Extracting a Gaze Region with the History of View Directions

Norimichi Ukita, Akihito Sakakihara and Masatsugu Kidode
Graduate School of Information Science, Nara Institute of Science and Technology
{ukita,kidode}@is.naist.jp

Abstract

We propose a method for extracting a gaze region from an observed image by analyzing human's view directions and image information. The view direction of the user, which is represented as a 2D gaze point in the observed image, is obtained by an eye-mark recorder at every image-capturing timing. All gaze points are translated to one of the images for extracting the gaze region based on the history of the view directions. The system divides all gaze points into several groups by comparing color information etc., and then generates several convex hulls as initial regions. Each initial region is extended based on its color information and the spatial distribution of the gaze points. All regions are finally integrated and regarded as the gaze region.

1. Introduction

If a camera mounted by a person can capture the same image he/she is looking at, that image can provide invaluable information about what he/she is thinking about. The observed image may have several components, including not only gazed objects but also a background. To understand the user's attention, the region of interest to the user should be extracted from the observed image. We call this image region of interest the *gaze region*. [1] proposed a stereo-vision system that estimates a gaze region in an observed image. In this system, however, only a 3D planar object can be extracted. For general applications, a segmentation method that can extract a gaze region corresponding to arbitrary 3D object(s) is required.

We employ the following device and method to extract an arbitrary gaze region. 1) With a device that estimates in which direction a person is looking, the points in the observed image gazed at by him/her can be obtained. The temporal history and the spatial distribution of these points are taken into account when initially estimating the gaze region. 2) Next, the color distribution and the edge detected around the gaze points are analyzed to determine the gaze region precisely.

2. System Overview

To measure a user's view line, we employ the EMR-8 made by *NAC Inc.* The direction of a user's view (*view direction*) is measured by the *corneal reflection-pupil center method*. The view camera is positioned at the mid-point half way between the user's eyes. The view direction at each observation timing is represented as a 2D point in the image observed by the view camera. We call this point a *gaze point*. The above data is acquired at 15 fps.

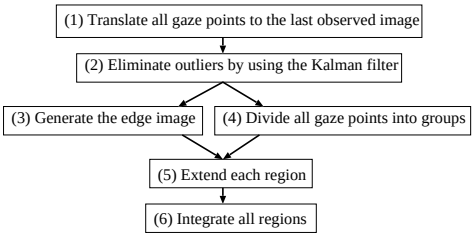

Figure 1. Processing flow diagram.

The view camera captures sequential images from when a user starts gazing at an object to when he/she stops. The task of the system is to extract the region of the gazed object from the image observed when a user stops gazing at the object, namely from the last observed image. In our system, a user has to intentionally move his/her view directions so that 1) all gaze points exist within the gaze region and 2) its area is almost covered by the gaze points. We evaluate the accuracy of the estimated result by comparing the extracted region and the ground truth given by a user in Section 4.

Our system has the following restrictions about a target.

- Gaze at an unmovable rigid object and fix a user's head during the observation: If this restriction is broken, the system has to calculate where each gaze point is translated to in the last observed image. If this restriction is kept, on the other hand, all gaze points can be superimposed directly on any observed image.
- Gaze at an object whose region in the observed image is sufficiently: Since a view direction is unsteady even if a person tries to gaze at a fixed position, it is difficult to gaze at a small region.

The basic scheme of the system is illustrated in Fig. 1.

1. **Translate all gaze points to the last observed image:** All gaze points are displayed in the last image.
2. **Eliminate outliers by using the Kalman filter:** A gaze point that is away from the estimated trajectory of the gaze points is eliminated.
3. **Generate the edge image:** The edge lines around the trajectory of the gaze points are erased.
4. **Divide all gaze points into several groups:** A group of the divided gaze points form a initial region.

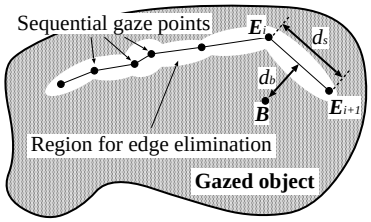

Figure 2. Edge elimination along the trajectory of gaze points.

5. **Extend each region:** Each region is extended to absorb adjacent pixels whose image information is similar to that of the extending region.
6. **Integrate all regions:** The integrated region is regarded as the whole region of the gazed object.

3. Extracting a Gaze Region using the History of Gaze Points

3.1. Representation of View Directions

Suppose that the system observes N images, I_1, I_2, $\cdots, I_N$, each of which has the information of a gaze point. All the gaze points have to be translated to I_N. Since the region of the gazed object is fixed in the all images, the translated 2D positions of the gaze points in I_N are identical to the original 2D positions observed in $I_1, I_2, \cdots, I_N$ (mentioned in Section 2).

3.2. Outlier Elimination using the Kalman Filter

Although a user intentionally moves his/her view directions so that all gaze points exist within the gaze region, several gaze points might be out of the gaze region because 1) a view direction sways inevitably and 2) the estimated result of an eyemark recorder includes errors. While it is hard to detect all error points, a point that swerves from the estimated gaze trajectory can be detected as follows.

The state vector of a gaze point is defined as

$$\boldsymbol{x}_i = \left\{(x_i, y_i, v_{i_x}, v_{i_y})\right\}^\top, \tag{1}$$

where (x_i, y_i), (v_{i_x}, v_{i_y}) denote the position and the velocity of the gaze point observed in I_i, respectively. Let $\boldsymbol{y}_i$ denote the measurement vector. The state equation and the measurement equation are, then, defined as follows:

$$\boldsymbol{x}_{i+1} = \boldsymbol{F}\boldsymbol{x}_i + \boldsymbol{G}\boldsymbol{w}_i, \tag{2}$$

$$\boldsymbol{y}_i = \boldsymbol{H}_i + \boldsymbol{v}_i, \tag{3}$$

where $\boldsymbol{w}_i$ and $\boldsymbol{v}_i$ represent the system noise and the measurement noise, respectively, and the state transition matrix $\boldsymbol{F}$, the control matrix $\boldsymbol{G}$ and the measurement matrix $\boldsymbol{H}$ are represented as follows:

$$\boldsymbol{F} = \begin{pmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & 0 & \Delta T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$\boldsymbol{G} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}^\top, \qquad \boldsymbol{H} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix},$$

where ΔT denotes an interval between sequential measurements. In the above representations, the motion of a view direction is assumed as constant linear velocity. With these definitions, we employ the following equations as the Kalman filter:

$$\boldsymbol{K}_i = \tilde{\boldsymbol{P}}_i \boldsymbol{H}^T (\boldsymbol{I}_{2\times 2} + \boldsymbol{H}\tilde{\boldsymbol{P}}_i \boldsymbol{H}^T)^{-1}, \tag{4}$$

$$\tilde{\boldsymbol{x}}_{i+1} = \boldsymbol{F}\{\tilde{\boldsymbol{x}}_i + \boldsymbol{K}_i(\boldsymbol{y}_i - \boldsymbol{H}\tilde{\boldsymbol{x}}_i)\}, \tag{5}$$

$$\tilde{\boldsymbol{P}}_{i+1} = \boldsymbol{F}(\tilde{\boldsymbol{P}}_i - \boldsymbol{K}_i \boldsymbol{H}\tilde{\boldsymbol{P}}_i)\boldsymbol{F}^T + \frac{\sigma_w^2}{\sigma_v^2}\Lambda, \tag{6}$$

where

- $\tilde{\boldsymbol{x}}_i = \hat{\boldsymbol{x}}_{i|i-1}$,
- $\hat{\boldsymbol{x}}_{i|i-1}$ denotes the estimated $\boldsymbol{x}_i$ when $\boldsymbol{y}_0, \ldots, \boldsymbol{y}_{i-1}$ are given,
- $\tilde{\boldsymbol{P}}_i = \hat{\boldsymbol{P}}_{i|i-1}$,
- $\hat{\boldsymbol{P}}_{i|i-1} = \hat{\sum}_{i|i-1}/\sigma_v^2$,
- $\hat{\sum}_{i|i-1}$ denotes the estimated error covariance matrix,
- $\boldsymbol{K}_i$ denotes the Kalman gain, and
- $\Lambda = \boldsymbol{G}\boldsymbol{G}^T$.

The proposed system compares the observed point $\boldsymbol{y}_{i+1}$ with the estimated point $\tilde{\boldsymbol{x}}_{i+1}$. If the distance between these two points is longer than the predefined threshold, $\boldsymbol{y}_{i+1}$ is regarded as an outlier and removed from the set of the observed gaze points.

3.3. Edge Detection and Elimination based on the Gaze Trajectory

For image segmentation, the boundary line between observed objects provides useful information. The boundary line can be detected by edge detection.

The detected edge lines, however, include not only the boundary line between objects but also various components, e.g., texture patterns and shadows. Edge lines except for the boundary line between objects disturb the correct extraction of a gaze region. To reduce this harmful influence, the trajectory of gaze points is useful information. Since all gaze points must be within the region of the gazed object, all edge lines between gaze points can be eliminated.

Edge elimination is practically implemented as follows (Fig. 2). Let 1) E_i and E_{i+1} denote two subsequent gaze points and 2) d_s denote the distance between E_i and E_{i+1}. When the sobel operator is applied to the observed image, a threshold for binarizing the pixel at $\vec{B}$ (represented by T_e) is determined by the following equation:

$$T_e = T_b + \frac{T_{gain}}{d_s \times d_b}, \tag{7}$$

where d_b and T_{gain} denote the distance between $\vec{B}$ and $\overline{E_i E_{i+1}}$ and a predefined constant, respectively. With this threshold, as the distance between $\vec{B}$ and the gaze trajectory becomes shorter, the threshold at $\vec{B}$ becomes larger.

3.4. Generating Initial Regions

All gaze points are divided into several groups. We call these groups *gp-groups*. To divide the gaze points into gp-groups, we should consider the following conflict problems:

- A large initial region is suitable for extending itself without being interfered by error edge lines.
- If many gaze points distributed in a wide area are segmented to a gp-group, the convex hull including these points might be partly out of the region of a gazed object.

Considering the above two problems, we group all gaze points taking into account the following three points of information:

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 are compared with each other for determining whether or not these gaze points are divided 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.

$$d_e = \sqrt{(C_x - x_m)^2 + (C_y - y_m)^2}, \qquad (8)$$

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:

$$d_b = \sum_{i=l}^{m-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}, \qquad (9)$$

where (x_l, y_l) and (x_m, y_m) denote the coordinates of the gaze points that are lastly classified into a gp-group and newly evaluated whether or not classified into this gp-group, respectively.

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

$$C = \alpha \times \cos\theta + \beta \times d_e + \gamma \times d_b, \qquad (10)$$

where α, β and γ denote predefined weighted-constants. 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.

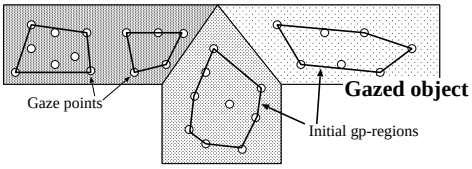

Figure 3. Generated initial gp-regions.

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

After all gaze points are divided into gp-groups, 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 [3]:

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, arbitrary three 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 one of the sides 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 out of the convex hull.

Figure 3 shows an example of initial gp-regions.

3.5. Region Extension

The area of each gp-region is extended to estimate the region of a gazed object. Previous region-extraction methods take into account edge lines and color distribution for estimating boundary lines. Most of these methods, however, have the following problems:

Problem 1 An edge line strongly suppresses the extension of a region.

Problem 2 If an edge line is not plainly detected at the boundary of a target object, the extended region is widely spread outside the target.

We cope with these problem as follows:

Problem 1 Since crowded edge lines/points generated by texture patterns should not prevent the extension of a gp-region, the system evaluates whether or not a detected edge is the part of a straight line.

Problem 2 As the boundary of a gp-region is far from its gaze points, the system suppresses the extension.

Let R be a square region whose center P is a point at the boundary line of a gp-region. The region of R, except for the gp-region, is denoted by R_0 as illustrated in Fig. 4. If the following two conditions are satisfied, R_0 is added to the gp-region.

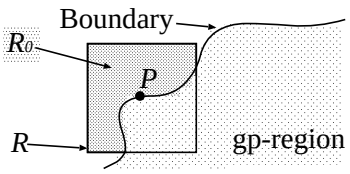

Figure 4. Region extension.

Condition 1 There is no straight edge line in R_0. To determine whether or not an edge line exists in R_0, the following value p is evaluated:

$$p = \frac{\sum(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum(X_i - \bar{X})^2}\sqrt{\sum(Y_i - \bar{Y})^2}}, \quad (11)$$

where (X_i, Y_i) and $(\bar{X}, \bar{Y})$ denote the x-y coordinates of each gaze point and their centroid, respectively. If p is below a predefined threshold, it is considered that there is no straight edge line in R_0.

Condition 2 The difference between the standard deviations of RGB values in R and R_0 is less than a predefined threshold T.

$$T = \frac{D_{avr}}{D} \times T_0, \quad (12)$$

$$\frac{|\,\sigma_0 - \sigma\,|}{\sigma} \leq T, \quad (13)$$

where 1) σ and σ_0 denote the standard deviations of RGB values in R and R_0, respectively, 2) D_{avr} denotes the average of the distances between the gaze points in the gp-region and their centroid, 3) D denotes the distance between P and the centroid of the gaze points in the gp-region, and 4) T_0 denotes the predefined threshold. With this condition, the extension of the boundary line is suppressed depending on the spatial distribution of the gaze points.

Region extension halts when the above two conditions are not satisfied at any point p at the boundary line of a gp-region. All the gp-regions are, then, added to a single region, which is regarded as the whole region of the gazed object.

4. Experimental Results

We conducted experiments to verify the effectiveness of the system. In these experiments, a user gazed at 1) a backpack on a sofa and 2) a backpack and a sofa. In each experiment, a user gazed at an object for about five seconds.

The experimental results are shown in Fig. 5 and 6. The size of each image is 640 $\times$ 480 pixel. In each figure, (a) observed gaze points, (b) gaze points except for outliers, (c) edge image, (d) result of edge elimination, (e) initial gp-regions, and (f) extracted result are shown. From these results, we can confirm that the proposed system can correctly extract a gaze region depending on the user's focus area.

We also evaluated the quantitative performance of the system by comparing the extracted result and the ground truth given by a user (Table 1). While the true-positive rate in the case of gazing at the backpack is low, all other results show the effectiveness of the proposed system.

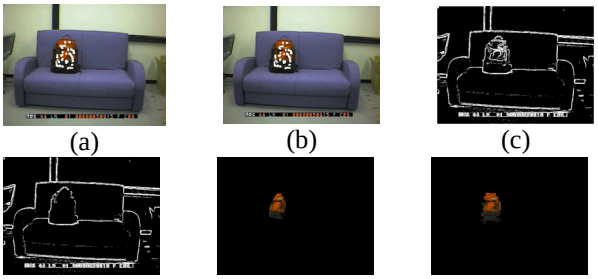

Figure 5. Experimental results: backpack.

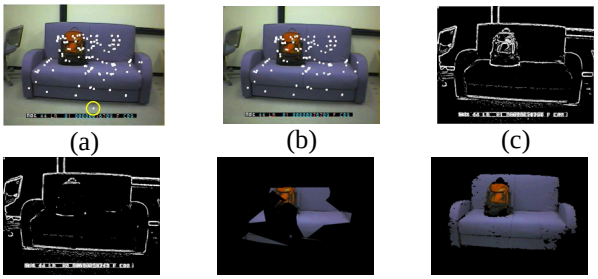

Figure 6. Experimental results: sofa.

Table 1. Quantitative evaluation.

	true-positive	faulse-positive
backpack	65.7%	0.0%
backpack and sofa	88.9%	0.0%

5. Concluding Remarks

The system integrates the temporal sequence of gaze points and extract the image region gazed at by a user.

Following are the future works:

Implementation in real time Since several computations take a great deal of time, the proposed algorithm cannot work in real time.

Extraction of a moving object All gaze points have to be correctly translated to one of the observed images based on image correlation or other criteria.

This research is supported by CREST Program of JST and the Grant-in-Aid for Scientific Research, No.15700157, 2004.

References

[1] A. Sugimoto, A. Nakayama, and T. Matsuyama: "Detecting Gazing Region by Visual Direction and Stereo Cameras", in Proc. of ICPR 2002, Vol.3, pp.278–282, 2002.

[2] M. Kass, A. Wiktin, D. Terzopulos: "SNAKES: Active Contour Models", in Proc. of 1st ICCV, pp.259–268, 1987.

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