A NEW IMPLEMENTATION OF HEAD-COUPLED PERSPECTIVE FOR VIRTUAL ARCHITECTURE

MOHAMMAD RAHMANI ASL1, CHENGDE WU2, GIL ROSENTHAL3 and WEI YAN4
1,2,4 Texas A&M University, College Station, United States
{mrah, chdwu22, wyan}@tamu.edu
Texas A&M University, College Station, United States
grosenthal@bio.tamu.edu

Abstract. The process of projecting 3D scenes onto a two-dimensional (2D) surface results in the loss of depth cues, which are essential for immersive experience in the scenes. Various solutions are provided to address this problem, but there are still fundamental issues need to be addressed in the existing approaches for compensating the change in the 2D image due to the change in observer’s position. Existing studies use head-coupled perspective, stereoscopy, and motion parallax methods to achieve a realistic image representation but a true natural image could not be perceived because of the inaccuracy in the calculations. This paper describes in detail an implementation method of the technique to correctly project a 3D virtual environment model onto a 2D surface to yield a more natural interaction with the virtual world. The proposed method overcomes the inaccuracies in the existing head-coupled perspective viewing and can be used with common stereoscopic displays to naturally represent virtual architecture.

Keywords. Virtual reality; virtual architecture; head-coupled perspective; depth perception.

1. Introduction

Traditionally three-dimensional (3D) scenes are rendered by projecting 3D geometry onto a two-dimensional (2D) plane. This process results in the loss of several depth cues fundamental for reconstructing the 3D structure of the scene for the observers’ cognition of the scene (Li et al., 2012). Human brain uses multiple monocular and binocular perceptual cues in different situations to perceive the depth of the 3D world. This makes perception of depth a cen-
tral challenge for visual systems, particularly for observers moving relative to the scene (Nadler et al., 2008).

Correct depth may be perceived from a 2D image rendered from a 3D environment under certain circumstances. However, when the observer moves, some of the depth cues are lost immediately and the image looks flat. In Figure 1 the abovementioned point is illustrated by simulating both real and virtual world in one model. Figure 1-a shows a 3D environment in reality which includes an observer looking at a flat screen. Figure 1-b shows a virtual environment, a cube in the field of view (FOV) of a camera. The camera in the virtual environment is adjusted in a way that the screen in the reality is correctly fit into the FOV of the camera in the virtual environment. Figure 1-c shows the image rendered from the camera in the virtual environment projected onto the screen in the real environment. When the observer moves to the exact point where the camera is set, he/she will perceive a correct 3D space which simulates natural 3D view (Figure 1-d). However, when the observer moves to a place different from camera position, he/she will immediately notice that it is a 2D image projected on the screen (Figure 1-e).

![Figure 1](image)

**Figure 1.** (a) Real environment scene setting, (b) Virtual environment scene setting, (c) Rendered image of virtual environment by the camera, (d) Observer’s view standing at the same position of the camera, (e) Observer’s view when the position is changed.

Different types of observer movement cause different problems in perceiving correct 3D virtual space. The lateral, vertical, and axial movements of the observer as well as its head rotation are demonstrated in the Table 1. As one can see, observer’s lateral and vertical position changes cause incorrect perspective and position. Axial movement causes incorrect perspective, object size, and position. However, rotation by itself does not cause incorrect perception of the 3D virtual space and does not require any correction.

Our study shows that using the conventional head-coupled camera approach (attaching the camera on viewer’s head) does not solve the problem for cases that the observer moves axially towards or away from the screen.
With the conventional settings, if the observer moves toward the screen he/she perceives the object larger than what it should be and if he/she moves away from the screen the objects appears smaller than what it should be.

Table 1. Problem in perceiving correct 3D space based on observer’s movement.

<table>
<thead>
<tr>
<th>Observer Change Position</th>
<th>Observer View</th>
<th>Diagram</th>
<th>Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lateral movement (left-right)</td>
<td><img src="image1" alt="Observer View" /></td>
<td><img src="image2" alt="Diagram" /></td>
<td>Incorrect perspective and position</td>
</tr>
<tr>
<td>Vertical movement (up-down)</td>
<td><img src="image3" alt="Observer View" /></td>
<td><img src="image4" alt="Diagram" /></td>
<td>Incorrect perspective and position</td>
</tr>
<tr>
<td>Axial movement (front-back)</td>
<td><img src="image5" alt="Observer View" /></td>
<td><img src="image6" alt="Diagram" /></td>
<td>Incorrect size, position, and perspective</td>
</tr>
<tr>
<td>Lateral rotation (Look around)</td>
<td><img src="image7" alt="Observer View" /></td>
<td><img src="image8" alt="Diagram" /></td>
<td>N/A - Rotation alone does not cause problem in 3D space perception</td>
</tr>
<tr>
<td>Vertical rotation (Look up/down)</td>
<td><img src="image9" alt="Observer View" /></td>
<td><img src="image10" alt="Diagram" /></td>
<td>N/A - Rotation alone does not cause problem in 3D space perception</td>
</tr>
</tbody>
</table>

Figure 2-a demonstrates an ideal case that the observer can see the virtual world through a window at time zero (t₀). Figures 2-b shows the observer after moving away from the window at time one (t₁). As demonstrated in Figure 2, the projected object size on the window pane changes correctly (from a/2 to 2a/3).
Figure 2. (a) Initial scene setting, (b) The observer perceives the object correctly through the window after moving back from the window.

Figure 3-a demonstrates the mounted camera on observer’s head at time zero (t0). Figure 3-b shows the camera at time one (t1), moved away from the object in the virtual world. Since the projection plane of the camera moves with the camera, the projected object size on the camera project plane is different from the previous case (now it is a/3).

Figure 3. (a) The camera in the virtual world, (b) The camera moved away from object in the virtual world and the size of projected object is properly changed.

Figure 4 shows the observer in the real world in front of a computer screen at t1. The projection plane of the camera is mapped on the screen. The camera projection is made based on that the camera is mounted on the observer’s head. The difference between the projected object size in image 2-b and 4 shows that the traditional head coupled perspective still causes incorrect perception of object size (Figure 2-b shows the correct size while Figure 4 shows the incorrect size).

Figure 4. The camera projection plane is mapped on the screen at t1. The observer does not perceive the correct object size.
The calculation below shows that the observer perceives the size of the object half of its real size using the conventional head-coupled settings. Note that in the calculation below the perceived size of the object at \( t_0 \) is proportional to \( a \), and simply defined to be \( a \).

\[ l_0: \text{Projected object size on window pane and camera plane} = a \times \frac{L}{2L} = \frac{a}{2} \]

\[ l_1: \text{Projected object size on window pane} = a \times \frac{2L}{3L} = \frac{2}{3} \times a \]

\[ \text{Projected object size on camera plane and screen} = a \times \frac{L}{3L} = \frac{a}{3} \]

**Correctly perceived object size through window (Figure 2b)**

\[ \frac{2}{3} \times a \times \left(\frac{3L}{2L}\right) = a \]

**Incorrectly perceived object size using conventional head coupled camera approach (shown in Figure 4)**

\[ \frac{a}{3} \times \left(\frac{3L}{2L}\right) = \frac{a}{2} \]

Therefore the perceived size after the movement is actually 1/2 of the correct size. By the way, without using the head-coupled camera approach (i.e. the camera doesn’t move), the perceived size after the movement is actually 3/4 of the correct size:

**Incorrectly perceived object size by observer**

\[ \frac{a}{2} \times \left(\frac{3L}{2L}\right) = \frac{3}{4}a \]

This research describes in detail an implementation method for head-coupled perspective to correctly perceive 3D virtual space in real-time from rendered images projected on a 2D screen, and further prove the validity of the method.

2. Literature review

The interaction with 3D views is inevitable in dealing with computer applications. Realizing that the human mind uses a set of perceptual cues to perceive depth and understand the structure of 3D environment encouraged researchers to focus on the same set of rules to stimulate the visual system in the same way (Limbacher, 1969). Some of the perceptual cues are perceivable from 2D projections such as occlusion, perspective, familiar size, and atmospheric haze. However, a few of the perceptual cues are lost when 3D scenes are rendered on a 2D surface including binocular parallax, movement parallax, accommodation, and convergence (Neil A. Dodgson, 2005).

Stereoscopic techniques try to simulate binocular parallax by providing different images for each of the viewer’s eyes. Wheatstone (1838) invented the stereoscope that showed three dimensional photographs by presenting slightly different images to each eye and presented the first research results regarding stereoscopic vision (Fauster and Wien, 2007). Stereoscopic 3D technologies include polarized projection, lenticular image, altering field shutter technology, and anaglyph projection (Peris, 2010).
On the other hand, variation in viewing position results in a distorted stereo-oscopic image and causes wrong perception and eye fatigue, which has been discussed in various related works (Son et al., 2002; Kim et al., 2012; Woods et al., 1993). Motion parallax contributes to the naturalness of vision in 2D representation of 3D world and can compensate distortion in the stereoscopic representation technique. There are various studies on stereoscopic viewing and perspective projection coupled to the head position of the observer. Ware et al. (1993) called their model “Fish Tank Virtual Reality” and evaluated 3D task performance. The experiment by Ware and Franck (1994) shows that the effects of stereoscopic 3D and head coupled perspective are additive – each has a positive effect on revealing increased structural information and the effects can be combined.

In the abovementioned research studies stereoscopy and motion parallax are used together in order to achieve a realistic image representation but a true natural image could not be perceived yet. Runde (2000) reported that despite of his expectations, representation of motion parallax considering viewing position to compensate stereoscopic technique artifacts did not gain at the geometrically correct position of virtual camera. He could not find out the reason for this deviation and the problem remained open. Li et al. (2012) reported that having virtual camera corresponded to an assumed head position with simple mapping does not always produce good results. These examples show that there is a fundamental problem in the existing approaches.

Lee (2007) presented the result of a head tracking method to correct the camera perspective issue. However, the implementation method is not explained in the reference. In addition, based on our literature review, no explanation could be found for the implementation method. While studying camera perspectives, we independently developed our own implementation method that can produce correct results and intuitive to understand. The proposed implementation accurately renders a 3D virtual environment onto a 2D surface to yield a natural interaction with the virtual world. This method will overcome the issues in the existing head-coupled perspective viewing and may be used in with common stereoscopic displays to naturally represent the virtual world.

3. Geometrical representation of the proposed method

To explain the proposed model a geometrical representation of virtual camera, screen, and the eye of the observer is created. Since the position difference between the camera and the observer causes incorrect space perception, in our proposed model the virtual camera always follows the observer.
We hypothesize that the 3D space can always be correctly presented with respect to the observer’s movement by (1) pointing the camera at the imaginary screen (the area bounded by P1, P2, P3, and P4 in Figure 5 left) in the virtual world (the imaginary screen represents the computer screen in the real world), (2) rendering the camera perspective image (Figure 5 left), (3) reversing the perspective projection of the screen (the area bounded by P1, P2, P3, and P4) back to a rectangular image (Figure 5 right), and (4) displaying the rectangular image on the computer screen.

4. Visual Simulation and Proof

To visually cross-check the validity of the theory, we developed a simulation tool in Unity Pro. In this simulation model virtual and real environments were modelled in a single Unity scene sharing the same coordinate system, but separated by layers.

4.1. REAL AND VIRTUAL ENVIRONMENTS SCENE SETTINGS

The real environment includes an observer, a wall shown in light grey, and a screen hanging on the wall in dark grey (Figure 6-a). A red dot at the left bottom corner of the screen denotes the origin of the coordinate system in the real scene. The floor is mapped with checker pattern for better visual confirmation of consistency with the virtual scene. The virtual world includes a 3D model of a hotel lobby and a camera (Figure 6-b). The red dot denotes the origin of the coordinate system that virtual world is based on.
When the observer in the real scene moves, the camera in the virtual scene always follows the observer. The camera will process the image on the camera projection plane according to the method that we hypothesized above, and project final image to the hanging screen in the real scene. The goal is to make the screen to act like an opening that will seamlessly connect the virtual world - hotel lobby and the real world. This connection should be visually examined by looking at the checker patterns in the two worlds.

4.2. SYSTEM CONFIGURATION

Two key points in this simulation system are perspective transformation and interpolation. For perspective transformation we used WorldToScreenPoint function of the camera class in Unity. This function uses matrix transform to convert a point from the world coordinate system to the camera coordinate system. Transforming non-rectangular area to rectangular area usually involves pixel interpolation because the area of the final rectangular shape is usually designed to be larger than the non-rectangular quadrilateral shape in order to maintain image quality. The interpolation process drops the performance of the simulation system. Therefore, we used the Inverse Mapping method, which means mapping each pixel coordinates of the rectangular shape to the non-rectangular shape and use the colour of the mapped pixel on the non-rectangular shape as the colour of the pixel on the rectangular shape. This is faster than transforming each pixel from the non-rectangular shape to the rectangular shape. The Inverse Mapping method automatically uses the mapped pixel colour instead of interpolation and results in faster simulation.

4.3. DEMONSTRATION

The simulation results from changing position of the observer are provided in Table 2. The rendered image projected on the screen perfectly matches the real environment all the time, as if a hotel lobby behind the wall can be seen through an opening. This result can be seen clearly by examining the continuity of the checker pattern on the floor.

<table>
<thead>
<tr>
<th>Movement</th>
<th>Observer/Camera Position</th>
<th>Observer’s View</th>
<th>Notes</th>
</tr>
</thead>
</table>

Table 2. Simulation of observer movement and modified view projection on screen.
5. Conclusion

In this research a head-coupled perspective method is implemented to provide users with a dynamic and realistic 3D perspective for environmental models. Conventional perspective projection methods have flaws that observers are not able to correctly perceive the depth from a rendered 2D image when they change their position. The method proposed in this research solves this problem and enable observers to perceive depth correctly in real-time regardless of their type of movement. The visual verification of the proposed method is provided in detail (due to the limitation of the paper length, the geometrical proof will be given in a separate venue). The advantages of this method are:

- The observer doesn’t have to always look at the centre of the screen.
- The observer’s FOV does not influence calculating result.
- Virtual camera and the observer can have different FOV and orientations.
- Correct depth perception can be achieved for all observer movements.

While the method has a limitation that the image projected on the screen only can be correctly perceived by a single observer, the implementation method may be applied to various fields such as architectural design review.
(in which the perception of spatial / object distance and size is important), visual experiments for behaviour study (in which the subjects can correctly perceive the object size in the virtual environment), video games, etc.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. IOS-1045226.

References


