Leroy: "Robust Regression and Outlier Detection," John Wiley & Sons, 1987.

[19] C. B. Barber, D. P. Dobkin, H. Huhdanpaa: "The Quickhull Algorithm for Convex Hulls", ACM Trans. on Mathematical Software, Vol.22, No.4, 1996.

[20] S. Finer, B. MacIntyre, and D. Seligmann: "Knowledge-based Augmented Reality," Communication of the ACM, Vol.37, No.7, pp.52–62, 1993.

[21] Y. Rathi, N. Vaswani, A. Tannenbaum, and A. Yezzi: "Particle Filtering for Geometric Active Contours with Applications to Tracking Moving and Deformable Objects," in Proc. of IEEE Conference on Computer Vision and Pattern Recognition, Vol.2, pp.2–9, 2005.

[22] I. Mitsugamai, N. Ukita and M. Kidode: "Estimation of a 3D Gaze Position using View Lines," in Proc. of 12th International Conference on Image Analysis and Processing, pp.466–473, 2003.