A Vision-Based Head Tracker for Fish Tank Virtual Reality – VR without Head Gear –

Jun Rekimoto

SCSL-TR-95-004

February 14, 1995

Sony Computer Science Laboratory Inc.
3-14-13 Higashi-gotanda, Shinagawa-ku,
Tokyo, 141 JAPAN

Also appeared in *Virtual Reality Annual International Simposium 1995* (VRAIS'95).

A Vision-Based Head Tracker for Fish Tank Virtual Reality – VR without Head Gear –

Jun Rekimoto

February 14, 1995

Abstract

A practical and robust head-position tracking method using computer vision is presented. By combining two simple image processing techniques, this tracker can report the position of the user's head in real time. Whole image processing is performed by software running on normal mid-range workstations. This tracker can support desk top virtual reality (also referred to as "fish tank VR"), thereby enabling a user to use a wide range of 3D systems without having to put on any equipment. An experiment conducted by the author suggests this tracker can improve the human's ability in understanding complex 3D structures presented on the display.

1 Introduction

One of the major drawbacks in virtual reality (VR) is its cumbersome devices. A typical VR system requires a user to wear goggles and a position tracker on the head for 3D immersion, and a DataGlove for gesture recognition. Although VR has great potential, such equipment prevents users from accessing its capability in normal situations. Aside that a head-mounted display (HMD) shields a user from the real world, these devices require time to put on and take off, thus making it impossible to quickly switch between *VR mode* and real life mode. The HMD's impact on human health is not yet clear, especially when it is used for long periods of time. It is still impractical and not yet acceptable to wear VR equipment in an office environment.

To overcome these limitations, another approach has emerged recently which uses a normal display screen (either monocular or binocular) coupled with a head tracker that dynamically updates a 3D projection matrix according to the viewer's head position [4, 1, 3]. Arthur, Ware and Booth coined the term "fish tank virtual reality" for this kind of system [1, 16]. With such systems, the user looks through the screen as if looking into a fish tank. Fish tank VR does not provide strong immersion, but is suitable for certain applications such as 3D-CAD or visualization systems because of its ease of use and ability to present high-quality images. For example, Liang's JDCAD system [7], which is a mechanical 3D-CAD system using a 6-degree-of-freedom (DOF) input device, employs a fish tank configuration instead of a normal VR environment.

However, most fish tank VR systems still require a device to track the user's head position. Arthur et al.'s system uses ADL-1, which is a mechanical position tracker and the user is connected to a mechanical rod. Deering's system uses an ultrasonic tracking device; a user must wear an ultrasonic transmitter on their head. As in immersion VR systems, these head trackers also limit the usability of the VR systems.

Due to progress in hardware and the recent boom in multimedia, many of today's workstations and personal computers include a video capturing unit as a standard input device. Using real-time video processing as a method for human-computer interaction is a natural idea,

Figure 1: A snapshot of the system

and has finally become practical. As Aukstakalnis and Blatner claimed in their book, vision-based position tracking "shows great promise for virtual reality systems because of its relative simplicity of use."(*Silicon Mirage* [2], page 36)

Although estimation of human position and orientation using video images under uncontrolled conditions is still only a research topic in computer vision, vision-based head tracking used only for fish tank VR is within reach of today's technology. There are two reasons for this: (1) We can assume the rough position of a user, because the user sits in front of the screen, and (2) we can omit estimation of orientation, because the user is looking at the screen most of the time. These assumptions make it easier to apply head tracking techniques based on image processing in actual 3D systems.

In this paper, I describe a vision-based head position tracker and a fish tank VR system as its application. With this system, a user does not need to wear any special gear. The vision system automatically tracks the position of the user's head while the user is sitting at the desk. A tracking system uses the simple image processing techniques of frame subtraction and template matching based on correlation. Even though image processing is performed by software, the system can achieve 15 frames per second (fps) on a comparatively slow workstation using a MIPS R3000 CPU.

2 Head Tracking Using Computer Vision

Figure 1 shows the system in use. A 3D scene is displayed on a screen monitor. A video camera on top of the screen captures a user's images and estimates its position in real time. The system updates the transformation matrix with respect to the user's head position, and updates 3D images according to it. This causes an illusion that the user is looking at a 3D object through the display screen. Motion parallax caused by head position movement enriches that illusion.

2.1 Head position estimation

Head position estimation takes two steps of image processing techniques as illustrated in Figure 2.

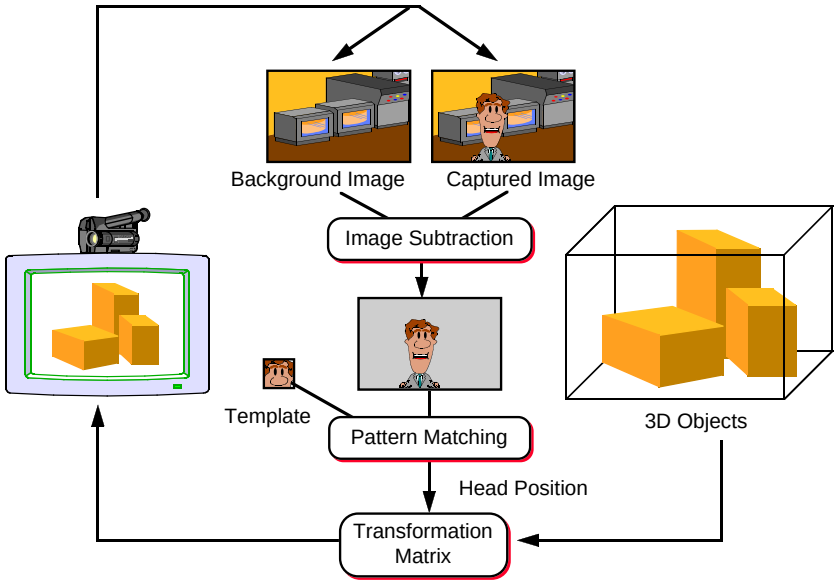
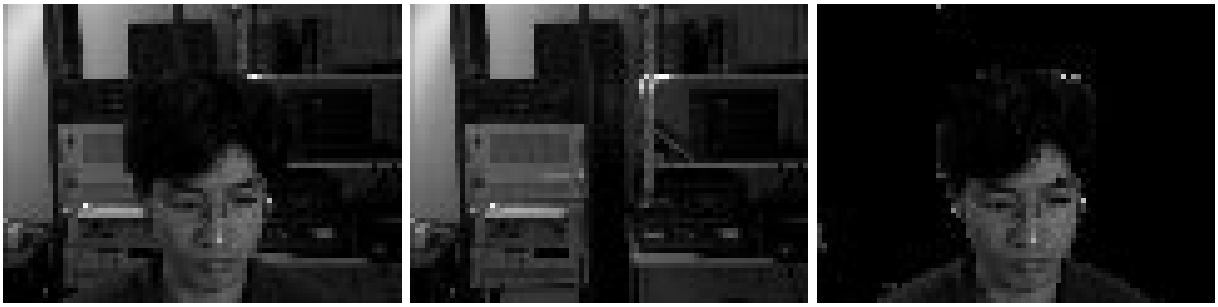


Figure 2: The system overview



(a) Input (b) Background image (c) Result of subtraction

Figure 3: Image Subtraction

First, to detect the user's face area, a pre-stored background image is subtracted from a captured image, pixel by pixel (i.e., a pixel that exceeds a threshold is treated as the user's image). To increase robustness, we use distance in YUV space instead of intensity distance space. By using YUV thresholding, we can get a clear segmentation of the user's image under complex background images (Figure 3).

Second, the system searches for the center of the user's face by using template matching. A partial area of the user's face is stored as a template. Typically, the area between the left and the right eyebrows is used (the user can change the area at any time by clicking a mouse button on the face image). The system calculates correlation coefficients between the template and every area in the face image, and assumes the area with the highest score as the center.

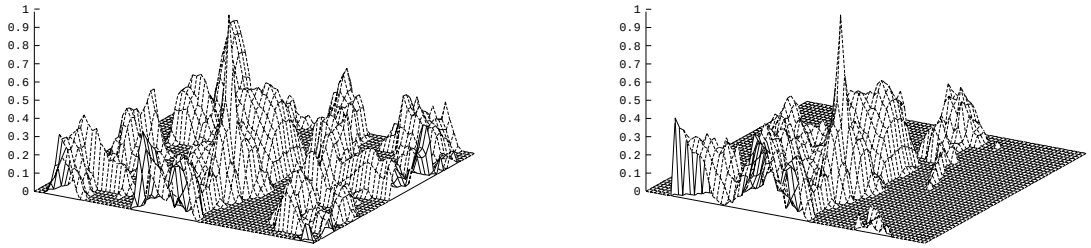
Finally, using (u, v) position on the captured image plane, the system estimates the users head position (x, y, z) by using the following equation:

$$x = C_x - \frac{D}{F}u,$$

$$y = C_y + \frac{D}{F}v,$$

$$z = C_z - D.$$

Where $C_{x,y,z}$ is a camera position in the world coordinates, F is the focal length of the camera, and D is a distance between the user and the camera. Note that we assume the distance



(a) Captured image (b) After frame subtraction

Figure 4: Correlation Intensity values

(a) (b)

Figure 5: Comparison of two position tracking methods. A cross mark between eyes is based on the correlation method, and a mark below is based on the gravity center method. This is shown in (a). When the face is upright, the two methods give similar results. However, when the face is tilted (b), the gravity center method is less accurate than the correlation method.

between the user and the display if fixed. It is not always true and the system performs incorrect perspective transformation when the user is too close to (or too far away from) the display. We will discuss this problem later.

2.2 Template matching with background elimination

Although only template matching seems to be sufficient for position estimation, we combine frame subtraction with template matching as a preprocessing step for the following reasons:

Increasing performance. We can simply omit the background area from the correlation calculation since it is obvious that this area does not include a pattern we are looking for. Doing this greatly improves the performance of position tracking. Figure 4 shows a correlation map for before and after background subtraction.

Increasing robustness. Background elimination also lowers the possibility of mismatching, because the search area is limited to the user's face and body images and the background area is excluded. Doing this increases the robustness of head tracking. Even when the user does not turn their head toward the camera correctly, or tilts their head, the template can still find the correct position. The reason is no other area is as good as that position in calculating correlation values. A simple silhouette-based technique (which uses the gravity center of the

silhouette as a head position) for locating the head position would fail in such a case (Figure 5).

2.3 Off axis perspective projection

Extracted head positions are used to construct a transformation matrix that projects 3D objects onto a display screen.

Traditional perspective projection used in computer graphics assumes that a viewer's line of sight is perpendicular to the view plane (i.e., the screen), and that a view volume is a symmetric frustum. With a fish tank configuration, however, the viewer might look at the screen from a slanted position, thus making a view frustum asymmetric. In such a case, the system must generate an image that looks correct from the user's viewpoint, but might be skewed from a frontal position.

To implement such projections, the system uses a variant of transformation matrices that supports an asymmetric view frustum which is similar to those described in Deering's paper [4]. Using OpenGL [10] or GL [12], such perspective transformation is easily realized by calling the library functions **glFrustum** or **window**, respectively.

2.4 Implementation details and performance

The image processing is done entirely by software except for RGB to YUV conversion. The current system uses 160 $\times$ 120 pixel images for head tracking, and a 12 $\times$ 12 pixel image as a template. With a comparatively slow workstation (SGI IRIS 4D 320VGX using a MIPS R3000 CPU), the system can process at about 15 frames per seconds (fps) for incoming images. To increase performance, the system assumes that the head does not move too fast, and first searches the neighborhood area around the previous head position. When there is a point in the neighborhood area that exceeds the predefined correlation threshold, it is taken as the next head position. If the nearby search fails, the system switches its mode to global search. The current implementation uses a 32 $\times$ 32 pixel area around the previous position for the nearby search.

Although the current frame rate is not as fast as other position trackers such as magnetic or ultrasonic trackers, the user was able to experience a good illusion of motion parallax. Of course, since the processing rate depends on the speed of the CPU, the performance could reach the video frame rate (30 fps) by using faster CPUs which will be available within a few years.

3 Evaluation

To study how our optical head tracker helps a viewer's 3D perception skill, we conducted a task analysis which is originally designed by Sollenberger and Milgram [13]. Similar experiment was also achieved by Arthur, Booth and Ware [1], to evaluate their fish-tank virtual reality system.

3.1 The experiment

In this experiment, three 3D trees standing at the corners of an equilateral triangle are presented to a subject (Figure 6). A leaf of one of the trees is labeled by a small square mark and color. The subject's task is to detect which tree contains this leaf and give the answer via the keyboard. One session consists of 50 questions. The subject alternatively answers these questions with and without head tracking. The system generates the marked leaf and shapes of the trees randomly each time.

Six subjects participated in the experiment. All of them were computer scientists, but were not familiar with 3D computer graphics systems. Each subject had two sessions, answering 100 questions in total. No practice trials are given prior to the experiment. The second session

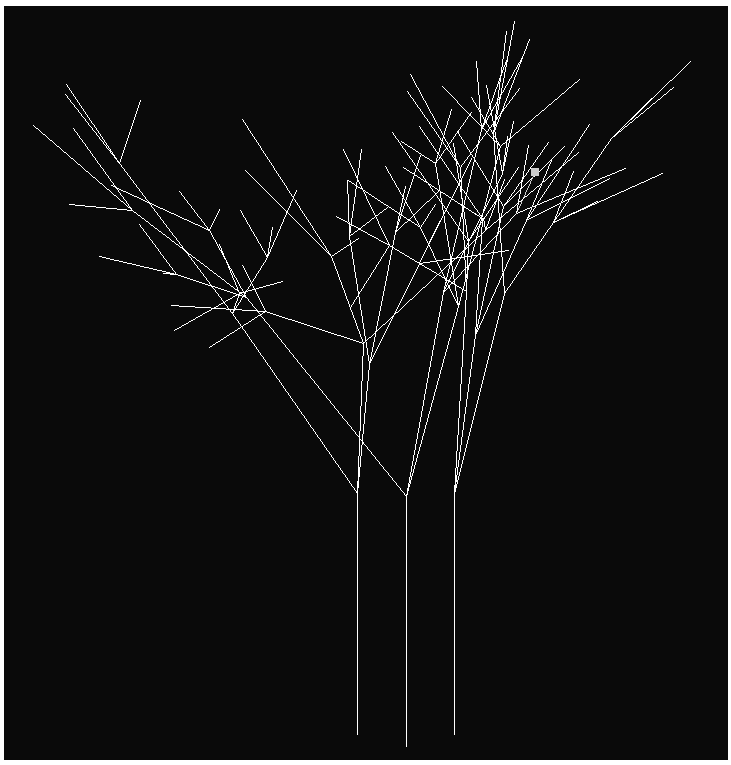


Figure 6: A snapshot of the tree test.

Method	Response Time (sec)	errors (%)
With head tracking	5.95	5.0
Without head tracking	3.50	21.3

Table 1: Experimental results

consisted of exactly the same sequence of questions (marked edges and shapes of the trees), but switched between the tracking methods. The subject thus answered two times for each question, once with head tracking and once without head tracking. To make sure that subjects did not memorize the trial sequence, the sessions were at least a day apart.

3.2 The results of the experiment

The results of the experiment are listed in Table 1, and in Graphs 7 and 8. With head tracking, the subjects took a longer time to answer, but had lower error rates. There are significant differences at the 0.01 level of significance between the two methods for both (average times and error rates). This result is quite similar to Arthur et al.'s experiment (though they used a mechanical head tracker). We thus assume our vision-based head tracker caused the same effect on the subject as Arthur et al.'s mechanical tracker.

A simple explanation for why head tracking was slower is that it requires time for the subject to move their head. Let us discuss this phenomenon more carefully.

As shown in the graph (Figure 9), we did not observe any learning effects in this experiment, though no training trials ware given to the subject. Thus, we can conclude that each response time roughly reflects how difficult the question was. In addition, we often observed that subjects without head tracking often gave up in difficult cases while subjects with head tracking kept

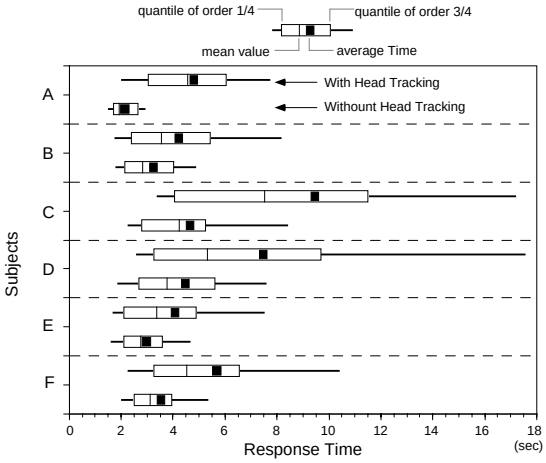


Figure 7: Response times for each subject

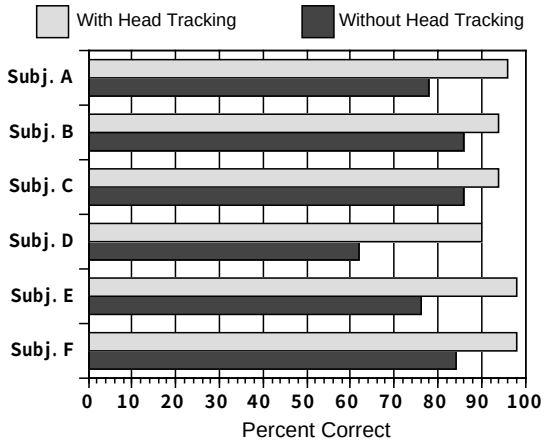


Figure 8: Correct answers ratio for each subject

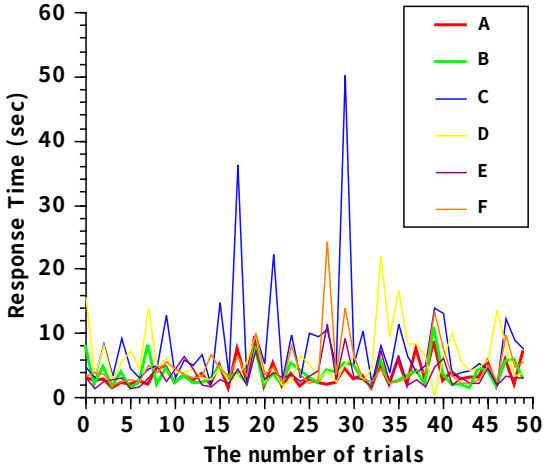


Figure 9: Response time and the number of trials

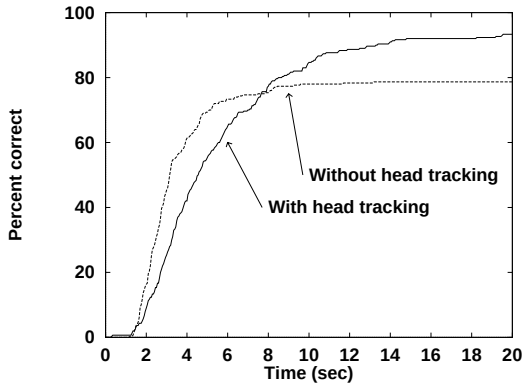


Figure 10: Relationship between response time and the percentage of correct answers

trying by moving their head repeatedly. This was confirmed during informal interviews after the experiment.

The graph in Figure 10 reveals this situation clearly. This graph is a relationship between response time and the percentage of correct answers. As illustrated in the graph, the curve of no-head tracking is saturated more rapidly than that of head tracking. This backs up our observation and can help explain why the average response time for head tracking was longer.

Overall, the result implies that a head tracker can improve a user's ability to understand complex 3D structures presented on the screen (although at the expense of time). This offers a strong incentive for incorporating a head tracker into various 3D applications such as engineering CAD systems and scientific visualization systems.

4 Related Work

Estimation of a user's head position by using optical techniques is not a new idea. A recent survey on position trackers [9] reports various kinds of optical tracking systems as well as magnetic, ultrasonic, and mechanical position tracking systems. Most of them, however, require the user to wear some kind of active element, ranging from an LED array to a video camera [8, 5, 14, 6]. The Honeywell Videometric System [6] uses a passive element, which is a unique symbology pattern on the user's helmet, instead of an active device. Our system also uses a special pattern, an image of the user's face itself. This feature frees the user from having to wear any tracking gear.

Suenaga et al. implemented a vision-based finger position and orientation tracker which works under rather controlled environments[15]. Their system uses two cameras to determine the position of the finger in a 3D space. Some heuristics are applied on a silhouette image of the hand to detect the orientation of the finger.

5 Discussions

5.1 Applications

Since this head tracker does not require special equipment except for a camera and capturing hardware, the technique can be applied on a wide range of 3D applications. Notable applications include 3D engineering CAD and scientific visualization systems, because these systems often require correct perception of complex 3D structures. These applications are also suitable for fish tank VR because they do not require immersion, but require high-resolution images that exceed the standard of today's head mounted displays.

In our method, the template used for pattern matching is obtained from the user's face, so it is different for each user. This requirement becomes a problem when applying this method to a system used in a public environment. However, the template's image resolution is not that high, and satisfactory results can be obtained by preparing a small set of templates representing typical patterns of face images. This should be sufficient to cover most unspecified users.

5.2 Robustness

Robustness is a very important issue if one applies a technique using computer vision into actual applications. In an office, for example, we can assume neither controlled lighting nor a plain background behind a user in processing incoming images.

Regarding our method, the combination of frame subtraction and pattern matching is a good compromise between robustness and processing cost. In our method, frame subtraction is used only for reducing the search area for template matching (thus increasing performance). It can report accurate results even when someone occasionally goes across the background, a situation in which most of the silhouette-based position tracking methods fail.

The stored background becomes different from the actual background over time, partly because the natural lighting is always changing. The system provides simple commands to retake the background and the template. This can be done by simply pressing a key or clicking the mouse on the captured image, so users can change the background or the template at any time without interrupting their tasks. [1]

Although correlation matching shows slight tolerance for tilted or scaled images, it becomes unstable if the user bends their head too far. A possible solution for this problem is to use two or more templates that are the rotated and scaled versions of the original template, or to incorporate rotation invariant correlation filters such as those described in [11].

5.3 Accuracy in tracking

Currently, our tracker does not detect the distance between the camera and the user; it assumes the distance is fixed. This assumption causes the tracker to report inaccurate positions when the user is too close or too far away from the screen. A solution to this is to use two cameras and estimate the distance based on a photographic method. This would require additional hardware and image processing time. An alternative solution I am currently working on is to estimate the distance from the size of the face image. I would also like to mention that omission of distance estimation is less noticeable than other dimensions. When a user is close to the screen, for example, the image on the display looks larger because the physical distance between the user and the display becomes shorter. It has an effect similar to distance tracking.

6 Conclusion

In this paper, I described a vision-based head tracker using a video camera and software that performs image processing. The techniques used here are simple, but are robust and useful for adding reality to various 3D applications. The experimental results suggest this head tracker whill help a user to recognize complex 3D structures displayed on the screen.

I believe a video camera mounted on top of the display will soon be the third standard input device (i.e., after the keyboard and mouse) for desktop computers in the near future. In addition, a camera can be useful for multimedia applications such as teleconferencing, and will be vital for human-computer interaction.

[1] Some users preferred the part of their hairline over the area between their eyebrows for its robustness in matching.

Acknowledgment

I would like to thank Akikazu Takeuchi and other members of the Sony Computer Science Laboratory for their suggestions and comments on this project. Taketo Naito at Keio University provided the source code for a gravity-center-based position detector which eventually became the base of the system described in this paper. Many thanks also go to members in the laboratory who volunteered for the evaluation experiment.

References

[1] Kevin W. Arthur, Kelogg S. Booth, and Colin Ware. Evaluating 3D task performance for fish tank virtual worlds. *ACM Transactions on Information Systems*, Vol. 11, No. 3, pp. 239–265, 1993.

[2] Steve Aukstakalnis and David Blatner. *Silicon Mirage*. Peachpit Press, 1992.

[3] Carolina Cruz-Neira, Daniel J. Sandin, and Thomas A. DeFanti. Surround-screen projection-based virtual reality: the design and implementation of the CAVE. In *Computer Graphics Proceedings*, pp. 135–142, 1993.

[4] Michael Deering. High resolution virtual reality. *Computer Graphics*, Vol. 26, No. 2, pp. 195–202, 1992.

[5] Jih fang Wang, Vernon Chi, and Henry Fuchs. A real-time opcital 3D tracker for head-mounted display systems. *Computer Graphics (Special issue on 1990 Symposium on Interactive 3D Graphics)*, Vol. 24, No. 2, pp. 205–215, March 1990.

[6] F. J. Ferrin. Survey of helmet tracking technologies. In *SPIE Proceedings, Large-Screen-Projection, Avionic, and Helment-Mounted Displays*, volume 1456, pp. 86–94, 1991.

[7] Jiandong Liang and Mark Green. Geometric modeling using six degrees of freedom input devices. In *Proc. of the 3rd International Conference on CAD & Computer Graphics (CAD/Graphics '93)*, Beijing, China, 1993.

[8] R. Mann, G. Rowell, F. Conati, A. Tetwsky, D. Ottenheimer, and E. Antonsson. Precise, rapid, automatic 3-d position and orientation tracking of multiple moving bodies. In *Proceedings of the VIII international congress of biomechanics*, pp. 1104–1112, 1981.

[9] Kenneth Meyer and Hugh L. Applewhite. A survey of position trackers. *Presence*, Vol. 1, No. 2, pp. 173–200, 1992.

[10] Neider, Davis, and Woo. *The OpenGL Programming Guide*. Addison-Wesley, 1993.

[11] Gopalan Ravichandran and David Casasent. Advanced in-place rotation-invariant correlation filters. *IEEE trans. on pattern analysis and machine intelligence*, Vol. 16, No. 4, pp. 415–420, 1994.

[12] Silicon Graphics, Inc., Mountain View, California. *Graphic Library Programming Guide*, 1991.

[13] R. L. Sollenberger and P. Milgram. A comparative study of rotational and stereoscopic computer graphics depth cues. In *Proceedings of the Human Factors Society 35th Annual Meeting*, pp. 1452–1456, 1991.

[14] M. Starks. Stereoscopic video and the quest for virtual reality. In *SPIE proceedings, stereoscopic displays and applications II*, volume 1457, pp. 327–342, 1991.

[15] Yasuhito Suenaga, Kenji Mase, Masaaki Fukumoto, and Yasuhiko Watanabe. Human reader: An advanced human machine interface based on human images and speech. *Trans. Inst. Electronics, Inf. & Comm. Eng.*, Vol. J75-D-II, No. 2, pp. 190–202, 1992.

[16] Colin Ware, Kevin Arthur, and Kellogg S. Booth. Fish tank virtual reality. In *INTER-CHI'93 Conference Proceedings*, pp. 37–42, 1993.