In Proceedings of the 7th IEEE International Workshop on Robot and Human Communication (RO-MAN '98), pp. 121–126, Takamatsu, Japan, October 1998.

Improvement of Temporal Quality of HMD for Rotational Motion

Yasuyuki YANAGIDA, Masahiko INAMI and Susumu TACHI

Department of Mathematical Engineering and Information Physics,
School of Engineering, The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656 Japan
E-mail: {yanagida, minami, tachi}@star.t.u-tokyo.ac.jp

Abstract

A concept to improve the temporal quality of visual display by providing consistent optical flow of the displayed image is proposed. Based on this concept, a method to locally compensate the time delay of the image displayed through HMDs for the operator's rotational head motion is shown, by using signals obtained by additional sensors.

1 Introduction

In order to realize better sensation of presence provided by VR and/or tele-existence systems, it is important to improve both of *spatial* and *temporal* quality. The improvement of spatial quality includes development of visual display devices with high resolution and wide field of view as well as the generation of photo-realistic graphics images, whereas the improvement of temporal quality includes obtaining better response of the system against the operator's motion. However, many VR systems do not seem to have enough temporal quality yet. The temporal quality of visual displays can be analyzed by two elements: frame rate and total time delay.

The limitation of frame rate is caused by double-buffering scheme, which result in 10-20 frames per second (fps) for many systems, though 60 fps or more is preferred considering the characteristics of human beings' visual sensation. Unfortunately, this rate does not seem to be improved in spite of recent drastic improvement of graphics hardware, as application designers would like to display more complicated virtual world when faster graphics systems become available.

Another temporal performance is the total time delay of the system [1], which is measured from the occurrence of human motion through the calculation of the status of the world to the moment when the image reflecting that motion is displayed on the device. This delay always exists in VR systems, because of existence of the processing time at each element in the system (e.g. measurement subsystem and graphics subsystem) and the communication delay between these elements. Low frame rate and/or long time delay degrade the sensation of presence and inspire the operator with fatigue.

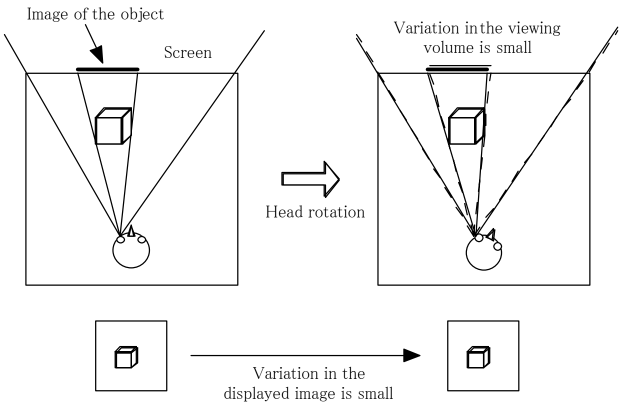
In this paper, we propose a method to improve the temporal quality of VR and/or tele-existence systems, by equivalently providing higher frame rate and smaller time delay, independent of the improvement of the performance of the graphics subsystems. We design the method focusing on the optical flow presented to the human operator, as well as considering simplest implementation which does not require any complicated calculation.

In Section 2, several related works are introduced and the overview of our approach is described. In Section 3, we compare two kinds of visual display systems, HMD and CAVE/CABIN, to find out one of the problems with HMD when the operator rotates his/her head. In Section 4, our method to compensate the image delay is described, including several kinds of implementation. In Section 5, an analysis is made to estimate the applicability of our method.

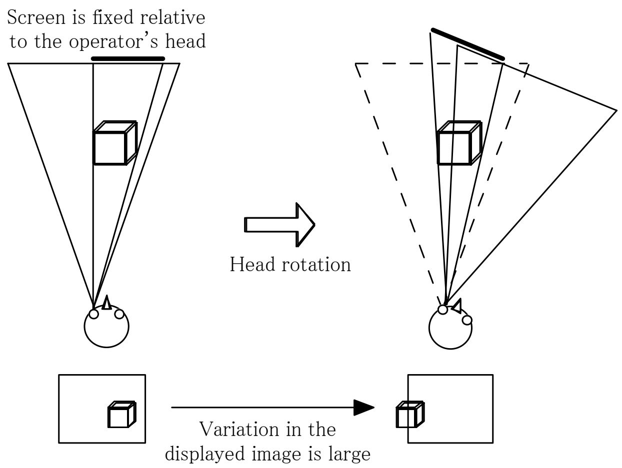
2 Related Works

Several attempts have been made to overcome the problems related with insufficient frame rate and large amount of time delay. For example, the method of frameless rendering [2] was developed to solve the problem with discrete image generated using double-buffering scheme, resulting in continuously varying pixel images. However, we cannot obtain correct optical flow over the entire screen using this method, as the pixels to be updated are randomly selected within the image area, leaving significant amount of pixels unchanged at each moment. Recently the post-rendering 3D warping technology [3] was developed to enable higher frame rate even if using complex and heavy rendering algorithm. The basic idea of this technology lies in that a local loop closely related to the change of the viewing point is configured, avoiding significant time delay which is necessary for generating images using complex rendering algorithm.

Our approach and the post-rendering warping technology have somewhat common basis in that both of them configure rapid local loop nearby the operator, aiming at realizing smooth interaction between the operator and the virtual world. However, they are different in that our method focuses on total aptitude of information needed for updating the image displayed to the human operator, rather than developing special algorithm for image generation. We use additional sensor suitable for detecting real-time motion (velocity) rather than limiting ourselves to use only conventional position and orientation sensors.



(a) Case of CAVE/CABIN-type Displays



(b) Case of HMD

Figure 1 Change of Image against Head Rotation

3 Temporal Problems with HMD

As the screens of HMD are fixed relative to the operator's head, it is necessary to generate, update and display the image in real time according to the operator's head motion, especially the rotational motion. There are no problems if the update of the image is truly real time, however, this situation is hardly realized. The low frame rate and/or large time delay degrade the sensation of reality and cause the operator's fatigue.

Now we point out the problems on temporal quality of HMD systems, by comparing with systems using fixed screens, such as CAVE [4] and CABIN [5]. The screens of CAVE/CABIN are fixed to the surrounding space, and the information needed for generating graphics images in CAVE/CABIN is only the position of viewing point. Hence the displayed image on the screen does not change significantly when the operator rotates his/her head.

On the other hand, the screens of HMD are fixed to the operator's head, and the image displayed on the screen moves drastically according to the operator's head rotation. This image motion causes uniform optical flow over the whole screen and directly effects on the operator's sensation of presence. If this optical flow does not match the operator's kinesthetic sensation, the operator feels as if the virtual environment followed after his/her head ratio with significant time delay, or the world vibrated when he/she rotates the head.

4 A Method to Compensate the Delay

4.1 Basic Principle

To overcome the problem mentioned above, an approach of improving the performance of the graphics subsystem might be considered. However, the frame rate for many applications, as mentioned above, would not be improved in spite of the emergence of high-performance graphics systems. Moreover, the time delay originated from motion measurement and other elements cannot be reduced by the improvement of graphics hardware, resulting in a certain amount of residual time delay.

In our approach, we propose a method to *locally* compensate the delay of the displayed image caused by the restricted frame rate and the system's overall time delay. The image generated by the graphics subsystem is locally adjusted before it is displayed to the human operator, using the information provided by additional sensors. As mentioned before, the characteristic of our method is to compose a high-speed local feedback loop at the stage of display device so that the effect of the operator's tor's motion is separated into the global (less temporally critical) and local (rapid) loops. Thus the image is displayed to the operator in real time so that the optical flow is presented correctly according to the operator's motion.

Here we restrict ourselves to deal with the operator's rotational head motion only, i.e., the situation which generates uniform optical flow over the entire image area. The proposed method is as follows:

1. Calculate the amount of rotation angle to be compensated.
2. Adjust the image to be displayed now, using most recently generated graphics image.

The generation of recomposed image should be taken place in real time, i.e., within the time allowed to realize desired frame rate (60fps or more). If the system is correctly configured, the displayed virtual world is recognized as a stable world to the operator. So the system can be called "World Stabilizer for HMD". The proposed concept is shown in Figure 2. In the following sections, several elemental analyses are described to implement the proposed method.

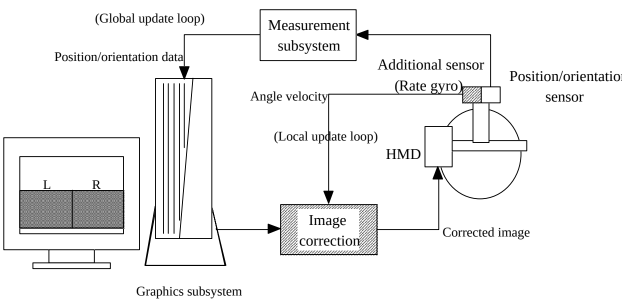


Figure 2 Concept of World Stabilizer

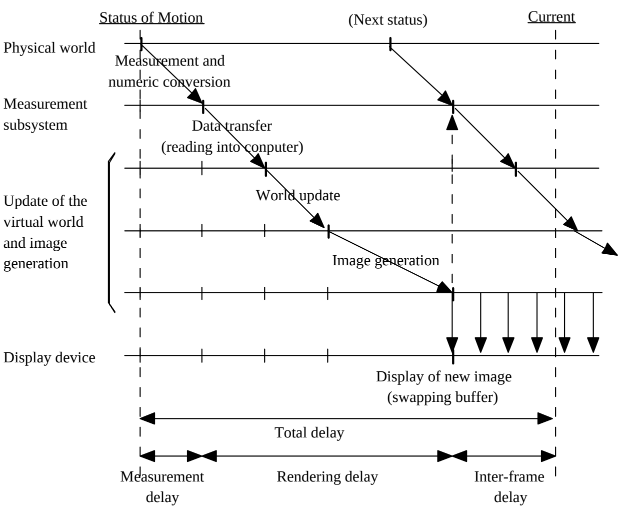


Figure 3 Delay included in displayed image

4.2 Analysis of Time Delay

The image displayed on the HMD screen reflects the *past* status of the world, as various kinds of delay exist through the process of motion measurement and image generation. There is also additional delay since the most recent image was displayed until the next updated image is displayed (Figure 3). In order to determine the necessary amount of rotation angle to be compensated, we separate the overall into the following categories according to their characteristics:

- Rendering delay: time consumed by the calculation of world status update and image generation. It appears as the period between the subsequent image output. It also includes the time necessary for reading measurement data.
- Measurement delay: time since the moment when the status of the physical world is captured by the measurement subsystem to the moment when the computer begins to read the data and starts the calculation loop for world update and image generation.
- Inter-frame delay: time since the most recent image was displayed until the current moment.

The total time delay of the currently displayed image will be the sum of above three kinds of delay.

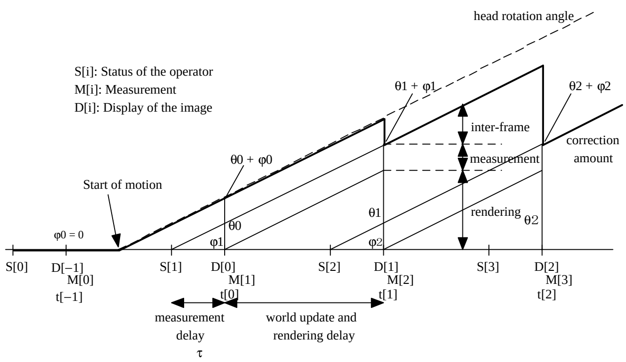


Figure 4 Head rotation angle to be compensated

4.3 Calculation of Compensation Angle

Based on this analysis, we selected rate gyro as a device to detect the rotation angle, as it is able to measure angle velocity much faster than and with much less time delay than absolute position/orientation sensors, such as magnetic trackers. If we use rate gyros to obtain absolute orientation of the operator's head, we have to be confronted with the problem of drift and integration error. However, we avoid this problem by developing algorithm to obtain the amount of rotation angle to be compensated, which only use instantaneous value of angle calculated from short-term integration of angle velocity and discard the old integrated value when the updated image is displayed. As the information containing value of angle is usually measured by ordinary position/orientation sensor and it will be reflected to the generated graphics image after a certain time delay, we have only to compensate short-term variation of the rotation angle.

The method to calculate the head rotation angle to be compensated is shown in Figure 4. The total compensated angle is the sum of those related to measurement delay, rendering delay and inter-frame delay, respectively. Among these angles, those related to measurement delay and rendering delay are constant while a single image is being displayed, whereas the angle related to inter-frame display varies according to the short-term integration of the instantaneous value of angle velocity.

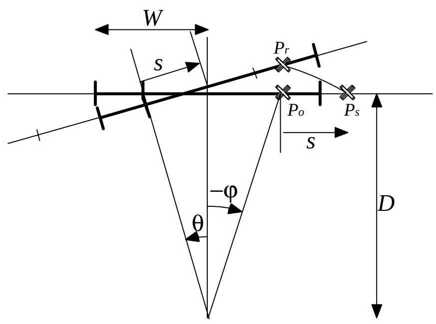
4.4 Methods of Image Correction

In order to make the operator recognize the displayed world to be stable, the image generated by the graphics subsystem will be displayed at the correct position, regarding the difference between the current head orientation and that of the moment at which the status is reflected to the generated image. There are several methods of image correction to be considered:

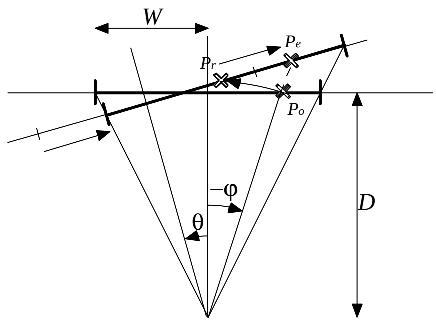
(a) Implementation using simple two-dimensional image shift and rotation. The displayed image is obtained by simply shifting the original image, and is suited to be implemented in the form of simple hardware. As there are difference between the shape of the viewing volume of the moment at which the status was captured and that of the moment at which the image is displayed, the displayed image has a certain distortion, especially at the peripheral vision. However, this implementation is suitable for local hardware so that the image correction can easily be done in real time.

(b) Implementation using image resizing as well as shift and rotation. In order to reduce the image distortion at peripheral vision in method (a), the whole image size will be adjusted. There is still a certain image distortion remained, if the entire image is uniformly expanded or shrunk. The non-uniform expansion or shrinkage is, if exact, equivalent to the method (c).

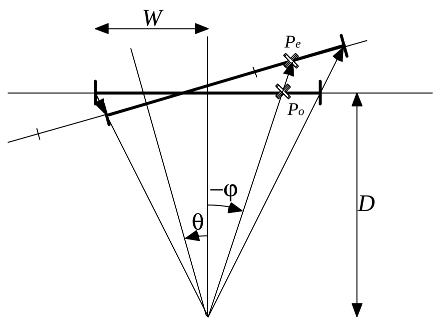
(c) Implementation using texture mapping. Three-dimensional compensation is taken place. This is the most accurate way of image correction and ideally there is no image distortion. Using this method, the system will be equivalent to CAVE-type systems, which use fixed screen. It is effective to use real-time video texture mapping to do this, but it requires expensive graphics/video hardware. Moreover, it requires significant amount of computation so that it is not suited to be implemented as local hardware.



(a) 2D shift and rotation only



(b) 2D shift, rotation and resizing



(c) Texture mapped

Figure 5 Methods of Image Correction

These three types of image correction are shown in Figure 5. In each figure, symbol θ shows the head rotational angle to be compensated (calculated by the method described in Section 4.3), and φ represents the angle of deviation of a certain point P_o on the original image. When the image is displayed after some delay, the display plane is rotated by θ. By method (a), the whole image is shift by *s* such that the center of the rotated image should coincide with that of the original image. By method (b), the rotated image is shifted and resized such that the direction of each end of the image should be the same as the original. By method (c), the direction of arbitrary point on the rotated image coincides with that of the original, so that there is no image distortion.

5 Consideration on Implementation

5.1 Implementation Strategy

The image correction by simple two-dimensional image shift and rotation is most suitable for *local* compensation, i.e., it can be easily implemented on HMD units. Using this method, we can design the processing unit using the following methods:

(1) Digital/electronic implementation: The image is once captured on the video memory, processed on it, and then sent out to the HMD. This is the most flexible way, but we must be careful to time delay of A/D, D/A conversion and image processing.

(2) Analog/electric implementation: The position of the image on the display device is controlled in order to obtain shifted image.

(3) Analog/optical + mechanical implementation: The position and orientation of the image of screen itself is controlled optically.

5.2 Estimation of Error by Simple Implementation

The implementation using simple image shift causes the position error and image distortion, as it is not an exact way to compensate the head rotation. By the image correction method (a), the amount of the image shift is calculated such that the direction of the center of the image should coincide with the original one. Let P_o be a point on the original image, P_s be obtained by translating P_o by s, P_r be obtained by rotating P_s by angle θ, and P_p be obtained by projecting P_r onto the original display plane. The origin is located at the viewing point, and the x-axis is set in the right hand direction. The coordinate of the point P_o is defined as

$$P_o = \begin{bmatrix} x \\ D \end{bmatrix}, \qquad (1)$$

and the angle φ is defined as in Figure 6. Then the shift amount is calculated as

$$s = D \tan\theta . \qquad (2)$$

As the coordinate of P_o can be expressed as

$$x = -D \tan\varphi , \qquad (3)$$

the x-coordinate of the point P_s and its corresponding angle are calculated as

$$x_s = x + s = -D\tan\varphi + D\tan\theta , \qquad (4)$$

$$\varphi_s = -\tan^{-1}\frac{x_s}{D} = \tan^{-1}\left[\tan\varphi - \tan\theta\right]. \qquad (5)$$

Rotating P_s by θ, we get the angle of the point P_r :

$$\varphi_r = \varphi_s + \theta = \tan^{-1}\left[\tan\varphi - \tan\theta\right] + \theta . \qquad (6)$$

Hence the error of the angle correspoinding to the image distortion is

$$\Delta\varphi = \varphi_r - \varphi = \tan^{-1}\left[\tan\varphi - \tan\theta\right] + \theta - \varphi . \qquad (7)$$

If we do not shift the rotated image, this error will be a constant over the whole image:

$$\Delta\varphi_n = \theta . \qquad (8)$$

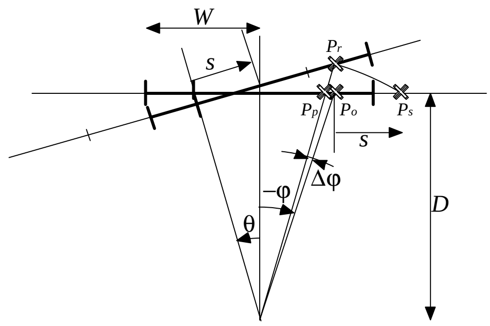


Figure 6 Direction error caused by the image correction by simple shift

Suppose that the operator is rotating his/her head at the rate of 90 degrees per second and the graphics subsystem is rendering the image at the rate of 10 frames per second, then the maximum value of correction angle is 18 degrees, including rendering delay and inter-frame delay. The error of the angle for this case is shown in Figure 7. The horizontal axis shows the angle between the direction to the point on the image and the front direction, and the vertical axis shows the estimated error angle between the point on original image and the corresponding point on the rotated and shifted image. The value of the error becomes larger as the point of interest approaches either end of the image, however, this error is rather small around the center of the image. This result shows that this simple implementation is tolerable for typical tasks, as long as the operator is gazing at the center of the HMD screen.

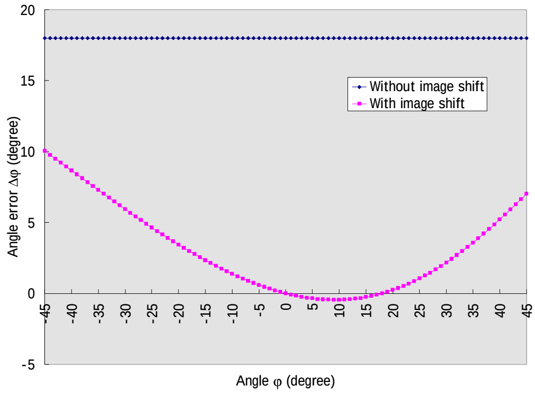


Figure 7 Estimation of error angle of the image caused by simple shift:90 degrees/s, 10 frames/s

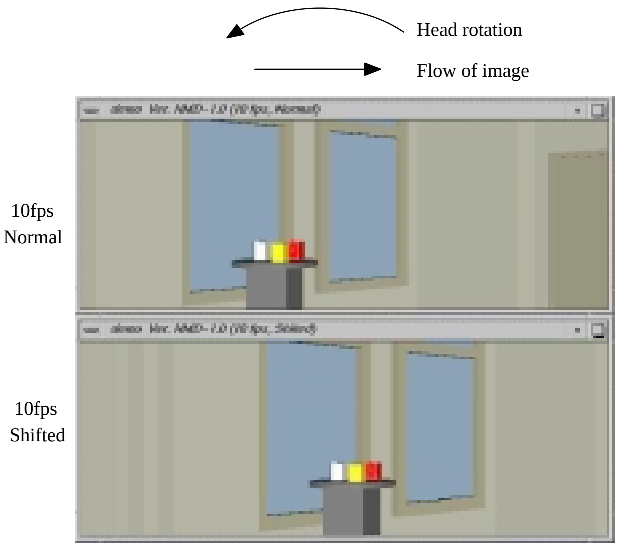


Figure 8 A sample image of real-time simulation

6 Real-time Simulation

A real-time simulation was taken place to show the effect of the proposed method. We used a graphics workstation (Silicon Graphics Indigo2 Maximum IMPACT), which can generate the image of a simple virtual environment at the rate of more than 60Hz, and the video image with low frame rate was simulated by inserting appropriate idle time. An example of graphics image is shown in Figure 8. The normal double-buffered graphics image is displayed in the upper window, and the image sequence corrected by the proposed method is shown in the lower window. At this moment, The operator is turning his/her head to the left and the image of the virtual world should flow to the right. Using our compensation method, the corrected image precedes the normal double-buffered image, which indicates the time delay is being compensated. Also the frame rate of the displayed image is improved to 60Hz, which realized smooth image flow according to the operator's head rotational motion.

7 Conclusion and Future Works

A method to improve the temporal quality of HMD systems is proposed focusing on the uniform optical flow caused by the operator's head rotational motion. Using this method, we can equivalently obtain higher frame rate and lower time delay. Our method is a simple and straightforward approach to the solution of problems with temporal quality for VR/tele-existence systems.

Our simplified implementation is not ideal; it is just an approximation. Our future work includes configuring the overall system to examine the effect of the proposed method.

References

[1] M. Mine: "Characterization of End-to-End Delays in Head-Mounted Display Systems," UNC-CH Dept. of Computer Science technical report TR93-001 (1993)

[2] G. Bishop, H. Fuchs, L. McMillan, E. S. Zagier: "Frameless Rendering: Double Buffering Considered Harmful," Proceedings of Siggraph 94, In Computer Graphics Proceedings, pp. 175-176 (1994)

[3] W. Mark, G. Bishop, L. McMillan: "Post-Rendering 3D Warping," Proceedings of 1997 Symposium on Interactive 3D Graphics, pp.7-16 (1997)

[4] C. Cruz-Neira et al.: Surround-Screen Projection-Based Virtual Reality: The Design and Imple-mentation of the CAVE, Proc. of SIGGRAPH '93, pp.135-142 (1993)

[5] M. Hirose et al.: "A Development of Immersive Multiscreen Display (CABIN)," Proceedings of the Virtual Reality Society of Japan 2nd Annual Conference, pp.137-140 (1997)