A Fast Vision-based Head Tracking Method for Interactive Stereoscopic Viewing

Narongsak Putpuek, and Nopporn Chotikakamthorn

Faculty of Information Technology &
Research Center for Communications and Information Technology
King Mongkut's Institute of Technology Ladkrabang, Bangkok Thailand
(E-mail: s3067144@kmitl.ac.th)

Abstract: In this paper, the problem of a viewer's head tracking in a desktop-based interactive stereoscopic display system is considered. A fast and low-cost approach to the problem is important for such a computing environment. The system under consideration utilizes a shuttle glass for stereoscopic display. The proposed method makes use of an image taken from a single low-cost video camera. By using a simple feature extraction algorithm, the obtained points corresponding to the image of the user-worn shuttle glass are used to estimate the glass center, its local 'yaw' angle, as measured with respect to the glass center, and its global 'yaw' angle as measured with respect to the camera location. With these estimations, the stereoscopic image synthetic program utilizes those values to interactively adjust the two-view stereoscopic image pair as displayed on a computer screen. The adjustment is carried out such that the so-obtained stereoscopic picture, when viewed from a current user position, provides a close-to-real perspective and depth perception. However, because the algorithm and device used are designed for fast computation, the estimation is typically not precise enough to provide a flicker-free interactive viewing. An error concealment method is thus proposed to alleviate the problem. This concealment method should be sufficient for applications that do not require a high degree of visual realism and interaction.

Keywords: Stereoscopic Display, Computer Vision, Interactive System

1. INTRODUCTION

A stereoscopic display system presents the left and right eyes of the viewer with images from different perspective viewpoints, just as the viewer sees the visual world [1]. From these two slightly different views, the eye-brain synthesizes an image of the world with stereoscopic depth. A single, not double, image is seen since the two are fused by the mind into one. Thus, the system can provide an illusion of depth perception just like a human vision experience when viewing a real-world object. Stereoscopic display is used for a variety of applications, from games to computer-aided design. However, a standard stereoscopic display system is limited to delivering an image pair perfectly viewed only from a single view point. Changing of viewer's position does not result in perspective change of the viewed scene. Interactive stereoscopic display system provides an answer to the problem. By interactively adjusting the rendering view point of the synthetic stereoscopic image pair according to the current viewer's position, a high degree of virtual realism is obtained. To obtain the important information about current viewer's spatial position, a head tracking sensor/subsystem may be employed. Tracking of a viewer's head position is achieved by using one of these sensing technologies: electromagnetic sensor, optical sensor, as well as inertial, acoustical or mechanical sensing devices.

Use of these sensing devices is limited to high-end applications such as computer-aided design or VR-based simulation systems. For desktop applications, simpler technique is required. Examples of previous work on desktop-based virtual and augmented reality (VR/AR) systems include those of [2-3]. The Personal Space Station (PSS) is described in [4]. The system allows its user to perform 3D interactive. In this system, all interactive 3D tasks are performed directly with the hands or by using task specific input devices. The PSS is designed with three major goals: low costs, ergonomics and direct 3D interaction. The system is based on an optical-based head tracking [4] technique. Two standard FireWire cameras are used to provide tracking. In PSS system user is seated in front of the mirror which reflects the stereoscopic images of the virtual world as displayed by the monitor.

In this paper, we present a fast and low-cost optical head tracking system that uses single low-cost camera and a simple shuttle-glass object segmentation method. By exploiting the color and shape of the shuttle glass device, additional markers are not required. An algorithm is developed to provide rough estimation of the viewer's position. Error concealment method is proposed to alleviate the inaccuracy in viewer's position estimation.

The paper is organized as follows. First, the proposed interactive stereoscopic viewing system is introduced along with necessary image processing notations. Next, detail of user's head position estimation is given in Section 3. Experiment results are provided with some discussion in Section 4. A conclusion is given in Section 5.

2. VISION-BASED INTERACTIVE STEREOSCOPIC VIEWING SYSTEM

The interactive stereoscopic viewing system is shown in Fig. 1. From the figure, a single low-cost video camera is placed on top of the CRT display unit. A stereoscopic shuttle glass is used to provide scene depth information to a viewer. A viewer is free to change his/her position relative to the camera. It is assumed, however, that local head rotation about the local *z*-axis is negligible. This assumption is made to simplify the heading tracking algorithm. The assumption should be approximately met for the usage scenario under study.

From Fig. 1 a user wearing a stereoscopic shuttle glass watches a stereoscopic computer-generated picture as shown in the display unit, semi-interactively. The term 'semi-interaction' here refers to the ability of the system to allow its user to change his/her physical viewpoint (e.g., by stepping to the right/left or back/front), and the rendered content is adjusted accordingly in a stepwise manner to provide certain degree of virtual realism. To create such virtual realism, the left and right images from which the stereoscopic picture is constructed must be rendered based on a viewer position.

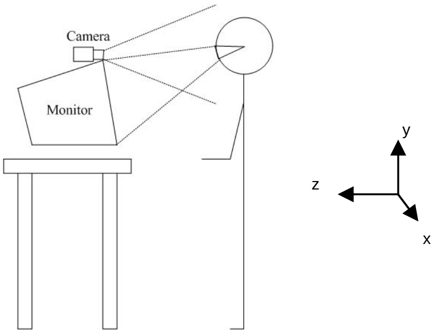

Fig.1 The interactive stereoscopic viewing system

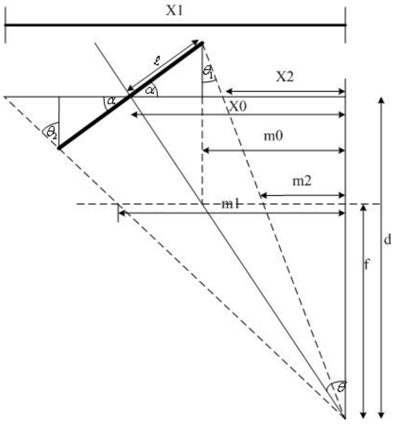

Fig. 2 Rendering of a stereoscopic image pair, based on a viewer position.

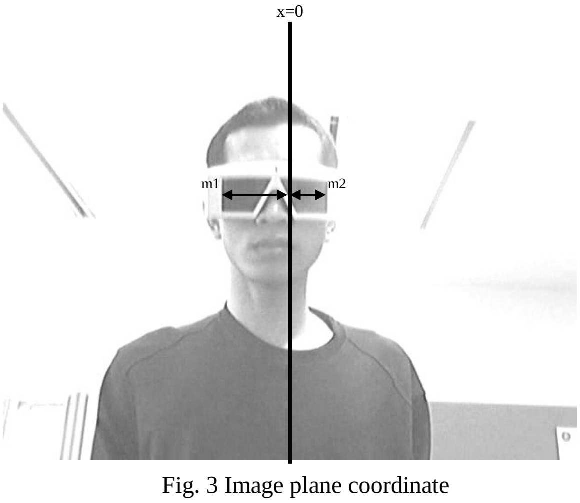

Fig. 3 Image plane coordinate

To realize such a system, estimates of a viewer's 'yaw' angles must be made available. In this paper, a simple vision-based method for estimating the global and local 'yaw' angles of a viewer's head is described. Estimations of both angular values (α and θ as shown in Fig. 2) are obtained by analyzing an image snapshot using a single low-cost video camera. Let's $I(x, y)$ be the intensity function at the (*x, y*) pixel coordinate of the gray-scale image taken by a video camera at any time instance. Note that, here the (0,0) pixel is located at the center of the image (see Fig. 3). From any snapshot $I(x, y)$, a user global nominal 'yaw' angle, as measured with respect to the video camera coordinate is first estimated from the segments of $I(x, y)$ corresponding to the left and right viewing areas of the shuttle glass. Then, estimation of the viewer's distance d is next obtained. The so-obtained estimated angle $\hat{\theta}$ and distance $\hat{d}$ are then used in making a decision on whether to change the rendering viewpoint of the stereoscopic image pair as displayed by the system's CRT.

In the next section, detail of an algorithm used to estimate θ and d is described.

3. ALGORITHM FOR HEAD'S POSITION ESTIMATION

3.1 Image segmentation and feature extraction algorithm

The first step in the head's positional estimation is to perform necessary feature extraction from the camera image snapshot. The algorithm is described as follows.

Image Segmentation and feature extraction algorithm

1. Convert $I(x, y)$ to a binary image. Segments of images areas of black pixels are then identified.
2. For each segmented black-pixel area in the binary version of $I(x, y)$, calculate the segment Center Of Mass (COM) position (as denoted by (x_c, y_c) in Fig. 4).
3. Let d_x and d_y represent the minimum width and height of the image area corresponding to either of the two glass's viewing (LCD) areas. From the two values, for each segmented black-pixel area, starting from the segment COM position, find pixels at the four boundaries of the segment (see Fig. 4). These pixel positions are denoted by $(x_i, y_i), i = 1, \cdots, 8$.
4. For each black-pixel image area, from the values of (x_i, y_i) as obtained from Step 3, determine m_{top} , m_{bottom} , m_{left} and m_{right} (see Fig. 4) from

$$m_{top} = \frac{y_2 - y_1}{x_2 - x_1} \qquad (1)$$

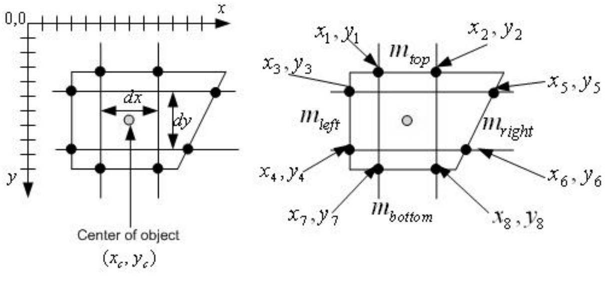

Fig. 4 Parameters corresponding to each segmented black-pixel area, in $I(x, y)$

$$m_{left} = \frac{x_4 - x_3}{y_4 - y_3} \qquad (2)$$

$$m_{right} = \frac{x_6 - x_5}{y_6 - y_5} \qquad (3)$$

$$m_{bottom} = \frac{y_8 - y_7}{x_8 - x_7} \qquad (4)$$

5. For each black-pixel image area, calculate two standard deviation values as given by the following equations.

$$\sigma_L = \sqrt{\frac{(m_{top} - \overline{m}_{left})^2 + (m_{left} - \overline{m}_{left})^2 + (m_{bottom} - \overline{m}_{left})^2}{2}} \qquad (5)$$

$$\sigma_R = \sqrt{\frac{(m_{top} - \overline{m}_{right})^2 + (m_{right} - \overline{m}_{right})^2 + (m_{bottom} - \overline{m}_{right})^2}{2}} \qquad (6)$$

where

$$\overline{m}_{left} = \frac{(m_{top} + m_{left} + m_{bottom})}{3} \qquad (7)$$

$$\overline{m}_{right} = \frac{(m_{top} + m_{right} + m_{bottom})}{3} \qquad (8)$$

6. Of the two values σ_L and σ_R, choose the smaller value and denote it by $\sigma_{\min}$.
7. Among all segmented areas, select the two black-pixel segmented areas in the binary version of $I(x, y)$ that have the smallest values of $\sigma_{\min}$. The two selected segments are then identified as corresponding to the two shuttle glass's viewing areas.

From the identified viewing areas as described, the next section details how to estimate θ and d.

3.2 Estimation of the user's relative distance from a screen

It is first assumed that the camera focal length *f* is known. From the image $I(x, y)$ obtained from the camera at any time instance, apply the feature extraction and segmentation algorithm as described above. From the so-obtained two shuttle glass's viewing areas as appeared in $I(x, y)$, the left and right borders, as well as the center point of the shuttle glass can be identified. It is assumed in the study that the user's head is oriented such that its y-axis is parallel to that of the camera, and the three projected points as mentioned are on the x-axis of the image coordinate (The latter assumption is made for ease of presentation). No assumption is made, however, regarding to the head's 'yaw' angle relative to the glass center (as denoted by α in Fig. 2). Based on these assumptions, the distances between the three points, measured from the image center along the image's x-axis, are m_0, m_1, and m_2 as shown in Fig. 2. Therefore, the three angular values in Fig. 2 can be calculated from the following equations

$$\theta = \tan^{-1}(m_0 / f) \qquad (9)$$

$$\theta_1 = \tan^{-1}(m_1 / f) \qquad (10)$$

$$\theta_2 = \tan^{-1}(m_2 / f) \qquad (11)$$

From Fig. 2, it can be derived that x_0, x_1, x_2 are related to one another by the following equations

$$x_0 = l\cos(\alpha) + l\sin(\alpha)\tan(\theta_1) + x_2 \qquad (12)$$

$$x_1 = l\cos(\alpha) + l\sin(\alpha)\tan(\theta_2) + x_0 \qquad (13)$$

where *l* is half the width of the shuttle glass. In addition, it can be verified that

$$x_i = d.m_i / f \quad , i = 0, 1, 2. \qquad (14)$$

From Eqs. (9-14), α can be obtained from

$$\alpha = \tan^{-1}(A / B) \qquad (15)$$

where

$$A = (m_0 - m_2) - (m_1 - m_0) \qquad (16)$$

$$B = (m_1 - m_0)\tan(\theta_1) - (m_0 - m_2)\tan(\theta_2) \qquad (17)$$

With α as obtained from Eq. (15), x_1 - x_2 can be obtained from Eqs. (12-13). As a result, by using Eq. (14), *d* is obtained from

$$d = f(x_1 - x_2)/(m_1 - m_2) \qquad (18)$$

Having obtained the estimates of both *d* and θ, the graphic rendering subsystem can use these values to adjust the rendering such that the left and right images match the user viewing position.

In practice, by asking a user to perform initial calibration process, the rough estimate of *f* can be obtained.

3.3 Error concealment method

Because the estimation as detailed in Section 3.2 is not precise, due to limitation in terms of the image acquisition device's accuracy, as well as the need to reduce algorithm complexity as much as possible. Therefore, truly real-time interactive system is unachievable with sufficiently low flicker. The system developed here must therefore work in a semi-interactive mode. In particular, change of rendering viewpoint is made based on the following conditions

- Change is made whenever the user's head position is sufficiently stationary. Head stationarity can be detected by calculating standard deviation of the two parameter estimates (*d* and θ) among those obtained from a few (say, 3-4) consecutive image snapshots.
- In addition, the whole viewing space is divided into discrete set of viewing areas. Only when the user's head position falls into a different viewing space that the system changes the rendering viewpoint.
- Switching between two image pairs of different viewpoints is performed gradually, by fading out the image pair corresponding to the current rendering viewpoint and fading in the image pair corresponding to the new rendering viewpoint.

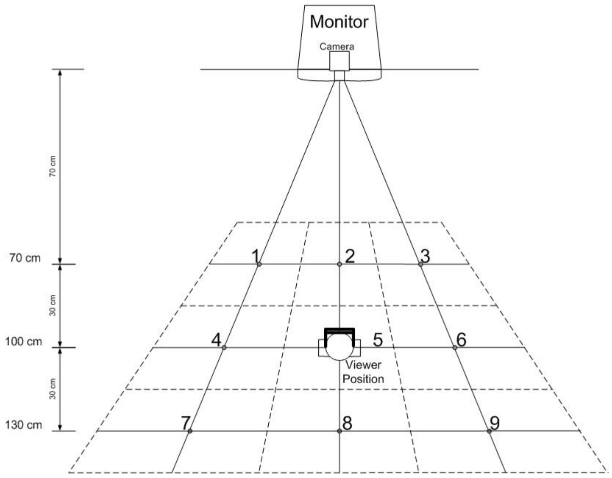

Fig. 5 Division of the whole viewing space into a few smaller viewing areas.

4. EXPERIMENT

In the experiment, the viewing space is divided into 9 smaller viewing areas. The center position of each area is shown in Fig. 5.

First, the initial calibration step was performed using sample image snapshots. Each was taken from each of 9 different viewing areas. These sample images were used to obtain the estimate of the camera focal length *f* by using Eq. (18). The value of $x_1 - x_2$ in Eq. (18) corresponds to the width of the shuttle glass as measured from a real shutter glass.

Next, for each of 9 viewer positions, 10 image snapshots were taken, and used to estimate θ and *d*. The result is show in Table 1.

The estimates of θ and *d* were then used to plot the viewer position over the segmented viewing areas, as shown in Fig. 6. The system chooses the rendering viewpoint based on the area where the estimated viewer position falls on. From Table 2, it was found that, based on the estimated viewer position, the averaged percentage of correct viewer position classification is 97.78 %.

Table 1 Means and standard deviations of the estimates of θ and *d* for each of 9 different view positions

Positions	**θ (degree)**		***d* (cm.)**	
	Mean	**Std.**	**Mean**	**Std.**
1	-17.42	0.41	79.25	0.90
2	0.60	1.41	82.94	1.05
3	14.76	0.63	74.27	1.12
4	-17.74	0.58	104.02	1.68
5	-0.64	0.55	113.95	1.27
6	14.64	0.64	99.70	0.82
7	-18.74	0.65	125.25	3.33
8	-2.27	0.36	134.39	1.99
9	16.73	0.67	128.10	7.16

Table 2 Viewer position area classification

Position No.	**Correct Classification (%)**	**Incorrect Classification (%)**
1	100 %	0 %
2	100 %	0 %
3	100 %	0 %
4	100 %	0 %
5	90 %	10 %
6	100 %	0 %
7	90 %	10 %
8	100 %	0 %
9	100 %	0 %
Total	97.78 %	2.22 %

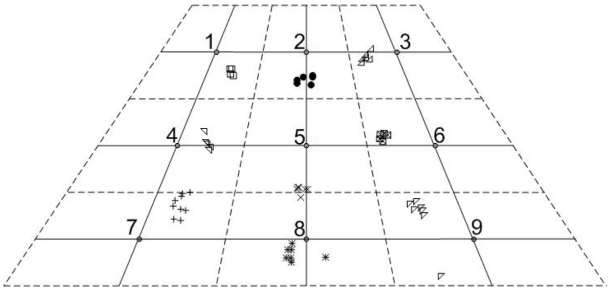

Fig. 6 Estimated viewer positions, as overlaid on the actual viewing areas.

5. CONCLUSION

In this paper, a simple interactive stereoscopic viewing system has been described. A simple image segmentation and feature extraction method has been developed. From the so-obtained feature, a method for estimation of the viewer's head position has been detailed. Error concealment strategy has been given to alleviate the problem due to imprecise head positional parameter estimation.

REFERENCES

[1] L. Lipton, "StereoGraphics Developers' Handbook", *Stereographics Corporation*, 1997.

[2] J. D. Mulder, J. Jansen, and A. van Rhijn, "An Affordable Optical Head Tracking System for Desktop VR/AR Systems", *Eurographics Workshop on Virtual Environments (2003)*, pp. 215-233, 2003.

[3] J. D. Mulder and R. van Liere, "The Personal Space Station: Bringing Interaction Within Reach", *Proceedings of the Virtual Reality International Conference 2002, VRIC 2002*, pp. 73-81, 2002.

[4] R. van Liere, J. D. Mulder, "Optical Tracking Using Projective Invariant Marker Pattern Properties", *Proceedings of the IEEE Virtual Reality 2003*, pp. 191-198, 2003.