Naked Eye Pseudo 3D Display Technology Outside the Screen

Ziyao Wang¹,* and Haikun Wei¹
¹ School of Automation, Southeast University, China

*zy_wang@seu.edu.cn

Abstract. The 3D display technology based on motion parallax has the advantages of no need to wear glasses and has no limitation on the viewing angle. However, the current problems of this method include 3D scenes and models can only be displayed within the screen; the system is sensitive to light. For the first problem, we proposed the concept of virtual bezel, which makes out system could produce a strong illusion that objects displayed outside the screen, and greatly increases the visual impact of 3D display. For the problem of light sensitivity, unlike other system using RGB camera, we use RGB-D images to detect and track the viewer’s face, which makes our system work properly even at night.

1. Introduction
3D display technology, with its ability to restore the real world in the most realistic way and brings the strongest visual impact to users, has been the direction of continuous efforts of many scientific researchers[1,2]. Currently, the most popular commercial 3D display technology uses stereoscopic parallax to generate depth perception, such as using masks, polarization or shutter glasses[3,4].

However, this solution requires users to wear special glasses. In order to remove glasses restriction, naked-eye 3D display technology emerged. Such systems are based on spatial multiplexing techniques or spatiotemporal multiplexing techniques[5], typically using a parallax barrier[6,7,8] or a lenticular lens[9] as a spatial multiplexing component, introducing a parallax barrier or a lenticular lens between the LCD and the viewer. This allows the viewer's left and right eyes to receive corresponding images respectively, thereby obtaining depth perception.

In daily life, the factors of human perception of depth include geometry, shadows, occlusion, stereo parallax, and motion parallax[10,11]. The technical solution using only stereoscopic disparity has many limitations, such as users need to be in a fixed position; in addition, the content that users see in different perspectives will not change, which is different from people's daily experience. The reason for this problem is the lack of motion parallax. It has been pointed out in experimental psychology that motion parallax is an important part of human perception of depth[12].

The 3D display technology based on motion parallax has the advantages of no need to wear glasses and has no limitation on the viewing angle. The basic principle to form motion parallax is to track the viewer's face and adjust the display content through feedback [13,14,15,16,17,18]. However, the current problems of this method include 3D scenes and models can only be displayed within the screen; the system is sensitive to light.

Our system follows the common basic principle to form the motion parallax. Mainly for the problem that 3D scenes and models are confined to the screen, we proposed the concept of virtual bezel and further utilize the principle of occlusion[19,20]. Which makes the system able to produce a strong illusion of objects displayed outside the display, greatly increases the visual impact of 3D
display. For the problem of light sensitivity, unlike other systems using RGB cameras, our system uses RGB-D images for face detection and tracking, which allows our system to work properly even at night.

2. Related work

2.1. Motion Parallax and Its Applications
Motion parallax is one of the most important factors in the process of constructing depth perception[12]. By observing the relative motion between different objects, people can obtain spatial information of different objects. Besides, people can obtain more details of the same object from different perspectives due to the different viewing angles brought by the movement.

Harrison and Hudson[13] designed a webcam-based remote video conferencing system. The system uses a single camera to capture the sender's scene, uses the pre-extracted background to separate the foreground person from the background. The receiver stretches the foreground and background pictures transmitted from the sender according to the viewer's angle relative to the display, reconstructing a new image and displaying it. Zhang et al. [14] used a time-of-flight camera to separate the foreground from the background, and a matting strategy [21] is used to improve the composition result. So the system is more robust. However, the common problem of these two systems is that their models are simple picture stretching and misalignment. Although they can provide a certain motion parallax, it lacks the details brought by different viewing angles and lacks realism.

2.2. 3D scene and model construction
Using foreground and background image stretching[13,14], the model is too flat and lacks realism. On the basis of depth perception generated by motion parallax, using 3D model scenes will further enhance depth perception.

KinectFusion [22,23] uses Kinect provided a real-time and stable room-scale modeling technique based on the GPU, and it can model the human body. But its disadvantage is that the memory occupancy rate is too high, and the model noise is large. Zhu et al. [15] uses Kinect in model building and uses parametric methods to model the human body. The model obtained by this method is faster and of better quality.

These types of modeling methods based on real-world scenarios often fail to meet the desired requirements in terms of speed and detail. With the development of computer graphics in recent years, there have been many game engines[24], such as Crystal Space, Unity3D, and Vision Engine. The game engine allows developers only need to consider the relationships between models when building 3D scenes, regardless of the details of the model itself. The construction of the model can use other specialized software, which makes the constructed model universal.

2.3. Occlusion and depth perception
In traditional painting art creation, the use of geometric shapes, shadows, occlusion, etc. enables people to obtain the depth perception on the 2D screen[19,20]. Through the occlusion relationship of different objects, we can easily distinguish the distance of objects, and can further produce the depth perception.

Rodrigo et al. [16] proposed the GridScape method for the screen discontinuity caused by the bezels of various LCD panels when studying wall-size displays composed of multiple LCD panel tiles. Through creating the motion parallax, the picture discontinuity caused by the original LCD panel bezel is converted to the effect of a floor-to-ceiling window frame. In addition, the screen of the display has an inward depth when it is referenced to the "floor-to-ceiling window frame".

Ian Stavness et al. [18] developed pCubee, a cube display system composed of five displays and based on motion parallax. The overall effect of the system is like a transparent box with bezels. By simulating the bezel which is covered on the other side when rendering each display's content, the viewer can obtain a strong depth perception.
Both of these work[16,18]enhanced the three-dimensional effect brought by the motion parallax through the occlusion effect caused by the bezel. The difference is that GridScape [16] uses real bezels, and pCubee [18] simulates bezels that should be seen. However, these works only create a depth perception within the screen. This paper will create a strong illusion that the model is displayed outside the display by setting a special virtual bezel.

3. System overview
Our system mainly includes a Kinect camera and a display. As shown in figure 1. Firstly, we use Kinect to obtain the position of viewer's face. Then use it to determine the direction of the viewer's line-of-sight. Finally, adjust the position and attitude of the camera model in the 3D scene by the direction of viewer's line-of-sight, and then the adjustment of the screen content is completed.

![System overview](image)

Figure 1. System overview.

In the next few chapters, we will first analyse the tools we used and their advantages briefly in Chapter 4. In Chapter 5, we will focus on the construction of virtual bezel and how the illusion of objects appearing outside the display. In addition, The derivation of the camera model's position and attitude will be given in this chapter. Chapter 6 will show the effectiveness of our system. Chapter 7 will give the summary and outlook of our work.

4. Related tools

4.1. Face Detection and Tracking using RGB-D Images Based on Kinect
Kinect is a motion-sensing input device produced by Microsoft. It not only can obtain traditional color images, but also can acquire depth images through depth sensor, thus forming RGB-D images. The depth sensor consists of an infrared laser projector and a monochrome CMOS sensor and it can captures depth data in any ambient light conditions[25].

The SDK developed by Microsoft for Kinect supports the detection and positioning of human faces. We use this SDK to detect and locate human faces. Figure 2 shows the three-dimensional coordinate system used by Kinect to locate the target. Based on Kinect, we can easily locate faces in real time and get exact 3D coordinates. In addition, the use of Kinect makes our system unaffected by light.

![Coordinate system](image)

Figure 2. The three-dimensional coordinate system used by Kinect to locate the target. The origin of the coordinate is at the center of the depth sensor, and the coordinate unit is meter.
4.2. **Unity3d-based 3D scene and model construction**

Unity3d is a cross-platform game engine[26] and it supports 2D and 3D graphics editing. This engine encapsulates a number of bottom-level graphics APIs: DirectX, OpenGL, WebGL, and many other components, such as rendering, audio, and physics. Based on Unity3d, the original screen rendering work will be simplified to control the position and attitude of the camera model. The virtual bezel described later is also based on the modification of the camera model.

5. **Content of the screen**

5.1. **Occlusion and depth perception**

Simulating the occlusion relationship between different objects on the 2D plane makes it easy to determine the distance between different objects. Figure 3 shows an example of creating depth perception using occlusion relationships. Against the gray background, two identical trees are placed in this two pictures. The left picture gives the impression that the two trees are in the same plane and are very flat. The right picture is based on the left picture, and add a number of white lines. It can be clearly felt that the two trees produce a sense of hierarchy.

![Figure 3. An example of using occlusion to create depth perception](image)

5.2. **Virtual bezel**

Using the occlusion relationship shown in figure 3, by adding a few lines, we can create the illusion that objects are displayed outside the screen. However, the lines destroy the content of the original picture. In order to create the illusion without destroying the content as much as possible, we can construct the virtual bezel.

When we use the narrow-bezel display in daily life, we can see the scene as shown in figure 4. The black border in the figure is the real bezel of the display. The screen with white background shows a tree. From our daily experience, we know that the tree will never exceed the bezel and is always within the display.

![Figure 4. Scenes in real life when we see a narrow bezel display. The black border in the figure is the real bezel of the display](image)
depth perception of different objects with the least content destroy, but also can generate the illusion that the object is displayed outside the screen.

![Display bezel](image)

**Figure 5.** Add virtual bezels on the original display

The effect brought by the virtual bezel is shown in Figure 6. The left part is the effect of adding a virtual bezel on the basis of figure 4. The right part is under the effect of a virtual bezel, producing the effects of hierarchy and illusion. Through displaying or occluding the tree’s partial structure in the virtual bezel area, we can obtain this feeling intuitively.

![Virtual display bezel](image)

**Figure 6.** The illusion that the object is displayed outside the screen

5.3. Camera model control

In this system, firstly, as shown in figure 7, we use the screen center as the origin and the screen plane as the X-Y plane to form the right-handed coordinate system. When we observe an object on the desktop, it can be approximated that we take it as the center and observe it at different locations. Therefore, in our system, when we overlap the center of the displayed object model with the coordinate origin, the viewer's observation of the object model can be taken as the viewer's observation at different positions around the coordinates origin.

![Three-dimensional coordinate system](image)

**Figure 7.** A three-dimensional coordinate system consisting of a display plane. The orange line between the viewer and the origin of the coordinate is considered as the viewer's line-of-sight.
The orange line is considered as the viewer’s line-of-sight. By using Kinect to detect and track the viewer’s face, it is easy to obtain the vector representation of the viewer’s line-of-sight:

\[ \mathbf{n}_f = (-x_f, -y_f, -z_f) \]  \hspace{1cm} (1)

\(x_f, y_f, z_f\) are the coordinates of the viewer’s face captured by Kinect in the three-dimensional space coordinate system constructed for the display.

Since we take the object as the center and the observation distance has the minimum limitation, we can ignore the impact of depth of field, and assume that the content seen in the same direction is approximately the same. After normalizing \(\mathbf{n}_f\), we can obtain the direction of the viewer’s line-of-sight \(\mathbf{n}_{f0}\), then we can adjust the content of the screen using \(\mathbf{n}_{f0}\).

In Unity3d, the content of the screen is determined by the camera model’s basic parameters, position, and attitude. Figure 8 shows a camera model with determined basic parameters and its local coordinate system. The three-dimensional object in the vertebral body is finally imaged on the screen through transmission projection. The blue Z-axis is the line-of-sight direction of the camera model.

**Figure 8.** An example of a camera model. By controlling the position and attitude of the camera model directly, the content of the screen can be adjusted.

**Figure 9.** The position of the camera model changes on the spherical surface centered on the model center. The black border is the virtual bezel model.

Since the same viewer’s line-of-sight direction \(\mathbf{n}_{f0}\) determines the same screen content, the Z-axis of the camera model, ie, the line-of-sight (The Z-axis in the local coordinate system of the camera model) always points to the center of the object model, and in the same direction as the viewer's line-of-sight. The camera model's movement area is a spherical surface centered on the object model center. In order to ensure that the screen content does not rotate in the direction of the line-of-sight, it is necessary to ensure that the red X-axis of the camera model (the X-axis in the local coordinate system of the camera model) is always horizontal. So when adjusting the position and attitude of the camera model, the following three conditions should be met:

1. The position of the camera model changes on the spherical surface centered on the model center;
2. The direction of the camera model’s line-of-sight, ie, the Z-axis, always points to the center of the object model and is in the same direction as the viewer's line-of-sight.
3. The X-axis of the camera model is always horizontal.

As shown in figure 9, a three-dimensional coordinate system is constructed with taken the model center as the origin. We can represent the camera model’s initial position using vector \(\mathbf{P}_0 = (0,0,-R)\) and the initial attitude using quaternion \(\mathbf{q}_0 = (1,(0,0,0))\). Here \(R\) is the radius of the sphere, a preset constant.

In the initial attitude \(\mathbf{q}_0\) of the camera model, the initial direction of each coordinate axis can be represented by \(\mathbf{n}_x = (1,0,0), \mathbf{n}_y = (0,1,0)\) and \(\mathbf{n}_z = (0,0,1)\), which fully satisfies the above three conditions.
The direction of camera model's line-of-sight is the same as the direction of the viewer's line-of-sight $\mathbf{n}_f$. Since the coordinate system where $\mathbf{n}_f$ belongs to is the right-handed coordinate system but the left-handed coordinate system is used in the model world, the coordinate transformation to $\mathbf{n}_f$ is needed. After the coordinate transformation, we can get the desired line-of-sight direction of the camera model, that is, the direction of the Z-axis of the camera model $\mathbf{n}'_z$:

$$\mathbf{n}'_z = \frac{(-x_f, -y_f, z_f)}{\|\mathbf{n}_f\|} = (z_x, z_y, z_z)$$  \hspace{1cm} (2)

From condition 3, we can get the directions $\mathbf{n}'_x$ and $\mathbf{n}'_y$ of X-axis and Y-axis which belong to the adjusted camera model:

$$\mathbf{n}'_x = \frac{(z_x, 0, -z_y)}{\|(z_x, 0, -z_y)\|}$$  \hspace{1cm} (3)

$$\mathbf{n}'_y = \mathbf{n}'_x \times \mathbf{n}'_x$$  \hspace{1cm} (4)

From conditions 1 and 2, we can get the position $\mathbf{P}_t$ of the adjusted camera model:

$$\mathbf{P}_t = -\mathbf{R} \cdot \mathbf{n}'_z$$  \hspace{1cm} (5)

Next, we will calculate the attitude of the adjusted camera model. We can set the attitude of the adjusted camera model as $\mathbf{q}_t = (s, \lambda \cdot \mathbf{r})$. Relative to the initial posture, the rotation amount is $\mathbf{\Delta q}$:

$$\mathbf{\Delta q} = \mathbf{q}_t \cdot \mathbf{q}_0^{-1} = (s, \lambda \cdot \mathbf{r})$$  \hspace{1cm} (6)

Where $\mathbf{r}$ is the direction of rotation axis. Since $\mathbf{\Delta q}$ is a quaternion representing rotation, the following relationship is satisfied:

$$s^2 + \lambda^2 = 1$$  \hspace{1cm} (7)

$$\|\mathbf{r}\| = 1$$  \hspace{1cm} (8)

$$\lambda > 0$$  \hspace{1cm} (9)

$\mathbf{n}_x$ becomes $\mathbf{n}'_x$ after $\mathbf{\Delta q}$ rotation, so there is equation (10)

$$(0, \mathbf{n}'_x) = \mathbf{\Delta q} * (0, \mathbf{n}_x) * \mathbf{\Delta q}^{-1}$$  \hspace{1cm} (10)

Let both sides of equation (10) be right-multiplied by $\mathbf{q}$ , equation (11) and equation (12) can be established.

$$(\mathbf{n}'_x - \mathbf{n}_x) \cdot \mathbf{r} = 0$$  \hspace{1cm} (11)

$$s(\mathbf{n}'_x - \mathbf{n}_x) + \lambda (\mathbf{n}'_x + \mathbf{n}_x) \times \mathbf{r} = 0$$  \hspace{1cm} (12)

Equation (11) and equation (12) also hold true for the camera model Y-axis and Z-axis. From equation (11), when the variation of each axis direction vector is not 0, the rotation axis $\mathbf{r}$ is perpendicular to the change amount of each axis direction vector. To take two non-zero change amount, such as the X-axis and Y-axis, according to the left-hand rotation order, we can calculate $\mathbf{r}$:

$$\mathbf{r} = \frac{(\mathbf{n}'_x - \mathbf{n}_x) \times (\mathbf{n}'_y - \mathbf{n}_y)}{\|(\mathbf{n}'_x - \mathbf{n}_x) \times (\mathbf{n}'_y - \mathbf{n}_y)\|}$$  \hspace{1cm} (13)

If we use equation (12), we can get s when the X-axis variation is not zero.

$$s = -\frac{(\mathbf{n}'_x - \mathbf{n}_x) \cdot [(\mathbf{n}'_x + \mathbf{n}_x) \times \mathbf{r}]}{(\mathbf{n}'_x - \mathbf{n}_x) \cdot (\mathbf{n}'_x - \mathbf{n}_x) - \lambda}$$  \hspace{1cm} (14)

Equation (14) also holds true for the Y-axis and Z-axis. If the X-axis variation is zero, similar equations can be obtained using the Y-axis or Z-axis. Through the simultaneous equations (7) (9) (14), the attitude $\mathbf{q}_t$ of the adjusted camera model can be obtained.
6. Experiments and evaluation
Our system uses Kinect for face detection and tracking and the screen content developed based on Unity3d. We used the robot model and the tree model respectively for experiments. Figure 10 shows the experimental results using a robot model, which shows the effects of different viewing angles and different distances. Under different viewing angles, the details we see are different, and the illusion that the model floats outside the display can be clearly obtained.

![Figure 10. Experiment results using a robot model](image1)

![Figure 11. Experiment results using a tree model](image2)

Figure 11 shows the experimental results using the tree model. We can also obtain the illusion that the model floats outside the display. This illusion is more intense than the robot model. This is because the tree model not only occludes the virtual bezels on the upper and lower sides but also occludes the left and right sides of the virtual bezels. Compared with the robot model, the tree model has a larger occlusion area for the virtual border.

7. Summary and outlook
This paper proposes a naked-eye pseudo-3D technology which can produce a strong illusion of objects displayed outside the screen. Our system is based on motion parallax, combined with the use of our proposed virtual bezels, allowing us to get this illusion on ordinary 2D displays. Our system uses Kinect to detect and track viewer's face based on RGB-D images, so it will not affected by the changes of light. In terms of screen content, we developed on the basis of Unity3d, using the concept of various models to control, so it has a stronger scalability.

The virtual bezel we proposed is the simulation of the real display bezels. How to obtain a more realistic simulation effect still needs further improvement. Besides, our system is more suitable for large-screen and dark environments, because it is easy to simulate a more effective virtual bezel, such as a larger virtual bezel area and more difficult to distinguish from the real bezel.

In terms of content, our system can make dynamic content based on game engines, which makes our technology has a promising application prospect in the advertising field.

References
[1] Tay S, Blanche P A, Voorakaranam R, Tunç A V, Lin W and Rokutanda S 2008 An updatable holographic three-dimensional display Nature 451(7179) 694
[2] Wan W, Qiao W, Huang W, Zhu M, Ye Y and Chen X 2017 Multiview holographic 3d dynamic display by combining a nano-grating patterned phase plate and lcd Opt. Express 25(2) 1114-22
[3] Dodgson N A 2005 Autostereoscopic 3d displays Computer 38(8) 31-6
[4] Konrad J and Halle M 2007 3-d displays and signal processing Signal Process. Magazine IEEE 24(6) 97-111
[5] J Johnson P V, Kim J and Banks M S 2015 Stereoscopic 3d display technique using spatiotemporal interlacing has improved spatial and temporal properties Opt. Express 23(7) 9252
[6] Mphepo W, Huang Y P and Shieh H P D 2010 Enhancing the brightness of parallax barrier based 3d flat panel mobile displays without compromising power consumption J. Disp. Technol. 6(2) 60-4
[7] Kim S K, Yoon K H, Yoon S K and Ju H 2015 Parallax barrier engineering for image quality improvement in an autostereoscopic 3d display Opt. Express 23(10) 13230
[8] Yoon K H, Ju H, Kwon H, Park I and Kim S K 2016 Diffraction effects incorporated design of a parallax barrier for a high-density multi-view autostereoscopic 3d display Opt. Express 24(4) 4057
[9] Kim H, Hahn J and Choi H J 2011 Numerical investigation on the viewing angle of a lenticular three-dimensional display with a triplet lens array Appl. Optics 50(11) 1534-40
[10] Allison R S, Rogers B J and Bradshaw M F 2003 Geometric and induced effects in binocular stereopsis and motion parallax Vision Res. 43(17) 1879-93
[11] Donaldson J K, Weinberger A J, Gagne J, Faherty J K, Boss A P and Keiser S A 2016 New parallaxes and a convergence analysis for the tw hya association Astrophys. J. 833(1)
[12] Gibson E J, Gibson J J, Smith O W and Flock H 1959 Motion parallax as a determinant of perceived depth J. Exp. Psychol 8(1) 40-51
[13] Harrison C and Hudson S E 2009 Pseudo-3d video conferencing with a generic webcam 10th IEEE Int. Symp. on Multimedia 236-41
[14] Zhang C, Yin Z and Florencio D 2009 Improving depth perception with motion parallax and its application in teleconferencing IEEE Int. Workshop on Multimedia Signal Processing Vol 0708.1150 pp 1-6
[15] Zhu Z, Martin R R, Pepperell R and Burleigh A 2016 3d modeling and motion parallax for improved videoconferencing Computational visual media 2(2) 131-42
[16] De Almeida R A, Pillias C, Pietriga E and Cubaud P 2012 Looking behind bezels: french windows for wall displays Int. Working Conf. on Advanced Visual Interfaces Vol 70 pp 124-31
[17] Zhang C, Yin Z and Florencio D 2009 Improving depth perception with motion parallax and its application in teleconferencing IEEE Int. Workshop on Multimedia Signal Processing Vol 0708.1150 pp 1-6
[18] Stavness I, Lam B and Fels S 2010 pCube: a perspective-corrected handheld cubic display Int. Conf. on Human Factors in Computing Systems (CHI 2010 Atlanta Georgia Usa April ) Vol 19 pp 1381-90
[19] Pirenne M H 1970 Optics, painting and photography Optics Painting and Photography
[20] Solso R L 1996 Cognition and the visual arts MIT Press
[21] Levin A, Lischinski D and Weiss Y 2006 A closed form solution to natural image matting IEEE Computer Society Conf. on Computer Vision and Pattern Recognition Vol 30 pp 61-8
[22] Izadi S, Kim D, Hilliges O, Molyneaux D, Newcombe R and Kohli P 2011 KinectFusion:real-time 3d reconstruction and interaction using a moving depth camera ACM Symp. on User Interface Software and Technology ( Santa Barbara Ca Usa October) pp 559-68
[23] Newcombe R A, Izadi S, Hilliges O, Molyneaux D, Kim D and Davison A J 2012 KinectFusion: real-time dense surface mapping and tracking IEEE Int. Symp. on Mixed and Augmented Reality pp 127-36
[24] The introduce of game engine, web site:https://en.wikipedia.org/wiki/Game_engine#References
[25] The introduce of game Kinect, web site:https://en.wikipedia.org/wiki/Kinect
[26] The introduce of game Unity3d, web site:https://en.wikipedia.org/wiki/Unity_(game_engine)