ICAT'98

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 poralized 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 build 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 poralized 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 general circumstances in front of a desk-top computer.

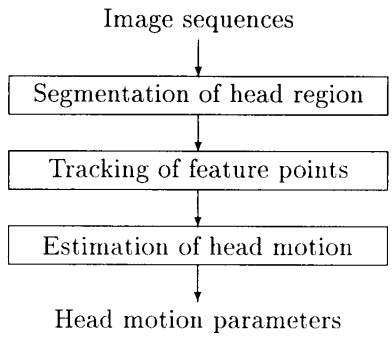

Figure 1: Flow of head tracking algorithm.

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 2: Example of a segmentation result.

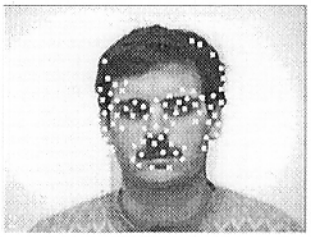

Figure 3: Example of selected feature points.

For cluttered backgrounds, a method combining the edge and color information [7] can be used.

Then, feature points are selected from the segmented region, and tracked over the successive frames. Here, we apply the Kanade-Lucas-Tomasi tracker [8] for feature point tracking, and apply a criterion that was proposed by Shi and Tomasi [9] for feature point selection.

In the Kanade-Lucas-Tomasi tracker, the displacement of feature points between two images is computed by the following equation.

$$\mathbf{Zd} = \int\int_{\mathbf{W}} [I(\mathbf{x}) - J(\mathbf{x})] \begin{bmatrix} I_x \\ I_y \end{bmatrix} \mathrm{d}\mathbf{x}, \qquad (1)$$

where $\mathbf{d}$ is the displacement vector, $I(x)$ and $J(x)$ are the intensity values of the two images, and $\mathbf{Z}$ is a correlation matrix between the components of the gradient vector shown below.

$$\mathbf{Z} = \begin{bmatrix} g_x^2 & g_x g_y \\ g_x g_y & g_y^2 \end{bmatrix}. \qquad (2)$$

Equation 1 is derived by minimizing the squared error between the two images (Eq. 3), and neglecting the terms of $\mathbf{d}$ greater than the second order.

$$\int\int_{\mathbf{W}} \left[I(x - \frac{\mathbf{d}}{2}) - J(x + \frac{\mathbf{d}}{2}) \right]^2 \mathrm{d}\mathbf{x}. \qquad (3)$$

The selection of feature points is based on the correlation matrix $\mathbf{Z}$, that is, the decreasing order of the second eigenvalue of matrix $\mathbf{Z}$. The second eigenvalue becomes positive only if the Taylor series of the intensity around the pixels has non-linear terms. Therefore, points on line segments or points on the linear edges are excluded from the selection of feature points by having points with a non-zero second eigenvalue be selected. These points are related to the "aperture problem," where the motion field is not fully defined at each pixel since we have only one equation between its coordinates that is derived from the conservation of intensity value. A larger second eigenvalue means a larger change of the intensity around the pixels. Therefore, tracking may be more robust for points with a larger second eigenvalue. An example of feature point selection is shown in Fig. 3.

Finally, the translation and rotation of the head are estimated based on a weak perspective projection model [10] (Fig. 4). In this model, an object is first projected onto a virtual plane at the center of the object orthographically. Then, the image on the virtual plane is projected onto an image plane perspectively. Since the depth of the object is assumed to be

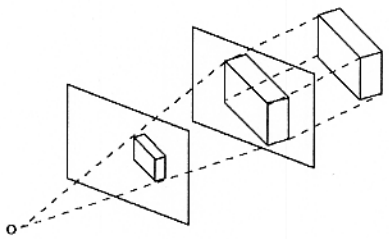

Figure 4: Weak perspective projection model.

zero, the focal length is regarded as infinity. Therefore, the epipolar constraint is simplified as shown below, where the coefficients are independent of the positions of points.

$$ax' + by' + cx + dy + e = 0, \qquad (4)$$

where (x, y) and (x', y') are the coordinates of the points for two images.

From the coordinates of successfully tracked feature points between two images, the coefficients in Eq. 4 are obtained by minimizing the squared errors shown below.

$$E(a, b, c, d, e) = \frac{\sum_i (ax'_i + by'_i + cx_i + dy_i + e)^2}{a^2 + b^2 + c^2 + d^2} \qquad (5)$$

Then, the rotation angle between the two images is calculated by the following equations.

$$\tan\phi = \frac{b}{a}, \quad \tan(\phi - \theta) = \frac{d}{c}, \quad s^2 = \frac{c^2 + d^2}{a^2 + b^2} \qquad (6)$$

where θ is the rotation angle of the x axis, ϕ is the angle between the x axis and the axis of rotation Φ parallel to the image plane, and s is a scaling factor due to the motion along the optical axis or the z axis.

The representation of the rotation angle mentioned above was proposed by Koenderink and van Doorn [11]. This representation is useful in that the rotation angle is divided into angles which can be estimated by a weak perspective projection model, i.e., θ and ϕ, and an angle that can not be estimated, i.e., ρ which is the rotation angle about the axis Φ.

To estimate the angle ρ, the depth information of the object is necessary because of the existence of the bas-relief ambiguity making it impossible to separate the angle ρ and the depth of points from the displacement of the points. Therefore, we assume that the shape of the head is a cylinder (Fig. 6). Then, the angle ρ is obtained from the following equation.

$$\rho = \frac{\triangle V_\perp}{V_z}, \qquad (7)$$

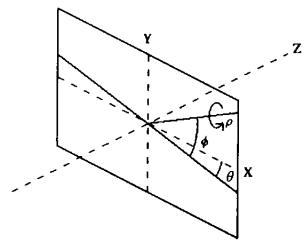

Figure 5: KvD representation of rotation.

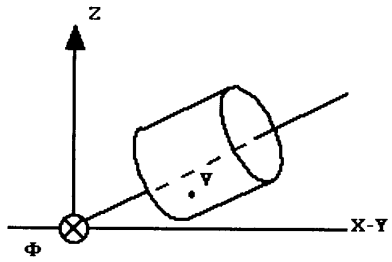

Figure 6: Cylinder model for a head.

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

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 successflly 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.

$$\mathbf{P} = \mathbf{R}\mathbf{P_0} + \mathbf{D}, \tag{8}$$

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, accomodation, and convergence [12]. While binoculr 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

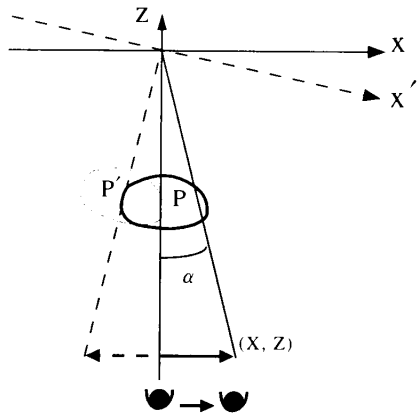

Figure 7: Projection model against head motion.

when the binocurar 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 α 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 $(\triangle X, \triangle Y)$ 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.

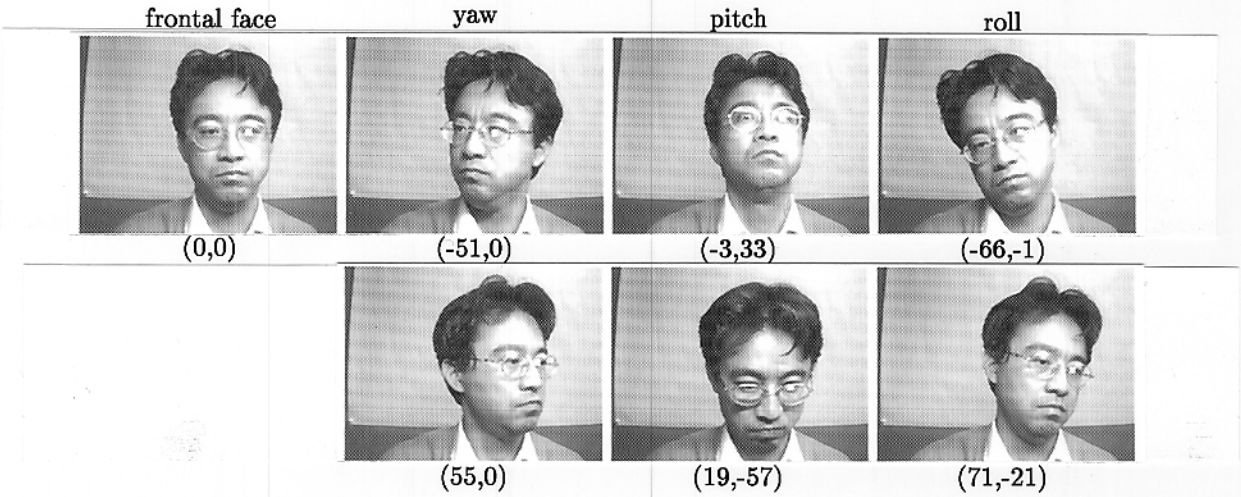

Figure 8: Example of head tracking results (white dot: estimated left eye position). The displacement vector is represented in pixels.

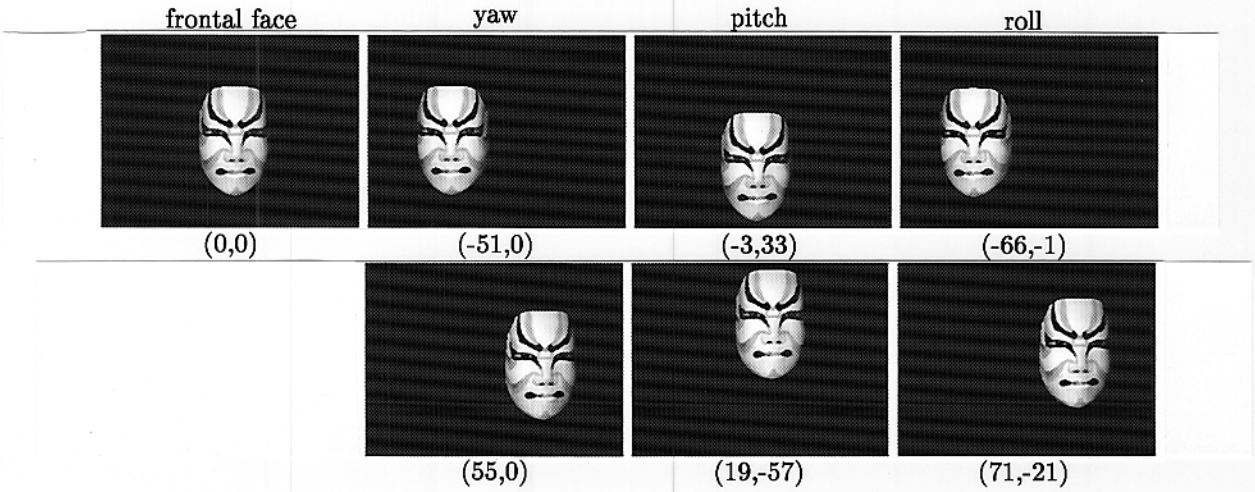

Figure 9: Example of scene rendering results for a 3D face model.

The proposed method was evaluated by using two 3D models. One was a 3D face model whose range data was measured by Cyberware. The other was a 3D city model created by a scene generation algorithm [3] from a stereo pair of images. Figure 9 shows the results for the 3D face model rendered for the head poses shown in Fig. 8. The surface of the model consisted of 642 polygons. The computational speed was about 8.3 Hz.

Figure 10 shows the results for a 3D city model. The surface of the model consisted of 4661 polygons. The computational speed was about 6.7 Hz.

The accuracy of the proposed method was subjectively evaluated by placing a long stick between the face and monitor, and the change of the tip of the stick and the point of the rendered scene was compared while the head was rotating. In this experiment, the placement of the stick did not affect the tracking of the head because the stick did not belong to the field of view of the camera.

From this experiment, it is not possible to estimate the accuracy quantitatively. However, the motion directions of the stick and the scene were almost same during the whole sequence. To evaluate the accuracy quantitatively, we believe it is necessary to measure the ground truth of the head motion and to record the video sequence from a camera attached on the head.

The image changes were very smooth and the response times were fast against the head motions. Therefore, the user could feel the existence of the 3D object in front of the screen of the monitor.

4 Conclusion

In this paper, a method has been proposed that uses motion parallax as a depth cue to 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 build on an SGI O2 workstation with its mounted camera used for head tracking. The experimental results were very good; users felt as if the 3D objects were placed in front of the screen of the monitor.

Future work includes the construction of a VRML browser to display 3D models written in VRML with

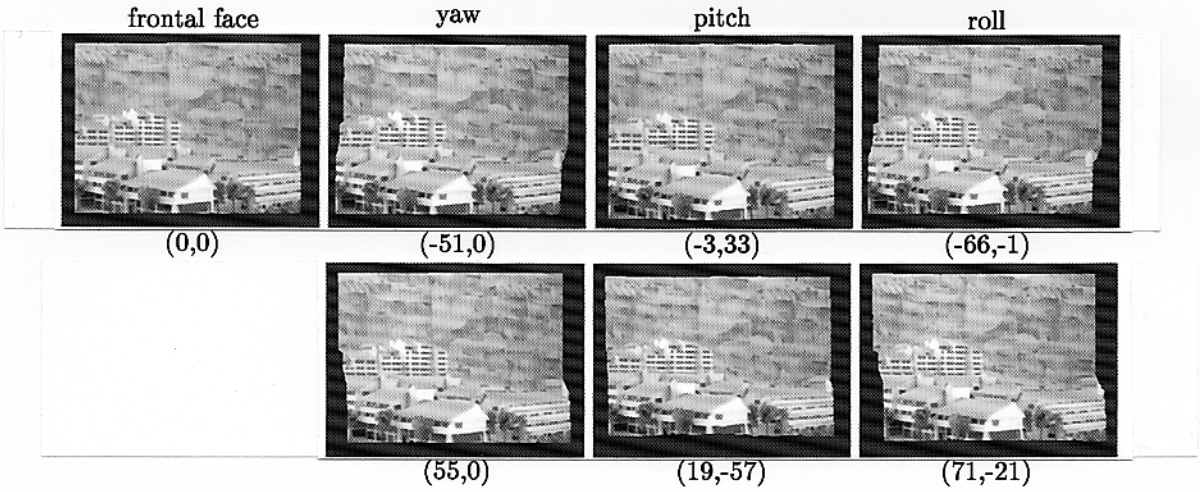

Figure 10: Example of scene rendering results for a 3D city model.

viewpoint changes according to the eye position.

References

[1] S. Seitz and C. Dyer, "Toward image-based scene representation using view morphing," *Proc. of the 13th ICPR*, pp. 84-89, 1996.

[2] A. Shashua and S. Avidan, "The Rank 4 constraints in multiple ($\geq$ 3) view geometry," *Proc. ECCV*, pp. 196-206, Apr. 1996.

[3] K. Sengupta and J. Ohya, "Novel scene generation, merging and stitching views using the 2D affine space," *Proc. IEEE Int. Conference on Multimedia Computing and Systems*, pp. 602-603, Jun. 1997.

[4] M. K. Kaiser and D. R. Profitt, "Using the stereokinetic effect to convey depth: Computationally efficient depth-from-motion displays," *Human Factors*, 34, pp. 583-600, 1992.

[5] F. H. Durgin, D. R. Proffitt, and K. S. Reinke, "Comparing depth from motion with depth from binocular disparity," *Journal of Experimental Psychology: Human Perception and Performance*, Vol. 21, No. 3, pp. 679-699, 1995.

[6] T. Otsuka and J. Ohya, "Real-time estimation of head motion using weak perspective Epipolar Geometry," *Proc. 4th IEEE Workshop on Application of Computer Vision*, pp. 220-225, Oct. 1998.

[7] S. Birchfield, "Elliptical head tracking using intensity gradients and color histograms," *Proc. CVPR'98*, pp. 232-237, Jun. 1998.

[8] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," *Proc. IJCAI-81*, pp. 674-679, 1981.

[9] J. Shi and C. Tomasi, "Good features to track," *Proc of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593-600, 1994.

[10] L. Shapiro, A. Zisserman, and M. Brady, "3D motion recovery via affine epipolar geometry," *Int. J. Computer Vision*, Vol. 16, No. 2, pp. 147-182, 1995.

[11] J. Koenderink and A. van Doorn, "Affine structure from motion," *J. of Optical Society of America*, Vol. 8, No. 2, pp. 377-385, 1991.

[12] L. Kaufman, "*Sight and Mind*," Chapter 7, Oxford, 1974.

[13] J. Neider, T. Davis, and M. Woo, "*OpenGL Programming Guide*," Addison-Wesley, Reading, MA, 1993.