July 10, 2009

RT0859
Human-Computer Interaction 6 pages

Research Report

3D Manipulation Device Based on Z-direction Moves of the Intersection of Two-Eye View Directions

Akira Koseki, Kiyokuni Kawachiya

IBM Research, Tokyo Research Laboratory
IBM Japan, Ltd.
1623-14 Shimotsuruma, Yamato
Kanagawa 242-8502, Japan

3D Manipulation Device Based on Z-direction Moves of the Intersection of Two-Eye View Directions

Akira Koseki, Kiyokuni Kawachiya

1. Introduction

The 3D virtual world gives new interaction between human and computers. To make it easier to point a 3D object in the virtual world, existing methods like PUPA6-337756 use the intersection of two-eye view directions. They allow users to point objects not using conventional pointing devices such as keyboards and mice. However, they assume that the intersection should be on the surface of objects. In contrast, presented approach gives more instinctive and immediate interactions using Z-direction moves of the intersection of two-eye view directions.

2. Problem

In the aforementioned methods, the pixel (or pixels when using a two-eye display) at the intersection of two-eye view directions is a point the user is looking at. Such a point usually goes on the surface of an object. However, the two-eye view directions don't necessarily intersect at the pixel (or pixels of two screens that correspond to the same point when using a two-eye display). As described in the figure 1, the two-eye view directions may cross the different pixels and intersect at a far point (in the center), or also cross the different pixels and intersect at a near point (on the right). The users' viewpoints thus move in the Z-direction. Our problem to be solved is then detecting such movements in the Z-direction to trigger actions associated to an object the user is looking at or user's current activity.

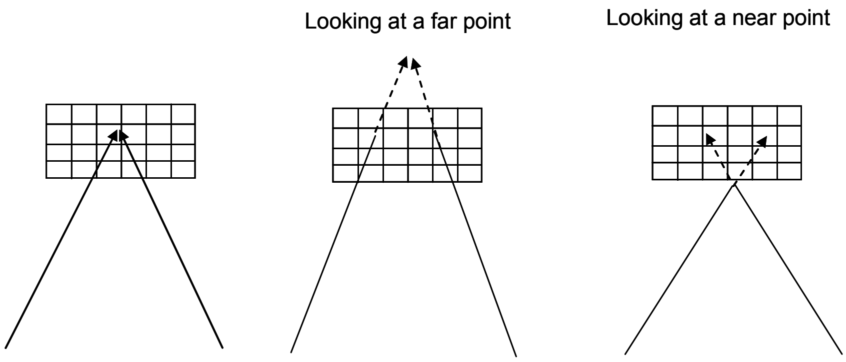

Figure 1

3. Our Approach

Our method detects the change of the direction of two eyes and pixels that the user is looking at on the screen (or screens when using a two-eye display). When the two-eye view directions intersect at the pixel (or pixels that correspond to the same point of the screens when using a two-eye display), the system detects the object that includes such a pixel and set the object to be the "seen object." In the case that the pixel is not included in any object, the system set the "background object" which is a special object in the system to be the "seen object."

When the two-eye view directions move to intersect at different pixels, the system recognize such changes as movements in the Z-direction and trigger appropriate actions associated to the "seen object" or to user's current activity.

Actions triggered based on such Z-direction movements can be provided by object designers but typical ones are as follows:

1. when the user looks at a far point, make the seen object translucent (Figure 2), using an existing algorithm. The object at the intersection point becomes visible by making objects in front of the object translucent.
2. when the user is walking through (or flying), raise the speed of movement when the user looks at a far point, and lower the speed when looking at a near point (Figure 3).
3. when the user looks at a far point, trigger a "push action" of the "seen object" (Figure 4)

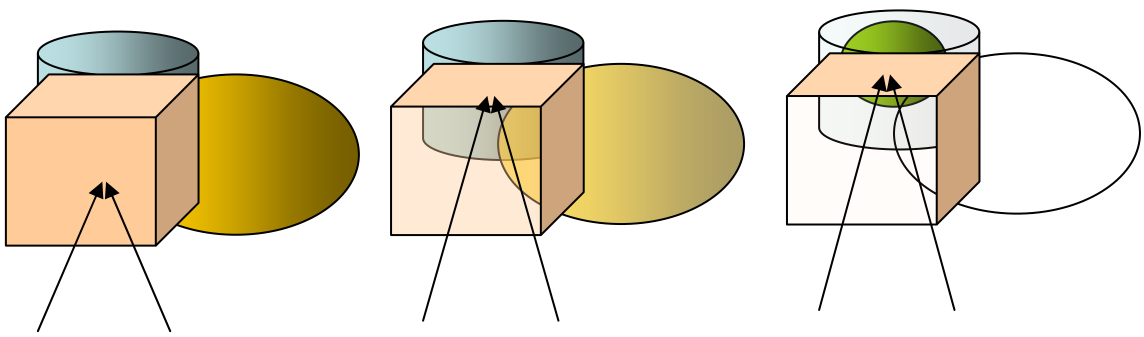

Figure 2

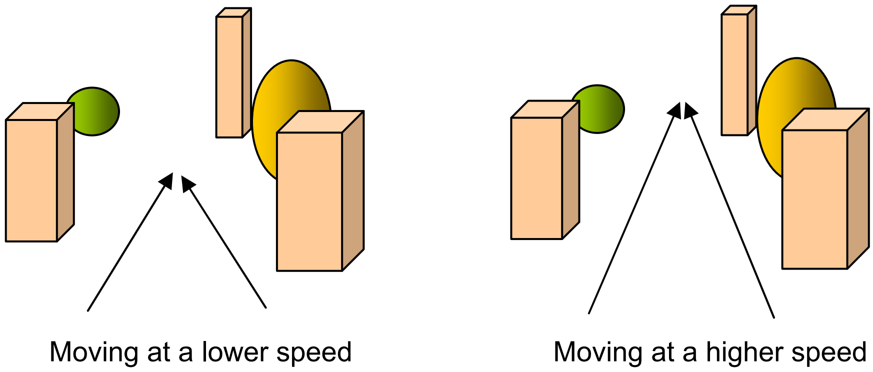

Figure 3

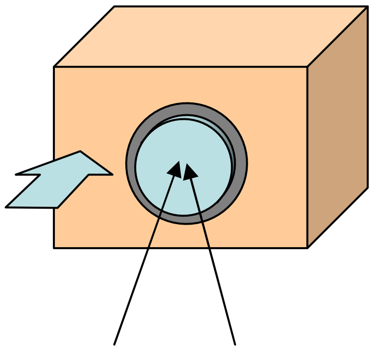

Figure 4

4. Implementation

Figure 5 shows the overview of our system, mainly consisting of the rendering system that draws the screen images of 3D virtual world from the position of an eye with its view direction, and the user device that displays those images and sends the physical direction and position data of the eyes and the face to calculate those direction and position in the virtual world. Main part of the user device has a glasses-like shape and can display the two different images for the eyes. Moves of human body are detected by another user device called positioning device and position data of the eyes and the direction data of the head are sent to the rendering system. The eye direction data are also detected by an internal device of the glasses-shaped device and are sent to the rendering system. The user also sends the navigation info to the rendering system using the navigation device consisting of a keyboard or mouse to rotate or move in the virtual world.

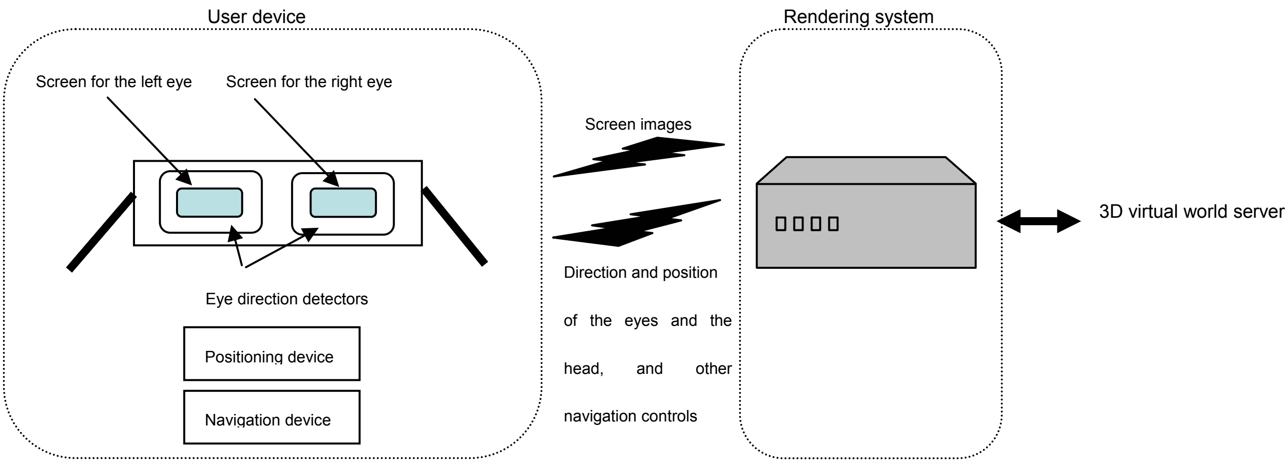

Figure 5

Figure 6 shows the flow chart of the system.

Figure 6

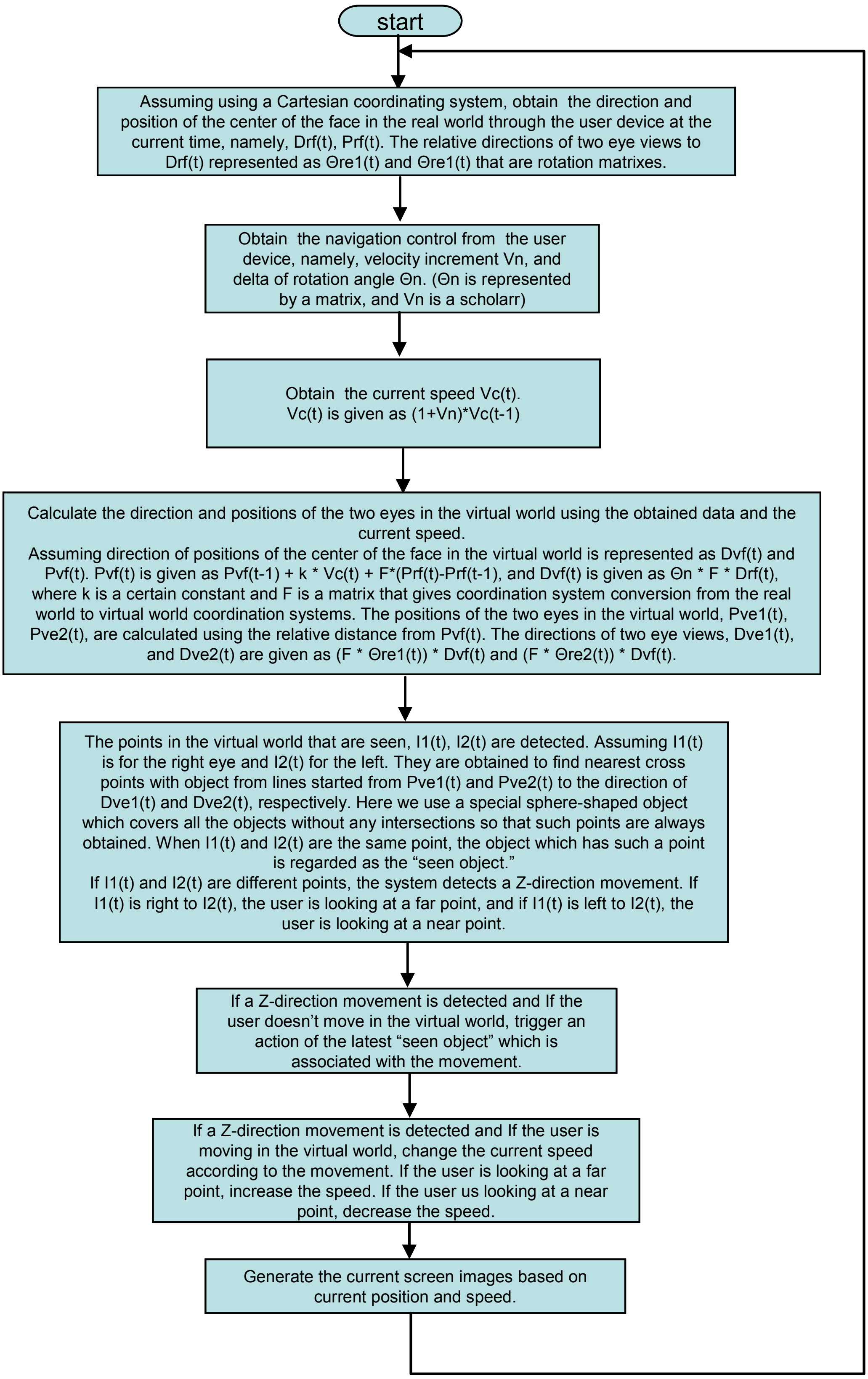

Figure 7 also shows the changes of screen images when a Z-direction movement is detected. First the user looks at the same point which is on the surface of a button object. Next the user looks at a far point, and a “push” action of the button object is trigged and the screen images are then changed to reflect the action.

Figure 7