Projection Simulator to Support Design Development of Spherical Immersive Display

Wataru Hashimoto[(✉)], Yasuharu Mizutani, and Satoshi Nishiguchi

Faculty of Information Sciences and Technology, Osaka Institute of Technology,
1-79-1 Kitayama, Hirakata City, Osaka 573-0196, Japan
{wataru.hashimoto,yasuharu.mizutani,
satoshi.nishiguchi}@oit.ac.jp

Abstract. This research aims to develop a simulator that supports the construction of a spherical immersive display, which is a system that can provide a realistic presence, as if the user exists in another space. In general, when developing a display, it is necessary to perform optical design of the projection system in considering special distortion correction on the dome screen. However, accuracy of the optical system that is actually manufactured is not guaranteed to be when it is simulated, and fine adjustment is again necessary when the display is used. In this research, we report on the development of a projection simulator that can perform optical system adjustment and distortion correction simultaneously during optical design of the projection system.

Keywords: Spatially immersive display · Optical design · Image registration · Distortion correction

1 Introduction

Immersive projection display is one of the most powerful systems for enhancing spatial presence and experienced realism. Cave-like immersive displays are increasingly being used for immersive Virtual Reality(VR) applications. A major problem with immersive displays is that they cannot be installed easily, due to the large amount of space required by the surrounding screens. In order to consider the space utility and reduce the volume, a fisheye lens and convex mirror, for example, are utilized to shorten the projection distance. An ultra-short focus projector is also an appropriate device for decreasing spatial cost. It is necessary to simulate the optical system of the projection display; however, optical simulation with certain special lenses is too complex to be easily implemented. Since the optical system is delicate in terms of installation precision, the designed projection cannot always be obtained. After assembling several pieces of display equipment, fine adjustments on the screen are necessary to correct the error due to the optical system for accurate projection. Furthermore, it is necessary to apply the distortion of the image as a result of the optical system to the display content.

The purpose of this research is to achieve design of an immersive display without difficulty, and to develop an environment that can correct error due to the optical system. We focus on the spherical immersive display (Fig. 1), which can be realized on an average room size scale among immersive displays, and organize the spherical

immersive display's development flow to extract problems that are obstacles to its development. Moreover, we aim to develop a simulator that can build a device without requiring special knowledge or experience.
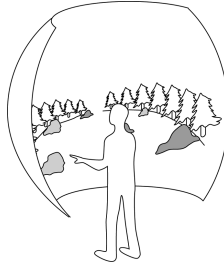

Fig. 1. Spherical immersive display

2   Spherical Immersive Display Features

A planetarium is one of the most similar systems to a spherical immersive display. Since the projection technology of the planetarium was established, many technical resources [1] have been discussed for projecting onto the half-spherical dome. Recently, there have been several examples of the DIY planetarium, due to the widespread use of projectors or other such optical devices. Although the planetarium can be classified as a spherical immersive display, the display utilized in the VR field has the following features.

- *Interactivity*

    The user can interact with the image; therefore, the projected image is not pre-rendered like those in movies, but must be generated immediately.

- *Rear projection*

    Rear projection is suitable for surrounding the user completely with a screen, and can also prevent the user from casting a shadow on the screen.

- *Viewpoint position dependency*

    As the curvature radius of the spherical screen is shortened, the user begins to feel the distortion of the projected image on the screen's surface; therefore, the viewpoint position dependency results in restriction of the user's movement.

    Based on the above features, we organize the development flow of the spherical immersive display.

3   Spherical Display Development Flow

The development flow classification of the spherical display can be roughly divided into the following four areas.

1. Optical design

Optical simulation is necessary in order to project the image onto the spherical surface correctly. The simulation determines, among others, the size, shape, and placement of the screen, as well as the direction and focus distance of the projector. Points to consider in optical design include whether the projected image is focused on the screen, the projector light is blocked by the user or structure, and the projector pixels are used effectively. It is also necessary to focus on the seam of the projected image when multiple projectors are utilized simultaneously. Since the optical simulation can also predict the distortion of the projected image onto a curved screen, a corresponding table for distortion compensation, utilized in flow 4, is generated through the simulation.

2. Spherical screen prototype

A spherical screen and its optical system are built based on the results of flow 1. It is extremely difficult to form a spherical screen with precision. In general, FRP molding or a scraping of polystyrene mass is adopted to form a large curvature radius for the spherical screen.

3. Adjustment

The prototype display optical system usually contains undesirable manufacturing error. Since the optical system is delicate in terms of error, it is impossible to obtain an ideal design result. Figure 2 shows the grid (latitude/longitude) computed by the optical design described in flow 1. The grid in the left-hand image is clearly shifted from the ideal position shown in right-hand picture, due to manufacturing error in the optical system. Therefore, it is necessary to physically move the screen or projector and adjust the grid position on the screen in order to approach the ideal state shown on the right. Several studies (for example, [3]) use external cameras to observe the projection state for automatically adjusting the grid position.
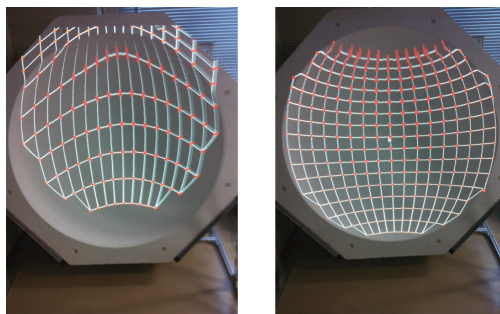

Fig. 2. Result of fine alignments for domed screen, shown on the right. Manufacturing error causes some misalignments, as shown on the left

4. Visual content creation

In order to project visual content onto the spherical surface accurately, a pre-distorted image must be generated, for which the two-pass rendering method is a

suitable approach. The rendering requires the use of a corresponding table, obtained by the adjustment described in flow 3. A rendering method using a game engine such as Unity has also been proposed [4].

The adjustment mentioned in flow 3 is an essential process, due to the error involved in the manufacturing and assembly of the optical system described in flow 1. As the number of points in the adjustment grid increases, the distortion correction accuracy increases. However, adjustment becomes troublesome with a very large number of points.

In this research, we propose a simulator that can perform optical design and adjustment simultaneously, so as to reduce the cost of adjustment while maintaining sufficiently high distortion correction accuracy. In particular, we aim to fine-adjust all parameters related to placement by confirming the distortion correction result. This involves estimating the actual parameters, including manufacturing error, from the state of the distortion correction that is actually displayed. If the all parameters are precisely estimated, it is easy to increase the number of points in the distortion correction grid.

4   Simulator Design

The projection of omnidirectional images can take several formats: the dome master format can cover the projection range of the fisheye lens; the equirectangular projection format extends a 360-degree image to a rectangle, as with a world map; and the cube-map format covers all user directions with a cube, as shown in Fig. 3. We opted to use the cube-map, which can also be applied to the resulting display during the adjustment process. With this approach, it is only necessary to render images in all six directions and map their textures onto the spherical shape.
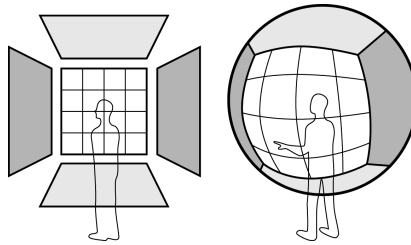

Fig. 3. Omnidirectional image rendering with cube-map

To realize the cube-map rendering, distortion correction is performed based on the relationship between the light ray $\overrightarrow{PI}$ of the projector and the projection point S, as shown in Fig. 4. Point S, which is located on the spherical screen, is the extension of a straight line from the ideal viewpoint V to the feature point M of the cube-map. In the case of Fig. 4, $\overrightarrow{PI}$ reaches point S after being reflected by the convex mirror C. Given the convex mirror's position, orientation, and curvature, S can be obtained from the

reflection from P at C. However, although S and P are already known, it is difficult to determine the intersection point C of $\overrightarrow{PI}$; that is, it is easy to follow the projector's ray in the forward direction, but difficult to calculate the inverse.
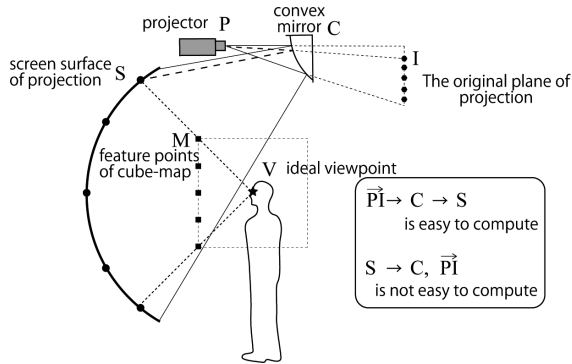

Fig. 4. Calculated points on projector-projected image plane, corresponding to feature points of cube-map

The correspondence between the original plane of projection (I) and the feature points (M) of a cube-map positioned on the ideal viewpoint is required for distortion correction. If the correspondence cannot be determined by the inverse calculation, another method must be considered. We decided to calculate all valid pixels displayed on the projection's screen surface in the forward direction, such as $\overrightarrow{PI} \rightarrow C \rightarrow S$. Then, the nearest point S is searched using the intersection point of the line VM and the screen surface, to obtain the relationship between M and I. In order to reduce the number of searches for the nearest point, we applied the hill-climbing method as a search algorithm. Since all of these ray trace calculations are independent of another, this method is easy to apply to parallel computation.

5   Simulator Implementation

Figure 5 provides an overview of the developed projection simulator. Several parameters related to the screen and projector configuration can be changed in the upper-right windows. The result of the ray tracing process is reflected immediately in the left window. The result of the distortion correction process is rendered in the lower-right window as the feature point of the cube-map. Projecting the lower-right window onto the full screen enables us to confirm the actual projection condition while changing parameters.

We applied this projection simulator to two spherical screen environments. Figure 6 shows a hemispherical screen with a diameter of 1800 mm, projecting by means of diffuse reflection with a convex mirror. Figure 5 shows the result obtained by fine adjustment of the parameters through observing the distortion correction. The result of projecting the distortion correction image onto the screen is shown in Fig. 7.
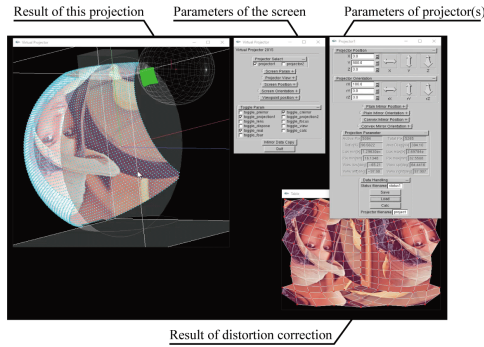
Fig. 5. Overview of projection simulator. The left window indicates the result of projecting a normal image onto the spherical screen. The lower-right window is the distortion corrected image. If the image is projected onto the screen, the viewer can see properly
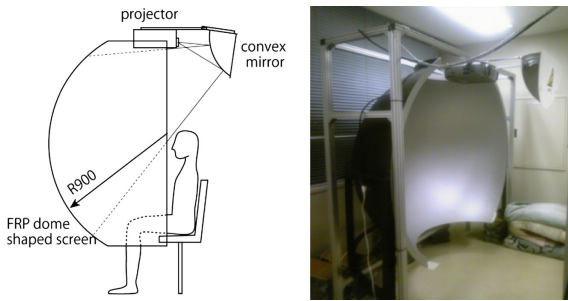

Fig. 6. Projection simulator applied to environment 1: this environment consists of a hemispherical screen of 1800 mm in diameter, projector, and convex mirror with a curvature radius of 259 mm

Fig. 7. Result of distortion correction in environment 1 (captured approximately 2 m behind the ideal viewpoint)

The distortion correction process requires a maximum of 0.21 s to complete the calculation in system equipped with an Intel CPU (Core i7-4710MQ, 16 GB main memory) when the total number of feature points is 121. If a full search algorithm is applied instead of the hill-climbing method, the process requires 7.05 s to complete the calculation. This means that the projection simulator using the hill-climbing method can follow the results in almost real time, and the user can verify the pixel distribution in the design or adjust the alignment at any time.

The second environment, shown in Fig. 8, consists of an acrylic dome screen with a diameter of 1000 mm and an ultra-short focus projector behind the screen. Figure 9 shows the distortion correction result of the cube-map on the actual screen. It can be confirmed that the left and right faces of the cube-map are displayed without distortion. The total time required for the distortion correction process was 6.8 s using a full search, 2.78 s using the optimized all-point search by OpenMP, and 0.17 s using a combination of the hill-climbing method and OpenMP.
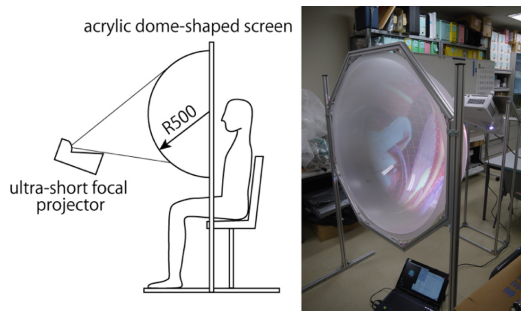

Fig. 8. Projection simulator applied to environment 2: this environment consists of a spherical screen of 1000 mm in diameter and an ultra-short focal projector behind the screen.
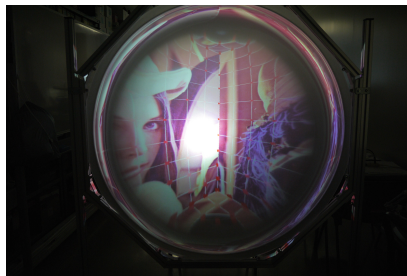

Fig. 9. Result of distortion correction in environment 2 (captured approximately 1 m behind the ideal viewpoint)

6   Conclusion

We have developed a projection simulator that can simultaneously perform optical design and adjustment of a spherical display. This simulator enables us to estimate substantive parameters related to the optical system by observing the projection state. The obtained parameters aid in increasing the number of feature points of the cube-map, for high distortion correction accuracy. The obtained distortion correction result is converted to the OBJ format, so that visual contents can be easily created with a game engine such as Unity.

The current simulator corresponds only to parametric shapes; therefore, in future work we will aim for our simulator to be able to support non-parametric shapes captured by devices such as a depth camera. In addition, further improvements in the computation time are expected when GPGPU is applied to the simulator.

References

1. IMERSA (2017). http://www.imersa.org/
2. International Planetarium Society (2017). http://www.ips-planetarium.org/
3. Raskar, R., van Baar, J., Willwacher, T., Rao, S.: Quadric transfer for immersive curved screen displays. In: EUROGRAPHICS, vol. 23, no. 3 (2004)
4. Bourke, P.: Dome: immersive gaming with the unity game engine. In: Proceedings of CGAT 2009, pp. 265–272 (2009)