Acquiring a Human's Attention by Analyzing the Gaze History

Norimichi Ukita, Akihito Sakakihara, Ikuhisa Mitsugami and Masatsugu Kidode
Artificial Intelligence Laboratory, Graduate School of Information Science,
Nara Institute of Science and Technology.
e-mail: {ukita,ikuhi-mi,kidode}@is.aist-nara.ac.jp

Abstract

Our objective is to realize wearable interfaces for expanding our information activities in daily life. To implement these interfaces, we employ vision-based systems, including a variety of camera, a head-mounted display and other visual devices. In this paper, we present two wearable vision interfaces; 1) 2D gaze-region extraction from an observed image and 2) 3D gaze-position estimation. In both of these interfaces, the eye-mark recorder is employed to obtain the history of human's gazing directions. The information of human's gaze is useful for analyzing human's attention. Experimental results have demonstrated the effectiveness of the proposed interfaces.

1 Introduction

We have been studying to develop man-machine interfaces for next-generation wearable information playing[1]. To realize these interfaces, we employ a vision-based system, including various cameras, a head-mounted display (HMD, in short) and other miniaturized visual devices. A wearable vision system observes the user's viewpoint image with camera(s) and analyses the observed image. Based on the result of the image analysis, the system show the useful information (e.g., images and annotations) to the user.

For the system to automatically understand what a user want to do, information of a user's attention is important. That is, the system should know which object a user takes an interest in. We can inform our attention of a computer system by several ways, e.g., finger pointing, gaze, and so on. Above all, a gaze direction has prominent features for acquiring a human's attention because 1) a human naturally gazes at an object that he/she is interested in and 2) a user can gaze at an object even while he/she performs another task. We, therefore, put our focus upon how to simply and intuitively acquire a human's attention by analyzing the gaze history.

In this paper, we introduce two wearable interfaces, each of which acquires the following information:

2D gaze region The region that a user gazes at is extracted from the observed image.

3D gaze position The position in the scene, which a user gazes at, is estimated.

In Sections 2 and 3, we describe the above two interfaces, respectively. Section 4 summarizes the works proposed in this paper and points out future works for next-generation wearable vision interfaces.

2 Extracting a 2D Gaze Region from an Observed Image

If a computer system recognizes which object a person is interested in, the system can give useful information of this object to him/her. Since a wearable camera near a user's viewpoint observes the image whose appearance is similar to his/her sight, this image is valuable information for understanding a user's current attention. There generally exist several objects, including a background scene, in the observed image. To understand a user's attention, the gaze region should be segmented and extracted from the complicated observed image.

Several works about how to obtain the gaze information (i.e., a gaze direction) have been reported and a wearable device for estimating a gaze direction has been also developed. With this device, a user can roughly specify the region of an object, which he/she is interested in, to a computer system.

Region segmentation is, on the other hand, one of the most important subjects in Pattern Recognition (see [2, 3], for example). The regions of all observed objects are segmented based on image information, e.g., edge lines, color distribution, and so on.

The function and idea of the above two works (i.e., estimation of a gaze direction and region segmentation) can be applied to the system that segments and extracts the image region gazed by a human. Accordingly, 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 a 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 projected to the last observed image for extracting the gaze region based on the history of the gaze directions. After the user stops gazing, the system divides all gaze points into several groups by comparing color information etc., and then generate convex hulls as initial regions. To acquire the whole gaze region, each initial region is

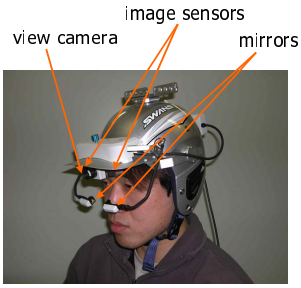


Figure 1: Binocular eyemark recorder EMR-8.

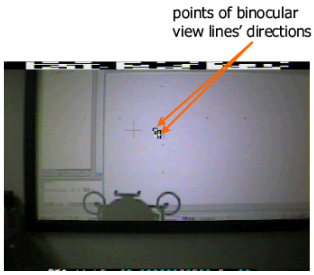


Figure 2: Output image of EMR-8.

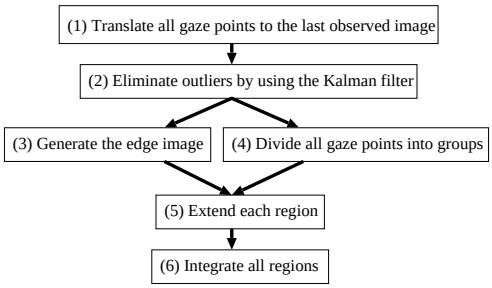


Figure 3: Processing flow diagram of the proposed system.

extended based on its color information and the spatial distribution of the gaze points. We show experimental results to confirm the effectiveness of the proposed method.

2.1 How to acquire and represent Gaze Information

To measure a user's view line, an eyemark recorder is often used. We exploit the EMR-8(Figure 1) made by *NAC Inc.* The EMR-8 consists of a view camera, two image sensors and two mirrors. The user's eyes are observed by the image sensors through the mirrors, and their directions (we call it a *view direction*) are measured by the *corneal reflection-pupil center method*[7]. The view camera is almost at the center of two eyes and observes 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 (Figure 2). We call this point a *gaze point*. The above data is acquired at 15 [frame/second].

2.2 System Overview

The system obtains sequential images, on each of which a gaze point is overlaid. Note that the gaze points observed by a single eye are employed though our EMR-8 can acquire the gaze points of both eyes simultaneously. These images are captured from when a user starts gazing at an object to when he/she stops it. The task of the proposed system is to extract the region of the gazed object from the image observed when the 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 all gaze points exist within the gaze region and its area is almost covered by the gaze points. The system estimates the region of the gazed object based on the spatial distribution of the acquired gaze points and the image information around them. We evaluate the accuracy of the estimated result by comparing the extracted region and the ground truth given by a user. The experimental results are shown in Section 2.5.

We assume the following restrictions:

- Gaze at an unmovable rigid object and fix a user's head during the observation:
  If one or more of these restrictions are broken, the appearances of the gazed object are varied among the observed images. The system, then, has to calculate where each gaze point is translated to in the last observed image. To make the problem easy, we impose the above restrictions on the system.
- Gaze at an object whose projected region is enough large in the observed image:
  Since a view direction is unsteady even if we try to gaze at a fixed position, it is difficult for a user to gaze at a small object and an object distant from him/her.

The basic scheme of the proposed system is illustrated in Fig. 3.

1) **Translate all gaze points to the last observed image:** To extract the region of the gazed object from the last observed image, all gaze information has to be translated to this image.
2) **Eliminate outliers by using the Kalman filter:** The trajectory of the gaze points is estimated with the Kalman filter. Gaze points with large error are then detected and eliminated from a group of the gaze points.
3) **Generate the edge image:** Initially, edge lines are detected with conventional edge detection. The edge lines along the trajectory of the gaze points are then erased.
4) **Divide all gaze points into groups:** All gaze points are divided into several groups based on the color information and the spatial distribution of the gaze points. A group of the divided gaze points form a initial region.
5) **Extend each region:** Each region is extended to absorb adjacent pixels whose image information is similar to each other.

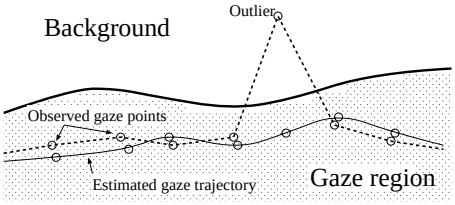


Figure 4: Outlier detection using the Kalman filter.

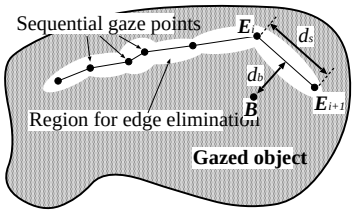


Figure 5: Edge elimination along the trajectory of gaze points.

6) **Integrate all regions:** Finally, all regions are integrated and the integrated region is regarded as the region of the gazed object.

In what follows, we describe the details of the above processes.

2.3 Gaze Information

2.3.1 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. As mentioned in Section 2.2, all the gaze points are within the region of the gazed object and it does not move in the observed images. The 2D positions of the gaze points in I_N (represented by $(x_1^N, y_1^N), (x_2^N, y_2^N), \cdots, (x_N^N, y_N^N)$) are, therefore, identical to the 2D positions observed in $I_1, I_2, \cdots, I_N$ (represented by $(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)$).

2.3.2 Outlier Elimination using the Kalman Filter

As mentioned in Section 2.2, 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 except what a user gazes at intentionally, a point that swerves from the estimated gaze trajectory can be easily detected as illustrated in Fig. 4.

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 at 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:

$$\boldsymbol{F} = \begin{pmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & 0 & \Delta T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{4}$$

$$\boldsymbol{G} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}^\top, \tag{5}$$

$$\boldsymbol{H} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \tag{6}$$

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:

$$\boldsymbol{K}_i = \tilde{\boldsymbol{P}}_i \boldsymbol{H}^T (\boldsymbol{I}_{2\times 2} + \boldsymbol{H}\tilde{\boldsymbol{P}}_i \boldsymbol{H}^T)^{-1}, \tag{7}$$

$$\tilde{\boldsymbol{x}}_{i+1} = \boldsymbol{F}\{\tilde{\boldsymbol{x}}_i + \boldsymbol{K}_i(\boldsymbol{y}_i - \boldsymbol{H}\tilde{\boldsymbol{x}}_i)\}, \tag{8}$$

$$\tilde{\boldsymbol{P}}_{i+1} = \boldsymbol{F}(\tilde{\boldsymbol{P}}_i - \boldsymbol{K}_i \boldsymbol{H}\tilde{\boldsymbol{P}}_i)\boldsymbol{F}^T + \frac{\sigma_w^2}{\sigma_v^2}\Lambda, \tag{9}$$

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 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 removed from the group of the observed gaze points. With this procedure, outliers in the observed gaze points are eliminated.

2.4 Extracting a Gaze Region using the History of Gaze Points

2.4.1 Edge Detection and Elimination based on the Gaze Trajectory

For image segmentation, the boundary line between observed objects provides useful information. In the observed image, the boundary line can be detected by edge detection. We, therefore, apply the sobel operator to the observed image for edge detection and utilize the generated edge image for region extension (will be mentioned in Section 2.4.3).

The detected edge lines, however, includes not only the boundary line between objects but also various factors, 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, edge lines between gaze points can be eliminated.

Edge elimination is practically implemented as follows (Fig. 5). The sobel operator is applied to the observed image. In addition to a constant threshold T_b, the trajectory of gaze points should affect how to binarize the output of the sobel operator. Let 1) E_i and E_{i+1} denote two subsequential gaze points and 2) d_s denote the distance between E_i and E_{i+1}. A threshold for binarization at $\vec{B}$ (represented by T_e) is, then, determined by the following equation:

$$T_e = T_b + \frac{T_{gain}}{d_s \times d_b}, \tag{10}$$

where d_b and T_{gain} denote the distance between $\vec{B}$ and $\overline{E_i E}_{i+1}$ a predefined constant, respectively. With this thresholding, as the distance between $\vec{B}$ and the trajectory of gaze points becomes shorter, the threshold at $\vec{B}$ becomes larger. That is, edge lines are suppressed near the trajectory of gaze points.

2.4.2 Generating Initial Regions

All gaze points are divided into several groups. We call these groups *gp-groups*. A convex hull of the gaze points in each gp-group is regarded as an initial region.

When the gaze points are divided into gp-groups, we should consider the following problems:

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

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

Color information RGB color information is represented as a 3D vector in the RGB space. The angle 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. This angle θ is represented by the following equation:

$$\theta = \cos^{-1}\left(\frac{\boldsymbol{a} \cdot \boldsymbol{b}}{\mid \boldsymbol{a} \mid\mid \; vctb \mid}\right), \tag{11}$$

where $\boldsymbol{a}$ and $\boldsymbol{b}$ denote two RGB vectors.

Spatial distribution The geometric information of a gp-group can be represented by the centroid of these gaze points. The distance d_e between this centroid and each gaze point is evaluated to determine whether or not this gaze point is classified into this gp-group.

$$d_e = \sqrt{(C_x - x_m)^2 + (C_y - y_m)^2}, \tag{12}$$

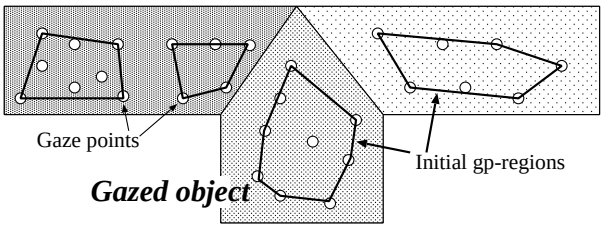


Figure 6: Generated initial gp-regions.

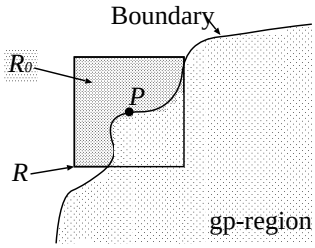


Figure 7: Region extension.

where (C_x, C_y) and (x_m, y_m) denote the centroid and the position of a gaze point observed at m.

Temporal trajectory The geometric information of gaze points consists of not only their spatial distribution but also their temporal sequence. This temporal information is represented by the length of the trajectory, d_b, represented by the following equation:

$$d_b = \sum_{i=l}^{m-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}, \tag{13}$$

where (x_i, y_i) and (x_{i+1}, y_{i+1}) denote the gaze points lastly classified into a gp-group and newly evaluated whether or not classified into this gp-group, respectively.

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

$$C \quad = \quad \alpha \times \cos\theta + \beta \times d_e + \gamma \times d_b, \tag{14}$$

where *alpha*, *beta* and *gamma* denote predefined weighted-constants.

Based on the above idea, we design how to segment all gaze points as follows:

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

Step 2 Suppose that N gp-groups of gaze points have been generated. The correlations $C_{1,\cdots,N}$ between these N gp-groups and (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 except for (x_0, y_0).

After all gaze points are divided into gp-groups, a region of each gp-group should be generated for region extension. 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 [5]:

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

Step 2 Arbitrary three gaze points are selected for generating a triangle that is regarded as an initial convex 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. These point and side are denoted by P_d and L_d, respectively.

Step 4 The two line-segments between the vertices and P_d selected in Step 3 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 6 shows an example of generated initial gp-regions.

2.4.3 Region Extension

The area of each gp-region is extended to estimate the region of a gazed object. Our extension method is based on the image segmentation method[2] that takes into account edge lines and color distribution.

The original method While [2] has 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.

Table 1: Quantitative evaluation.

	$true\text{-}positive(\%)$	$faulse\text{-}positive(\%)$
$blackboard$	95.4	9.5
$backpack$	65.7	0.0
$backpackandsofa$	88.9	0.0

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 a part of a straight line.

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

Let R is 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. If the following two conditions are satisfied, R_0 is added to the gp-region.

Condition 1 There is not any straight edge line in R_0. To evaluate whether or not an edge line exists in R_0, the following value p is calculated:

$$p = \frac{\sum(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum(X_i - \bar{X})^2}\sqrt{\sum(Y_i - \bar{Y})^2}}, \tag{15}$$

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 less than a predefined threshold, it is considered that there is not any 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.

$$T = \frac{D_{mean}}{D} \times T_0, \tag{16}$$

$$\frac{\mid \sigma_0 - \sigma \mid}{\sigma} \leq T, \tag{17}$$

where 1) σ and σ_0 denote the standard deviations of RGB values in R and R_0, respectively, 2) D_{mean} denotes the average of the distances between the centroid of the gaze points in the gp-region and these gaze points, 3) D denotes the distance between P and the centroid of the gaze points in the gp-region, and 3) T_0 denotes the base 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. This region is regarded as the region of a gazed object.

2.5 Experimental Results

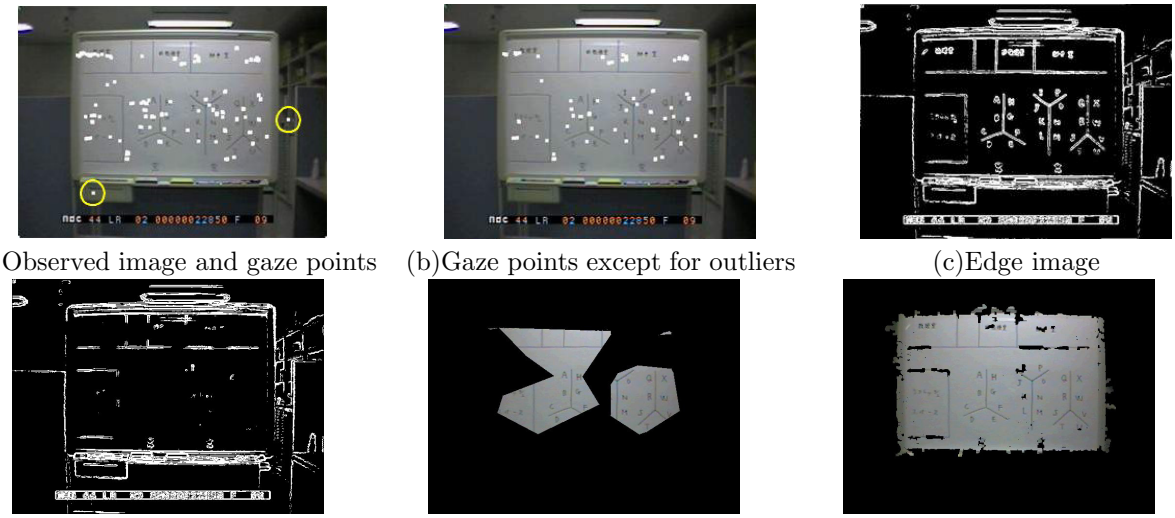
We conducted experiments to verify the effectiveness of the proposed system. In these experiments, a user gazed at the following objects:

- A blackboard.
- A backpack on a sofa.
- 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. 8, 9 and 10. The size of each image is 640 × 480 [pixel]. From these results, we can confirm that the proposed system can correctly extract a gazed object depending on the user's attention.

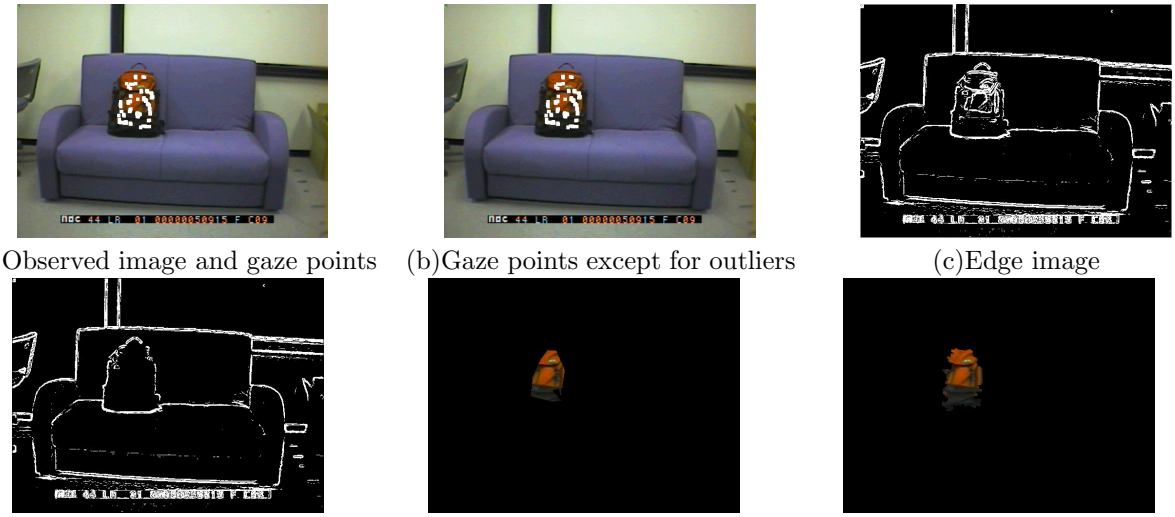
We also evaluated the quantitative performance of the proposed system by comparing the extracted result and the ground truth given by a user (Table 1). While the true-positive rate in the case of gazing at the backpack is low, all other results show the effectiveness of the proposed system.



(a)Observed image and gaze points (b)Gaze points except for outliers (c)Edge image

(d)Result of edge elimination (e)Initial gp-regions (f)Extracted result

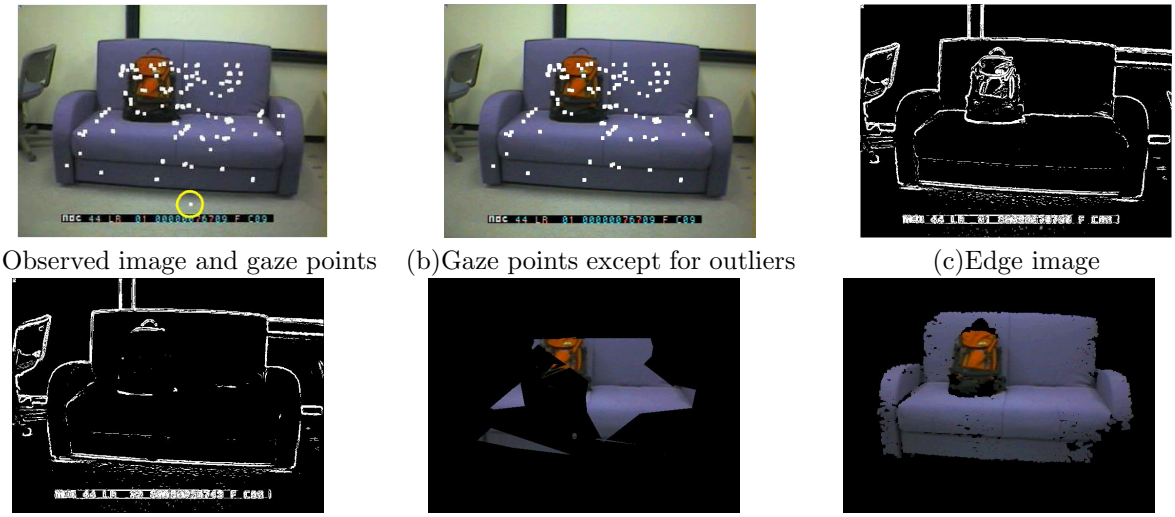
Figure 8: Experimental results: a blackboard.



(a)Observed image and gaze points (b)Gaze points except for outliers (c)Edge image

(d)Result of edge elimination (e)Initial gp-regions (f)Extracted result

Figure 9: Experimental results: a backpack.



(a)Observed image and gaze points (b)Gaze points except for outliers (c)Edge image

(d)Result of edge elimination (e)Initial gp-regions (f)Extracted result

Figure 10: Experimental results: a backpack and a sofa.

3 Estimating a 3D Gaze Position in an Environment

In this section, we propose a wearable system that can estimate a 3D position of a gaze point by measuring multiple binocular view lines. Although a 3D position is very simple information, it can be a basic factor to express various complex operations by combining multiple 3D positions:

Example 1 Select an object by specifying its position.

Example 2 Give a 3D distance by specifying its beginning and end positions.

Example 3 Move a mobile robot by specifying the goal position.

Several methods for pointing a 3D position in a real world have been proposed. A lot of works about recognizing a finger pointing have been studied (see [6], for example). Pointing by a 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 unknown.
2) A user has to stop his/her task while performing a gesture.

These problems can be avoided by employing gaze directions; 1) the direction of a gaze direction can be determined uniquely and 2) while a user performs another task, he/she can gaze at a 3D position.

To estimate the gaze position only from the user's binocular view lines, the triangulation can be employed as people always do. In this method, however, since the inter-ocular length is short and all measured view lines include errors, the estimated result is not reliable. Accordingly,we utilize view lines observed at multiple head positions in order to improve the accuracy of the estimated result; a user moves his/her head freely while gazing at a 3D position. We propose the stochastic algorithm, where each view line is described as a cone-shaped probability density distribution and the reliability of each gaze is considered. Experimental results demonstrate the effectiveness of the proposed algorithm.

3.1 Binocular view lines tracker

This system measures the view line of a user with the EMR-8 employed in the region extraction system proposed in Section 2.

3.1.1 Calibration

As mentioned in Section 2.1, the view direction measured by the EMR-8 is overlayed as a 2D position on the image observed by the view camera. To overlay a point representing a view direction on the observed image, the correspondence between the view direction and the 2D image coordinates is needed. In the EMR-8, the correspondence between them is directly computed because it is difficult to obtain the relative geometric configuration of the camera and the eyeballs. To calculate the correspondence, a flat plane in the environment (e.g., a wall) is used. While a user looks toward a wall, the view camera also observes it. Note that the wall has to be perpendicular to the view axis of the camera. Nine points are, then, superimposed in the observed image by the EMR-8. Suppose their positions in the 2D image coordinate $(X_i, Y_i)\,(i = 0, \cdots, 8)$ are known. All the points are projected to the wall in the real environment[1] , and the user gazes at each projected point in turn. Then, the 3D direction of each binocular view line $(x_i, y_i)\,(i = 0, \cdots, 8)$ is measured by the EMR-8. These values are fulfilling the following equations:

$$X_i = a_0 + a_1 x_i + a_2 y_i + a_3 x_i^2 + a_4 x_i y_i + a_5 y_i^2, \tag{18}$$

$$Y_i = b_0 + b_1 x_i + b_2 y_i + b_3 x_i^2 + b_4 x_i y_i + b_5 y_i^2, \tag{19}$$

where a_i, $b_i (i = 0, \cdots, 5)$ are unknown constants. These simultaneous equations are solved to calculate a_i, b_i. After a_i, b_i are obtained, the EMR-8 is able to overlay the view direction on the camera image rightly.

3.1.2 Error reduction

To estimate the user's gaze position precisely, it is inevitable to measure the view directions in accuracy. This section describes how to acquire the accurate results.

Lens undistortion of the view camera Generally an image observed by a camera is distorted because of its lens so that (X_i, Y_i) do not keep the correspondence under perspective projection. This distortion, therefore, breaks Equations (18) and (19). To obtain the image under perspective projection, we apply the *Tsai method*[8] for camera calibration to the view camera image.

Geometric configuration If the view axis of the camera is not perpendicular to the wall, Equations (18) and (19) are not sufficient. To improve the accuracy of the observed (X_i, Y_i), we draw 2×2 square grid points on the wall; by adjusting the posture of the view camera so that these points are observed as a square, the user can confirm whether or not the view axis is perpendicular to the wall.

[1] For example, a laser pointer is utilized to project nine points onto a wall.

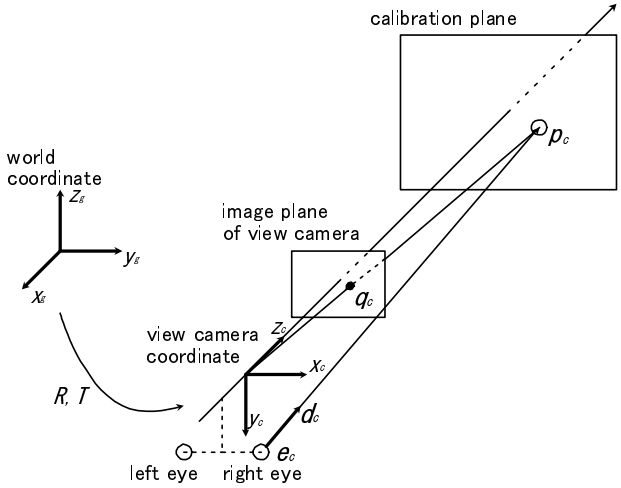


Figure 11: Representation of a view line.

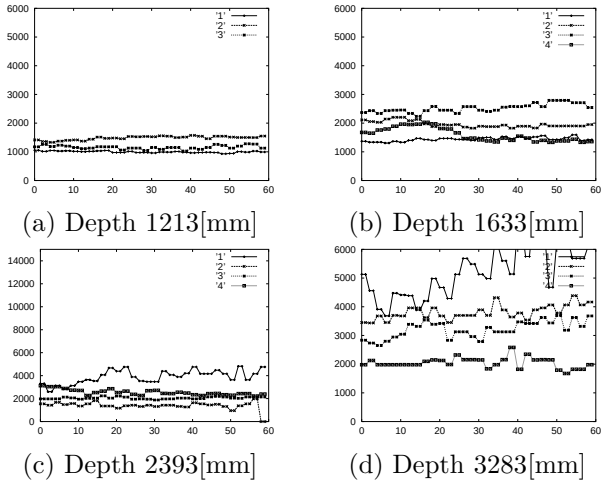


Figure 12: Depths estimated from a pair of binocular view lines.

Error reduction With some experiments, we confirmed the effect of these methods for error reduction. Without any contrivance, the average of view direction errors was 1.413 degree, and with the error reduction the average was 0.917 degree.

3.2 Representation of the view line

3.2.1 Formulation of a view line

In this section, we show the representation of a view line in the view camera coordinate with Figure 11. There is a line which indicates a view direction of a eyeball at $\boldsymbol{e}_c$. The intersection of the line and the wall for calibration (calibration plane) is indicated by $\boldsymbol{p}_c$. The point $\boldsymbol{p}_c$ is inversely projected to $\boldsymbol{q}_c$ on the image plane of the view camera. As referred in Section 3.1, the output information of the EMR-8 is $\boldsymbol{q}_c$. Then, by using the distance between the projection center of the camera and the calibration plane l_c and the focal length of the camera l_i, $\boldsymbol{p}_c$ is represented as $\boldsymbol{p}_c = \frac{l_c}{l_i}\boldsymbol{q}_c$. The view line is defined as a line from $\boldsymbol{e}_c$ to $\boldsymbol{p}_c$. The unit vector of this direction is represented as $\boldsymbol{d}_c = \frac{\boldsymbol{p}_c - \boldsymbol{e}_c}{|\boldsymbol{p}_c - \boldsymbol{e}_c|}$. With an arbitrary variant m, the view line can be represented as $\boldsymbol{e}_c + m\boldsymbol{d}_c$. It is represented as $(\boldsymbol{e}_c, \boldsymbol{d}_c)$ afterward.

3.2.2 Integration of multiple view lines

To integrate information of multiple view lines observed at multiple head positions, each view line should be described in the world coordinate system. Suppose that both 3D position and posture of a user's head are known[2] . Each head position is described by the camera's translation 3D vector and rotation 3×3 matrix ($\boldsymbol{T}$ and R, respectively) as shown in Figure 11. Let $\boldsymbol{e}_g$ and $\boldsymbol{d}_g$ denote the eyeball position in the world coordinate system and the directional vector of its view line. $\boldsymbol{e}_c$ and $\boldsymbol{d}_c$ is represented with these vectors.

$$\boldsymbol{e}_c = R\boldsymbol{e}_g + \boldsymbol{T}, \tag{20}$$

$$\boldsymbol{d}_c = R\boldsymbol{d}_g. \tag{21}$$

Then, $\boldsymbol{e}_g$ and $\boldsymbol{d}_g$ can be calculated as follows:

$$\boldsymbol{e}_g = R^{-1}(\boldsymbol{e}_c - \boldsymbol{T}), \tag{22}$$

$$\boldsymbol{d}_g = R^{-1}\boldsymbol{d}_c. \tag{23}$$

3.3 Simple method for estimating a gaze position

3.3.1 Reconstructing 3D position using binocular view lines

As the result of the discussion in Section 3.2, all of binocular view lines are represented in the world coordinate system. In this section, we estimate the user's gaze position simply only by a pair of left and right view lines.

[2] We discussed the necessity of the method for measuring human's position and posture in Section 4.

These view lines is represented as $(\boldsymbol{e}_{gl}, \boldsymbol{d}_{gl})$, $(\boldsymbol{e}_{gr}, \boldsymbol{d}_{gr})$.

$$\boldsymbol{e}_{gl} = \begin{pmatrix} e_{xl}, \\ e_{yl}, \\ e_{zl} \end{pmatrix}, \boldsymbol{d}_{gl} = \begin{pmatrix} d_{xl}, \\ d_{yl}, \\ d_{zl} \end{pmatrix}.$$
$$\boldsymbol{e}_{gr} = \begin{pmatrix} e_{xr}, \\ e_{yr}, \\ e_{zr} \end{pmatrix}, \boldsymbol{d}_{gr} = \begin{pmatrix} d_{xr}, \\ d_{yr}, \\ d_{zr} \end{pmatrix}.$$

The equation of the left view line is

$$\frac{x - e_{xl}}{d_{xl}} = \frac{y - e_{yl}}{d_{yl}} = \frac{z - e_{zl}}{d_{zl}}. \tag{24}$$

From Equation (24), the following two equations are

$$d_{zl}x - d_{xl}z = d_{zl}e_{xl} - d_{xl}e_{zl}, \tag{25}$$
$$d_{zl}y - d_{yl}z = d_{zl}e_{yl} - d_{yl}e_{zl}. \tag{26}$$

Two equations identical to Equations (25) and (26) exist for the right view line. From these four equations, the following matrix equation is obtained:

$$\begin{pmatrix} d_{zl} & 0 & -d_{xl} \\ 0 & d_{zl} & -d_{yl} \\ d_{zr} & 0 & -d_{xr} \\ 0 & d_{zr} & -d_{yr} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} d_{zl}e_{xl} - d_{xl}e_{zl} \\ d_{zl}e_{yl} - d_{yl}e_{zl} \\ d_{zr}e_{xr} - d_{xr}e_{zr} \\ d_{zr}e_{yr} - d_{yr}e_{zr} \end{pmatrix}. \tag{27}$$

By solving Equation (27), we can estimate the 3D gaze position $(x, y, z)^\top$.

3.3.2 Experimental results

We confirmed the accuracy of the simple method explained in Section 3.3. The results are shown in Figure 12. The vertical and horizontal axes of each graph indicate the time (the interval between observations was $\frac{1}{30}$ [sec]) and z value of the reconstructed position (the z axis is identical to the optical axis of the camera), respectively. At each graph, the lines $1, \cdots, 4$ show the results of different trials. The gaze points were at the distance of 1213[mm], 1633[mm], 2393[mm] and 3283[mm] from the view camera. Since 1) the inter-ocular length is too short to reconstruct and 2) z values of the estimated gaze position include much more errors than x or y, each graph shows the z value (the distance from the camera) transitions of the estimated positions.

From these graphs, we can obtain the following observations: 1) the accuracy of the result is relatively high when the gaze point is around 1200[mm], 2) the more distant the gaze position is, the lower its estimation accuracy become, and 3) the result is unreliable when the gaze point is beyond 3000[mm]. Since it is difficult to acquire an accurate result from a single pair of view lines, multiple view lines observed at different positions are required to improve the accuracy. In what follows, we describe how to estimate the 3D gaze position by integrating multiple view lines.

3.4 Stochastic algorithm for estimating a gaze position

All of view lines include inevitable errors. For integrating multiple data with errors, the least-square-error (LSE) method is often employed.In this method, all view lines are regarded as independent information. The directions of binocular view lines, however, are actually dependent on each other. The LSE method, therefore, is not suitable for our problem.

Our algorithm is a stochastic approaches and has advantage that it can consider a reliability of each user's gaze when integrating multiple view lines. The proposed algorithm consists of 3 processes, namely the initialization, updating, and estimating processes.

3.4.1 Initialization process

In this process, a voxel space is generated, in which probability density distribution is generated and updated. This voxel space is put around the initial position that is determined by the LSE method from the view lines observed at the first several head positions. This calculation is represented as the extension of Equation (27).

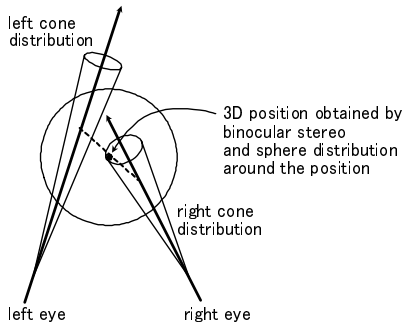


Figure 13: Distribution by a pair of binocular view lines.

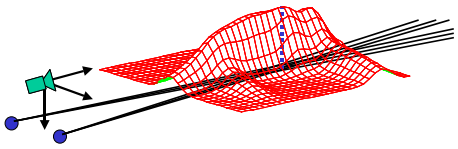


Figure 14: Unreliable probability density distribution generated by multiple cone distributions.

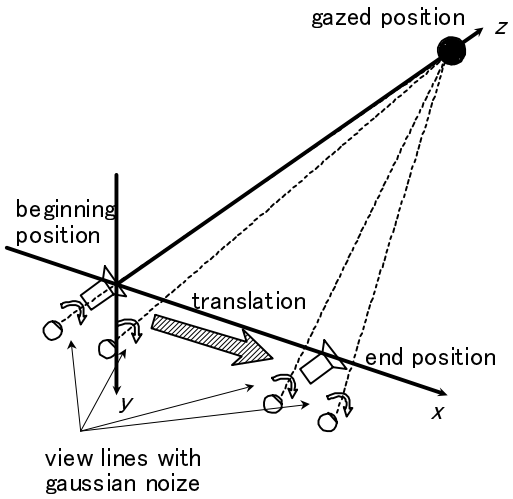


Figure 15: Environment of the simulation experiments.

3.4.2 Updating process

After generating the voxel space, the probability density distribution in it is updated at each observation moment.

Assume that each view line includes Gaussian noise. We regard each view line not as a line but as a cone distribution. This cone represents the 3D area in which the true view line probably exists; the cone is generated around the observed view line. When new binocular view lines are observed, two cones are generated and projected into the voxel space for updating the probability density distribution. Since a user's head motion may be short and the observed view lines are approximately parallel, the intersections of these view lines lengthen very long along z axis (as shown in Figure 14) and the maximum probability in the voxel space is unreliable.

To solve this problem, we focus on the reliability of the binocular view lines observed at the same moment. We suppose that the binocular view lines almost intersect each other if both two lines go towards a single gaze position. Based on this idea, the algorithm is modified as follows:

1) Calculate the 3D position P_n nearest from both binocular view lines observed at the same moment by using Equation (27).
2) Generate the Gaussian sphere distribution around P_n.
3) Calculate the intersections of each cone and the sphere and project them into the voxel space.

If the observed gaze is towards the gaze position accurately, both of the binocular view lines are close to P_n so that the density in the calculated intersections becomes quite high around P_n. If the gaze is not accurate, on the other hand, the view lines are distant from P_n enough to decrease the density in the intersections. It means that the weight of binocular view lines at each moment is variable according to the reliability of the gazing. Moreover, since the spread of each intersection is suppressed, the modified algorithm can improve the accuracy of the estimated result.

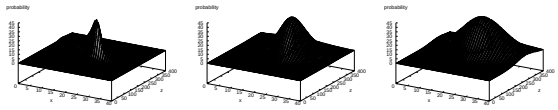
Initially, the voxel space is generated around the initial position whose reliability is very low. The probability density distribution is, therefore, easy to move out of the voxel space. To avoid this problem, we introduce the translation of the voxel space. After every update process, the system searches the maximum probability in the density distribution. Let $\boldsymbol{d}$ be the 3D vector from the voxel center to the position corresponding to the maximum probability (denoted by P_M). If P_M is out of the space H whose size is a half of the whole voxel space, the voxel space is translated $\boldsymbol{d}/2$. With this translation, P_M can be always around the centroid of the voxel space.

3.4.3 Estimating process

In this process, the 3D position corresponding to the maximum probability is considered to be the optimal solution (i.e., the 3D position gazed by a user) at each moment. The maximum probability is detected by scanning all voxels in the voxel space.

Table 2: Experimental results: the unit is [mm].

	Proposed method		LSE method	
TrueValue	Estimated	Error	Estimated	Error
$(0, 0, 1000)$	$(0, 0, 1010)$	10	$(5, 1, 951)$	49
$(0, 0, 2000)$	$(0, 0, 2010)$	10	$(18, 5, 1653)$	348
$(0, 0, 3000)$	$(0, 0, 3010)$	10	$(33, 11, 2031)$	967
$(0, 0, 4000)$	$(0, 0, 3980)$	20	$(47, 16, 2156)$	1844
$(0, 0, 5000)$	$(0, 0, 5010)$	10	$(58, 21, 2128)$	2873



(a) (0,0,1000) (b) (0,0,3000) (c) (0,0,5000)

Figure 16: $y = 0$ sections of 3D density distribution.

3.5 Experiments

3.5.1 Simulation experiments

To investigate the effectiveness of the proposed algorithm in the case that accurate user's head positions and directions are known, we conducted some simulation experiments. Figure 15 shows the experimental environment. In these simulations, the user moved his/her head in 20[cm] to the right[3] . While the user moved its head, the binocular view lines were measured 200 times at 10[mm] intervals. Each view line involves Gaussian noise whose standard deviation was 1.0 degree according to the actual view line's noise (see Section 3.1).

The results estimated by the proposed algorithm are shown in Table 2. For comparison, the results estimated by the LSE method is also shown. While the errors of the latter got larger as the distance to the gaze position became larger, the errors of the proposed algorithm remained very small.

The graphs of 3D density distributions in $y = 0$ plane are shown in Figure 16. The more distant the gaze position is, the less clear the maximal value of the distribution become. That is, the reliability gets lower as the gaze position gets more distant.

3.5.2 Experiments using the EMR-8

Next, we conducted experiments using the EMR-8 in a real environment. In these experiments, while the user's head was fixed, the gaze target moved. This situation is similar with one that the user gazes at a stationary point while moving his/her head. We, therefore, could acquire the positions and postures of the user by measuring the 3D position of the moving target instead of the information of his/her head.

Figure 17 shows several experimental results, in which only 3D depth (i.e., z value) is evaluated; errors along x and y axes are neglectable in contrast to that of z axis. The horizontal and vertical axis indicate the number of observations and z value of the position, respectively.

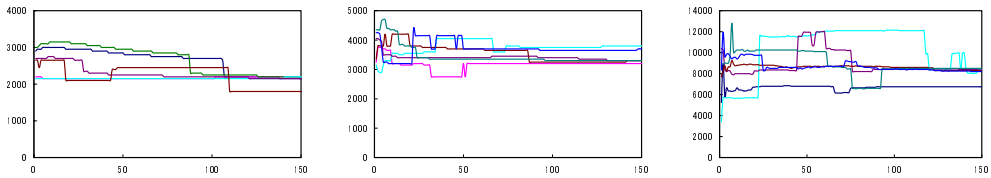
From these results, we can obtain the following considerations: 1) although the accuracy becomes better as the number of observations increases, it converges at about 100 observations and 2) the limit of the 3D reconstruction along z axis gets longer compared with the simple algorithm using binocular view lines described in Section 3.3.

4 Concluding Remarks

This paper presented two wearable vision interfaces, namely the object registration/retrieval and virtual tablet.

Extracting a gaze region from an observed image The system integrates the temporal sequence of gaze points and extract the region of an object gazed by a user.

Following are the future works:



(a) 2099[mm] (b) 3284[mm] (c) 4619[mm]

Figure 17: Experimental results: estimated 3D depth.

[3] 20[cm] is considered to be the maximal length of a human's spontaneous motion.

Implementation in real time Since several computations for region extension spend long time, the proposed algorithm cannot work in real time.

Extraction of a moving object To extract a moving object, all gaze points has to be translated to a single image based on image correlation or other criteria.

Estimating a 3D gaze position in the environment The system utilizes the view lines observed at multiple head positions and estimates the 3D gaze position by integrating them.

Following are the future works:

Implementation in real time In the current implementation, the algorithm runs very slowly because the update and the scanning of the voxel space cost a lot of time.

Tracking a user's head A method for tracking a user's head is required. We are planning to employ factorization[9] or the one that utilize stereo vision.

Gazing at a moving point The algorithm is implemented as an iterative method so that it can cope with a dynamic motion of a gaze position.

References

[1] M. Kidode: "Design and Implementation of Wearable Information Playing Station", in Proc. of *1st CREST Workshop on Advanced Computing and Communicating Techniques for Wearable Information Playing*, pp.1–5, 2002.

[2] H. Koh, H. Suzuki, and J. Toriwaki: "A segmentation method based on region information and edge information", Transaction of The Institute of Electronics, Information and Communi cation Engineers D-II, Vol.J74-DII, No.12, pp.1651–1660, 1991. (written in Japanese)

[3] M. Kass, A. Wiktin, D. Terzopulos: "SNAKES: Active Contour Models", in Proc. of 1st International Conference on Computer Vision, pp.259–268, 1987.

[4] 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.

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

[6] 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.

[7] The Vision Society of Japan: "Vision Information Processing Handbook", 2000.

[8] R.Y.Tsai: "A efficient and accurate camera calibration technique for 3D machine vision", in Proc. of Computer Vision and Pattern Recognition 86, pp.364–374, 1986.

[9] Conrad J. Poelman and Takeo Kanade: "A Paraperspective Factorization Method for Shape and Motion Recovery", CMU-CS-93-219, 1993.