Estimation of 3D Gazed Position Using View Lines

Ikuhisa Mitsugami, Norimichi Ukita and Masatsugu Kidode
Nara Institute of Science and Technology
Graduate School of Information Science
Takayama-cho 8916-5, Ikoma-shi, Nara, 630-0192 Japan
{ikuhi-mi, ukita, kidode}@is.aist-nara.ac.jp

Abstract

We propose a new wearable system that can estimate the 3D position of a gazed point by measuring multiple binocular view lines. In principle, 3D measurement is possible by the triangulation of binocular view lines. However, it is difficult to measure these lines accurately with a device for eye tracking, because of errors caused by 1) difficulty in calibrating the device and 2) the limitation that a human cannot gaze very accurately at a distant point. Concerning 1), the accuracy of calibration can be improved by considering the optical properties of a camera in the device. To solve 2), we propose a stochastic algorithm that determines a gazed 3D position by integrating information of view lines observed at multiple head positions. We validated the effectiveness of the proposed algorithm experimentally.

Keywords: *eye tracking, gazed position measurement, camera calibration, 3D probability density*

1 Introduction

In recent years, a variety of robots and other hardware have been developed. As their impact increases, they will provide more effective and advanced support for our daily lives, but at the same time we will need to deploy a wider range of operating commands to take full advantage of this increasingly sophisticated hardware. There is a wide range of information that a human user may need to transmit to such hardware for efficient operation, and in this research we focus on 3D positional information. This information is important to operate robots and other hardware intuitively and simply, because they often work in the human 3D environment. Although 3D positions are very simple information, consisting of only 3 parameters in 3D space, it is possible by combining them to express more complicated information, such as directions, lengths, areas, and so on.

Several methods for determining 3D position in the real world have been considered, many of which are based on finger-pointing (see [1], for example). Pointing by the finger has the following problems:

1) Even if the system can determine the exact 3D position of a user's fingertip, it is difficult to estimate the 3D position of an object being pointed at by the finger because the 3D direction of the finger is ambiguous; it changes depending on situations and individuals. For example, is 3D direction measured to the user's fingertip from the eye, the elbow, or the base of the finger?
2) The user has to interrupt his/her task while performing a gesture. It is desirable that the user can indicate a 3D position without being distracted from his/her task.

Considering the latter problem, a method using the user's view lines, which is a kind of gesture recognition, is very effective. Even while the user is performing a task, he/she can gaze at a 3D position. Moreover, since the direction of a view line can be determined uniquely, the former problem can be avoided. Its simplicity and intuitiveness are also advantages which we focus on in this work.

When estimating the position of a gazed object, humans use not only the information of their binocular view lines but also other information and knowledge subconsciously, such as the object's size, color, and shadow. In computer systems, the latter knowledge is hard to describe and therefore the system must estimate the gazed position using only the former information. To estimate the gazed position only from the user's binocular view lines, triangulation can be employed; however, since the inter-ocular length is short and all view lines include errors, the estimated result is unreliable.

By allowing the user to move his/her head freely, we can utilize view lines obtained at multiple head positions. We propose here a stochastic algorithm in which each view line is described as a cone-shaped probability density distribution, and the reliability of each gaze is considered. Experimental results demonstrated the effectiveness of this algorithm.

2 Binocular view lines tracker

2.1 Configuration

To measure the user's view lines, we used an EMR-8 eyemark recorder (NAC Inc.), which consists of a view

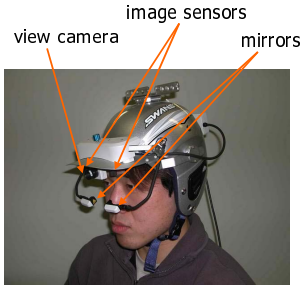


Figure 1. Binocular eyemark recorder EMR-8.

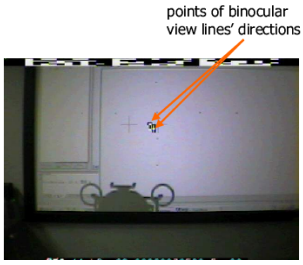


Figure 2. Output image of EMR-8.

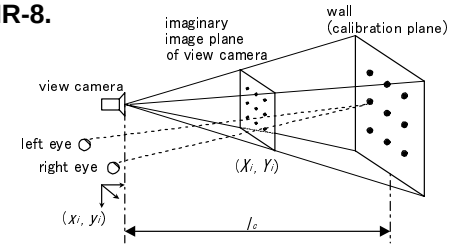


Figure 3. Calibration of EMR-8.

camera, two image sensors, and two mirrors (Figure 1). The user's eyes are monitored by the mirrors and image sensors, and their directions (hereafter called *view directions*) are measured by the corneal reflection-pupil center method [2]. The view camera is placed between the eyes, obtaining images of the user's view in real time. The view directions at each observation time are represented as points on the image obtained by the view camera and overlaid on it (Figure 2).

2.2 Calibration

To overlay the points representing the view directions onto the image observed by the view camera, correspondence between the view direction and the 2D image coordinates is needed. In the EMR-8, this correspondence 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 the user looks towards the wall, the view camera also observes the wall. Note that the wall has to be perpendicular to the view axis of the camera. Nine points are then superimposed on the observed image by the EMR-8. Suppose their positions in the 2D image coordinates $(X_i, Y_i)\,(i = 0, \cdots, 8)$ are known. All the points are projected onto the wall in the real environment, for example by a laser pointer, and the user gazes at each projected point in turn. Next, the 3D direction of each binocular view line $(x_i, y_i)\,(i = 0, \cdots, 8)$ is measured (Figure 3) by the EMR-8. These values are derived from the following equations:

$$X_i = a_0 + a_1 x_i + a_2 y_i + a_3 x_i^2 + a_4 x_i y_i + a_5 y_i^2, \quad (1)$$

$$Y_i = b_0 + b_1 x_i + b_2 y_i + b_3 x_i^2 + b_4 x_i y_i + b_5 y_i^2, \quad (2)$$

where $a_i,\ b_i (i = 0, \cdots, 5)$ are unknown constants. These simultaneous equations are solved to calculate $a_i,\ b_i$. After

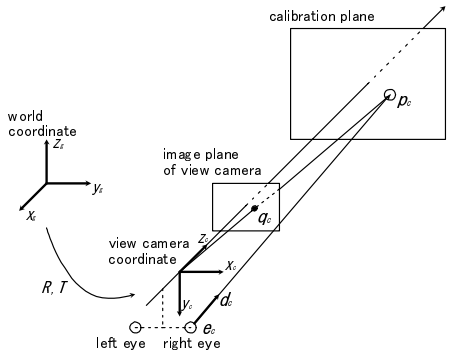


Figure 4. Representation of a view line.

$a_i,\ b_i$ are obtained, the EMR-8 is able to correctly overlay the view direction onto the camera image.

2.3 Error reduction

To estimate the user's gazed position precisely, it is essential to measure the view directions accurately. This section describes how to acquire accurate measurements.

2.3.1 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 maintain correspondence under perspective projection. This distortion, therefore, breaks Equations (1) and (2). To obtain an image under perspective projection, we apply the Tsai method [3] for camera calibration to the image.

2.3.2 Establishing the wall perpendicularity.

If the view axis of the camera is not perpendicular to the wall, Equations (1) and (2) 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 orientation 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.

2.3.3 Effect of error reduction.

We confirmed the effect of these methods for error reduction experimentally. Without any correction, the average view direction error was 1.4 degrees, whereas with error reduction the average was 0.9 degrees.

3 Representation of the view line

3.1 Formulation of a view line

In this section, we describe the representation of a view line in the view camera coordinate system as shown in Figure 4. A line which indicates the view direction of an eyeball originates 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 back-projected to $\boldsymbol{q}_c$ on the image plane of the view camera. As mentioned in Section 2, the output information of the EMR-8 is $\boldsymbol{q}_c$. Therefore, 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 described as follows:

$$\boldsymbol{p}_c = \frac{l_c}{l_i}\boldsymbol{q}_c. \tag{3}$$

The view line is defined as the line from $\boldsymbol{e}_c$ to $\boldsymbol{p}_c$. The unit vector of this direction is described by

$$\boldsymbol{d}_c = \frac{\boldsymbol{p}_c - \boldsymbol{e}_c}{|\boldsymbol{p}_c - \boldsymbol{e}_c|}. \tag{4}$$

With an arbitrary variant m, the view line can be represented as $\boldsymbol{e}_c + m\boldsymbol{d}_c$. It is represented as $(\boldsymbol{e}_c, \boldsymbol{d}_c)$ below.

3.2 Integration of multiple view lines

To integrate information from multiple view lines observed at multiple head positions, each view line should be described in the world coordinate system. Suppose that both the 3D position and the orientation of a user's head are known[1]. 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 4. Let $\boldsymbol{e}_g$ denote the eyeball's position in the world coordinate system and $\boldsymbol{d}_g$ the directional vector of its view line. $\boldsymbol{e}_c$ and $\boldsymbol{d}_c$ are represented by these vectors:

$$\boldsymbol{e}_c = R\boldsymbol{e}_g + \boldsymbol{T}, \tag{5}$$
$$\boldsymbol{d}_c = R\boldsymbol{d}_g. \tag{6}$$

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

$$\boldsymbol{e}_g = R^{-1}(\boldsymbol{e}_c - \boldsymbol{T}), \tag{7}$$
$$\boldsymbol{d}_g = R^{-1}\boldsymbol{d}_c. \tag{8}$$

4 Simple method for estimating a gazed position

4.1 Reconstructing 3D position using binocular view lines

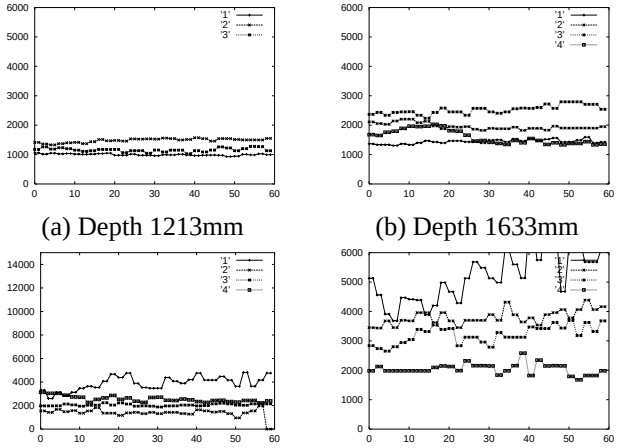
As discussed in Section 3, all binocular view lines are represented in the world coordinate system. In this section, we estimate the user's gazed position simply, using only a pair of left and right view lines. These view lines are described as $(\boldsymbol{e}_{gl}, \boldsymbol{d}_{gl})$, $(\boldsymbol{e}_{gr}, \boldsymbol{d}_{gr})$:

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

The equation of the left view line is

$$\frac{x - e_{xl}}{d_{xl}} = \frac{y - e_{yl}}{d_{yl}} = \frac{z - e_{zl}}{d_{zl}}. \tag{9}$$

[1]We discuss the necessity of a method for measuring the user's head position and orientation in Section 7.



(a) Depth 1213mm (b) Depth 1633mm

(c) Depth 2393mm (d) Depth 3283mm

Figure 5. Depths estimated from a pair of binocular view lines.

From Equation (9), the following two equations are derived:

$$d_{zl}x - d_{xl}z = d_{zl}e_{xl} - d_{xl}e_{zl}, \tag{10}$$
$$d_{zl}y - d_{yl}z = d_{zl}e_{yl} - d_{yl}e_{zl}. \tag{11}$$

Two equations analogous to Equations (10) and (11) exist for the right view line. From these four equations, the following matrix equation is obtained:

$$\begin{pmatrix} d_{zl} & 0 & -d_{xl} \\ 0 & d_{zl} & -d_{yl} \\ d_{zr} & 0 & -d_{xr} \\ 0 & d_{zr} & -d_{yr} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} d_{zl}e_{xl} - d_{xl}e_{zl} \\ d_{zl}e_{yl} - d_{yl}e_{zl} \\ d_{zr}e_{xr} - d_{xr}e_{zr} \\ d_{zr}e_{yr} - d_{yr}e_{zr} \end{pmatrix}. \tag{12}$$

By solving Equation (12), we can estimate the 3D gazed position $(x, y, z)^T$.

4.2 Experimental results

We confirmed the accuracy of the simple method explained in Section 4.1. The results are shown in Figure 5. 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. In each graph, lines 1 to 4 show the results of different trials. The gazed points were 1213mm, 1633mm, 2393mm and 3283mm from the view camera. Since 1) the inter-ocular length is too short to reconstruct, and 2) z values of the estimated gazed position contain many 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 make the following observations: 1) the accuracy of the result is relatively high when the gaze point is around 1200mm; 2) the more distant the gazed position, the lower its estimated accuracy becomes; and 3) beyond 3000mm, the result is unreliable. Since it is

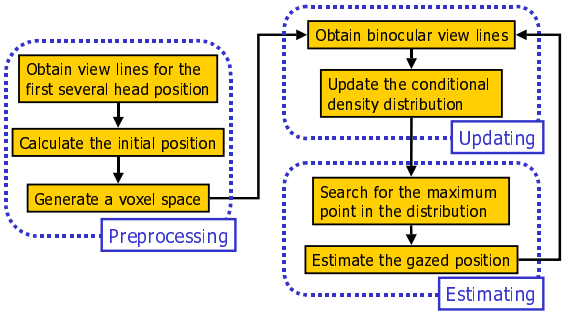


Figure 6. Flowchart of the proposed algorithm.

very difficult to acquire a satisfactory estimation from a single pair of view lines, more view lines observed at different positions are required to improve the accuracy of the reconstruction. We therefore propose, below, a sophisticated method for estimating the 3D gazed position by integrating multiple view lines.

5 Stochastic algorithm for estimating a gazed position

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

Our algorithm is a kind of stochastic approach and has the advantage that it can consider the reliability of each user's gaze when integrating multiple view lines. The proposed algorithm consists of 3 processes: preprocessing, updating, and estimating (Figure 6). Each of these is described below.

5.1 Preprocessing

In this process, a voxel space is generated in which probability density distribution is generated and updated. This voxel space is centered around the initial position determined by the LSE method from the view lines observed at the user's first several head positions. This calculation is represented as the extension of Equation (12).

5.2 Updating

After generating the voxel space, the probability density distribution within 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

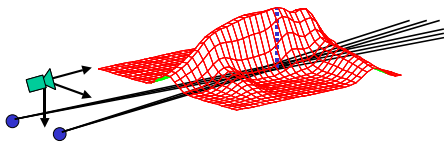


Figure 7. Unreliable probability density distribution generated by multiple cone distributions.

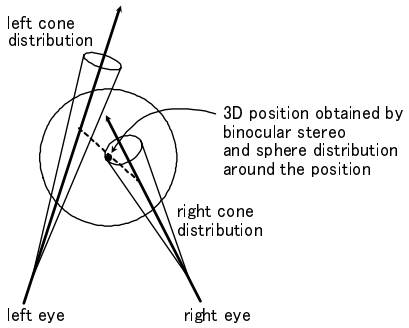


Figure 8. Distribution of a pair of binocular view lines.

for updating the probability density distribution. Since the user's head motion may be small and the observed view lines are approximately parallel, the intersections of these view lines lengthen greatly along the z axis (as shown in Figure 7) and the maximum probability in the voxel space is unreliable.

To solve this problem, we focus on the reliability of binocular view lines observed at the same moment. We suppose that the binocular view lines almost intersect each other if both lines point towards a single gazed position. Based on this supposition, the algorithm is modified as follows (Figure 8):

1) Calculate the 3D position P_n nearest to both binocular view lines observed at the same moment, by using Equation (12).
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 pointing accurately towards the gazed position, 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 enough from P_n to decrease the density in the intersections. This means that the weight of binocular view lines at each moment varies according to the reliability of the gaze. 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 a position whose reliability is very low. The probability density distribution may, therefore, easily move out of the voxel space. To avoid this problem, we introduce a translation

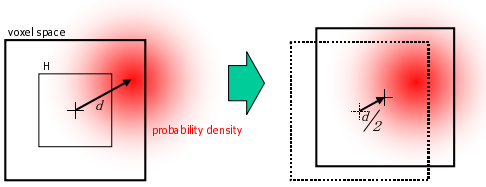


Figure 9. Translation of the voxel space.

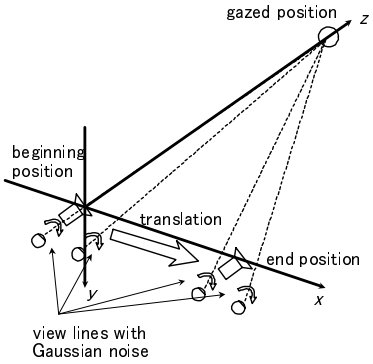


Figure 10. Environment of the simulation experiments.

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 outside the space H, whose size is half of the whole voxel space, the voxel space is translated to $\boldsymbol{d}/2$ (Figure 9). With this translation, P_M should always be near the center of the voxel space.

5.3 Estimating

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 the user) at each moment. The maximum probability is detected by scanning all voxels in the voxel space.

6 Experiments

6.1 Simulation experiments

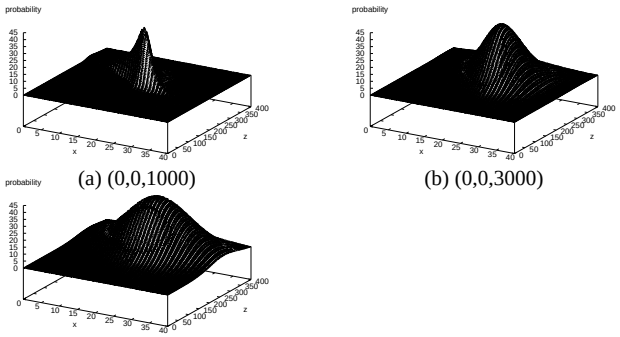
To investigate the effectiveness of the proposed algorithm when the user's head positions and directions are known accurately, we conducted simulation experiments. Figure 10 shows the experimental environment. In these simulations, the user moved his/her head 20cm to the right[2]. While the user's head was moving, the binocular view lines were measured 200 times at 1mm intervals. Each view line involved Gaussian noise, whose standard deviation was 1.0 degree according to the actual view line's noise (see Section 2).

The results estimated by the proposed algorithm are shown in Table 1. For comparison, the results estimated

[2]20cm is considered to be the maximal length of a human's spontaneous motion.

	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

Table 1. Experimental results: all values are mm.



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

(c) (0,0,5000)

Figure 11. $y = 0$ sections of 3D density distribution.

by the LSE method are also shown. While the errors of the latter increased as the distance to the gazed position became larger, the errors of the proposed algorithm remained very small.

The graphs of 3D density distributions in the $y = 0$ plane are shown in Figure 11. The more distant the gazed position is, the less clear the maximal value of the distribution becomes. That is, the reliability gets lower as the gazed position gets more distant.

6.2 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 as shown in Figure 12. This situation is similar to one in which the user gazes at a stationary point while moving his/her head. Therefore, we could acquire the positions and orientations of the user by measuring the 3D position of the moving target instead of the information about his/her head.

Figure 13 shows several experimental results in which only 3D depth (i.e., z value) was evaluated; errors along the x and y axes are negligible, in contrast to those on the z axis. The horizontal and vertical axes indicate the number of observations and z value of the position, respectively.

From these results, we can conclude the following: 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 the z axis gets longer compared with the simple algorithm using

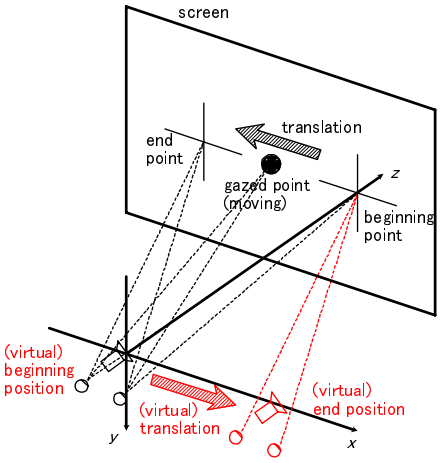
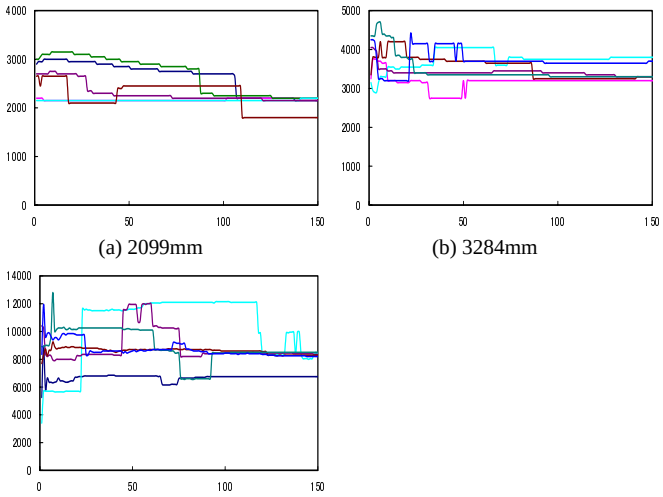


Figure 12. Environment of experiments using the EMR-8.



(a) 2099mm (b) 3284mm

(c) 4619mm

Figure 13. Experimental results: estimated 3D depth.

binocular view lines described in Section 4. With the proposed method also, however, the estimated results are unreliable. Since all the erroneous results are beyond the true position, we suspect that cone-shaped distributions widen too much, overlap with each other and generate some incorrect local maxima.

7 Concluding remarks

In this paper, we proposed a stochastic algorithm to estimate the 3D position that a user gazes at. It utilizes the view lines observed at multiple head positions, and estimates the 3D gazed position by integrating them. We confirmed the performance and effectiveness of the algorithm by experimentation.

Our planned future work is as follows:

- **Implementation in real time**
  In the current implementation, the algorithm runs very slowly because the resolution of the voxel space is so high that the updating and scanning of the voxel space cost a lot of time. It should be configured adequately according to the limit of estimation accuracy.
- **Tracking the user's head**
  Since the user's head positions and orientations are given in this paper, we need a method for head tracking to implement a system that works in the real world. Although there has been a lot of research about head tracking (see [4], for example) that is useful to our system, most of those proposals require the user to wear additional devices. In our system it is not desirable for the user to wear any devices except the EMR-8. We are planning to employ vision-based approaches such as factorization[5] or homography[6] in order to utilize the EMR-8 camera.
- **Learning from personal errors**
  There may be some errors in gazing that depend strongly on individuals and that decrease the estimation accuracy. We should study these to improve the accuracy of the estimated result.
- **Position estimation of a moving point**
  The algorithm is implemented as a stochastic and iterative method so that it can cope with dynamic motion of a gazed position. We would like to consider this extension in the future.

Acknowledgements

This research is supported by Core Research for Evolutional Science and Technology (CREST) Program "Advanced Media Technology for Everyday Living" of Japan Science and Technology Corporation (JST).

References

[1] 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.

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

[3] R.Y.Tsai: "A efficient and accurate camera calibration technique for 3D machine vision", in Proc. of *CVPR86*, pp.364–374, 1986.

[4] M. Maeda, T. Habara, T. Machida, T. Ogawa, K. Kiyokawa and H. Takemura: "Indoor Localization Methods for Wearable Mixed Reality", in Proc. of *2nd CREST Workshop on Advanced Computing and Communicating Techniques for Wearable Information Playing*, pp.62–65, 2003.

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

[6] R. Hartley and A. Zisserman: "Multiple View Geometry in Computer Vision", 2002.