

# Scene Rendering Method To Affect Motion Parallax Due to Head Movements

Takahiro Otsuka and Jun Ohya

ATR Media Integration & Communications Research Laboratories  
 Seika-cho, Soraku-gun, Kyoto 619-0288 Japan  
 Email: {otsuka, ohya}@mic.atr.co.jp

## Abstract

Techniques to generate novel scenes have been proposed in computer vision research. In addition, there have been a lot of 3D models written in VRML for viewing on the WWW. However, the depth of 3D images cannot easily be sensed without special hardware such as HMDs and without special equipment, e.g., shutter and polarized glasses. In this paper, a method is proposed that uses motion parallax as a depth cue to let users perceive the 3D structures of models rendered into images. In this method, the user's head is tracked and the motion parameters are estimated by using a computer vision technique. Then, the 3D object is rendered from the estimated eye position as a new viewpoint. The system has been built on an SGI O2 workstation with its mounted camera used for head tracking. The experimental results are very good; users feel as if the 3D objects are placed in front of the screen of the monitor.

## 1 Introduction

There are growing needs to view 3D models and synthesized 3D scenes with depth perception, as access to 3D objects has become easier and has grown in popularity. In computer vision research, various methods have been proposed that enable novel scenes to be generated from a wide range of viewpoints using two or three images taken from different viewpoints [1, 2, 3]. In addition, there are a lot of 3D models written in VRML for viewing on the WWW. However, the depth of 3D images cannot easily be sensed by humans without special hardware such as HMDs and without special equipment, e.g., glasses with liquid-crystal shutter lenses and polarized glasses.

In this paper, a method is proposed that uses motion parallax as a depth cue to let users perceive the 3D structures of models rendered into images. In this method, the user's head is tracked and the motion parameters are estimated by using a computer vision technique. Then, the 3D model is rendered from the estimated eye position as a new viewpoint.

Our work is related to the research on dynamic eye-point displays being carried out at NASA [4]; that is, ordinary 2D displays are used to affect realistic impressions by using motion parallax. In that research, however, special hardware devices such as magnetic sensors are used to track the head. Such usage is feasible when the research targets special applications such as simulations for astronauts and pilots. On the contrary, our research is centered on

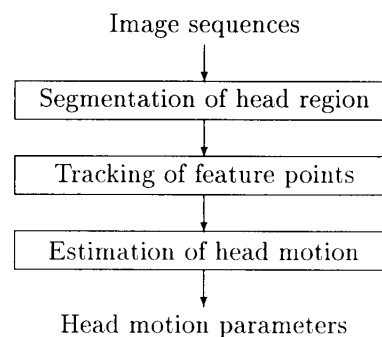


Figure 1: Flow of head tracking algorithm.

general circumstances in front of a desk-top computer.

In addition, our approach is supported by psychological evidence [5] showing a more accurate recovery of 3D information from motion parallax than from object rotation about the vertical axis. The latter technique had previously been a standard interaction method for displaying 3D objects.

This paper proceeds as follows. In Section 2, our proposed method is explained. Then, the experimental results are mentioned in Section 3. Finally, the paper is concluded in Section 4.

## 2 Method for Scene Rendering

Our proposed method consists of two steps. The first step involves head tracking, where the 3D coordinates of the head position and the rotation angle from the frontal face are estimated. The estimation results are used to compute the eye position. The second step involves scene rendering, where a 3D model is rendered from the estimated eye position as a new viewpoint.

### 2.1 Head tracking

Head tracking involves the tracking of feature points on the face from which the translation and rotation of the head are estimated [6]. The method for head tracking consists of three steps as shown in Fig. 1.

First, the head region is extracted by matching a circle to line segments in a gradient image (Fig. 2). If the background is white and not cluttered as shown in Fig. 2, this simple algorithm is robust and quick.



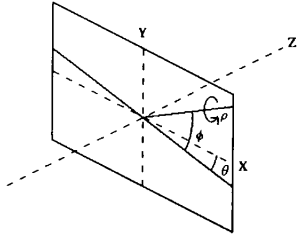


Figure 5: KvD representation of rotation.

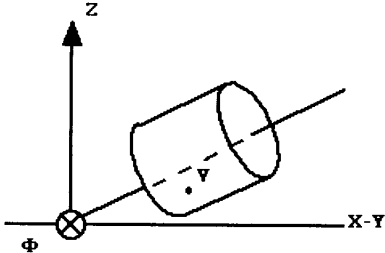


Figure 6: Cylinder model for a head.

where  $V_z$  denotes the z-component of the vector  $V$ , and  $\Delta V_\perp$  denotes the component of the displacement vector  $\Delta V$  perpendicular to the axis  $\Phi$ .

We have explained a way of estimating the rotation of the head. The translation of the head is estimated from the displacement of the centroid of the feature points successfully tracked between two images.

The estimated parameters are converted into a transformation vector and a rotation matrix, both of which are cascaded over a sequence from the first frame to get a vector  $D$  and a matrix  $R$ . Then, the position of the eye can be estimated as follows.

$$\mathbf{P} = R\mathbf{P}_0 + \mathbf{D}, \quad (8)$$

where  $\mathbf{P}$  and  $\mathbf{P}_0$  are the estimated and original coordinate of the eye, respectively.

## 2.2 Scene rendering

In scene rendering, a scene image or an object image is rendered from a 3D wire-frame model of the scene or object by some projection model, i.e., orthographic or perspective. The realism of the image can be increased by the techniques of texture mapping and lighting. Our proposed method is intended to increase the realism by rendering the image from the user's viewpoint so as to let the user perceive the motion parallax along with the head motion.

Motion parallax is one type of depth cue; others are binocular stereopsis, accommodation, and convergence [12]. While binocular stereopsis supplies depth information in terms of the difference in the position of the same point in two images, motion parallax supplies depth information in terms of the velocity of the points generated by the relative motion between the object and the viewer. Therefore, to generate motion parallax, the velocity at each point must be computed according to the depth of the point from the viewer.

The relationship between velocity and depth is illustrated in Fig. 7 in the case that the eye moves toward the positive direction of the  $x$  axis. Here, we assume that the viewer is looking at the scene with only one eye and closes the other eye. This is because the effect of motion parallax is eliminated

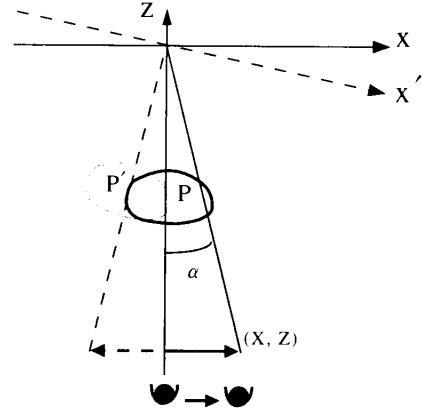


Figure 7: Projection model against head motion.

when the binocular stereopsis contradicts with the motion parallax. We also assume that the scene is placed in front of the monitor because the effect of motion parallax becomes larger when the point is nearer to the viewer. The figure shows that the movement of the eye is equivalent to the rotation of the world other than the eye about the  $y$  axis, where the  $x$ - $y$  plane is defined to be the plane of the monitor. Then, the image can be generated by rotating the scene by angle  $\alpha (= \tan^{-1}(X/Z))$ , and by projecting the scene onto the  $x$ - $y$  plane along the positive direction of the  $z$  axis. Finally, the image on the  $x$ - $y$  plane is projected to the  $x'$ - $y$  plane by enlargement in the  $x$  direction by  $\sec(\alpha)$ .

The matrix operations mentioned above such as rotation and scaling can be easily coded in OpenGL [13] with the optimized computational performance. In addition, the effect of motion parallax can be achieved along the  $z$  axis by the perspective camera model in OpenGL.

When the eye motions point to a general direction  $(X, Y, Z)$  from the original position  $(0, 0, Z)$ , the rotation angle  $\alpha$  becomes  $\tan^{-1}(\sqrt{X^2 + Y^2}/Z)$ , and the scaling for the  $x$  and  $y$  axes become  $\sqrt{X^2 + Z^2}/Z$  and  $\sqrt{Y^2 + Z^2}/Z$ , respectively. In the experiments mentioned below, the displacement of the eye position  $(\Delta X, \Delta Y)$  from the position at the first frame is estimated from the head tracking result, from which the scene is rendered.

## 3 Experimental Results

Experiments were executed on an SGI O2 workstation to evaluate our proposed method. An image of the users was captured from a camera mounted on the monitor. Sixty feature points were tracked to estimate the head motions such as translation and rotation. At the first frame, the center of the two eyes was marked manually to align the face position.

Figure 8 shows the tracking performance of our proposed method under various head motions such as yaw, pitch, and roll. As the tracking performance of the method depends on the speed of the head motions, the motions were made to be slower than usual. The figure shows a good tracking performance, i.e., the estimated left eye position is located near the true eye position. The error of the points was evaluated quantitatively as within 5 pixels for sequences of about 10 seconds [6]. The computational speed for the head tracking was about 11 Hz.





them due to space constraints. In the full paper, we will describe two examples of the touch among avatars, such as playing with the see-saw and the hand shake. In Fig.1 we show the scene where a bunny girl and a guy playing with the see-saw. The see-saw bar is placed as a body which represent the field in the virtual environments. The bunny and the guy are controlled as avatars by users.

Describing the touch among two avatars requires three or more bodies. Two bodies are used for representing avatars and at least one body is used for representing the field. In Fig. 2 we show the combination of three bodies connected by jacks and plugs. In some situations, there are two or more bodies which represent the fields. In Fig.3 we show the combination of four bodies. In any cases, no direct combinations are required among bodies for avatars.

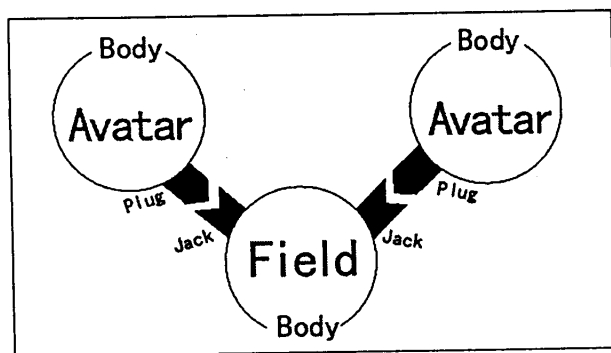


Fig.2 A combination of bodies to represent the touch among avatars

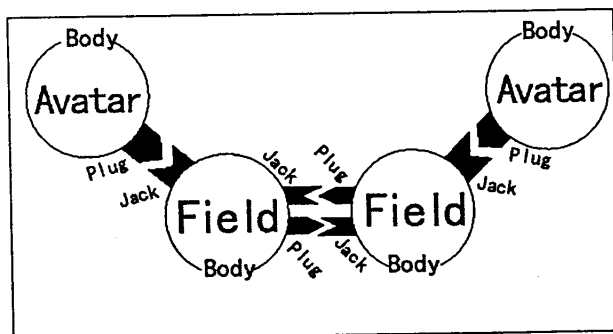


Fig.3 Another combination of bodies to represent the touch among avatars

## 5 CONCLUDING REMARKS

In this paper, we introduced the touch among avatars as a means of an interaction among 3D-objects. We introduced a model for describing interactions among 3D-objects in 3D-MUVEs and described examples of the touch among avatars according to our interaction model. We found these examples were properly described in the same way.

We have a further plan to describe other modes of non-verbal communications, such as facial expressions, gestures and etc according to our

interaction model. These modes can be described as the extension of the touch among avatars because these modes are used as the substitutions of the touch [2].

## REFERENCES

- [1] Takao Kurokawa, Non-verbal Interfaces, IEICE, Ohm, 1994
- [2] Marjorie F. Vargas, An Introduction to Nonverbal Communication, LOUDER THAN WORDS, 1987
- [3] Yoshiaki Araki, Body&Magnet: A Model for Dynamic Interactions among 3D-objects in Virtual Environmets, proc of Inteaaction'98, March 3-4 1998 Tokyo Japan, pp73-80
- [4] Zyda, Michael , Pratt, David R., Pratt, Shirley, Barham, Paul and Falby, John S. "NPSNET-HUMAN: Inserting the Human into the Networked Synthetic Environment," proc of 13th DIS Workshop, 18 - 22 September 1995, Orlando, Florida, pp.103-106.
- [5] Hiroshi Tamura, Human Interfaces, Ohm, 1998.