An Extended Modeling Method of Optical Illusion Objects in General Rendering Environments

Takashi Chiba [†], Tomoaki Moriya [‡], Tokiichiro Takahashi [‡ *]

[†] Graduate School of Science & Technology for Future Life, Tokyo Denki University, Tokyo, Japan

[‡] School of Science & Technology for Future Life, Tokyo Denki University, Tokyo, Japan

[*] ASTRODESIGN, Inc., Tokyo, Japan

{t-chiba, moriya, toki}@vcl.jp

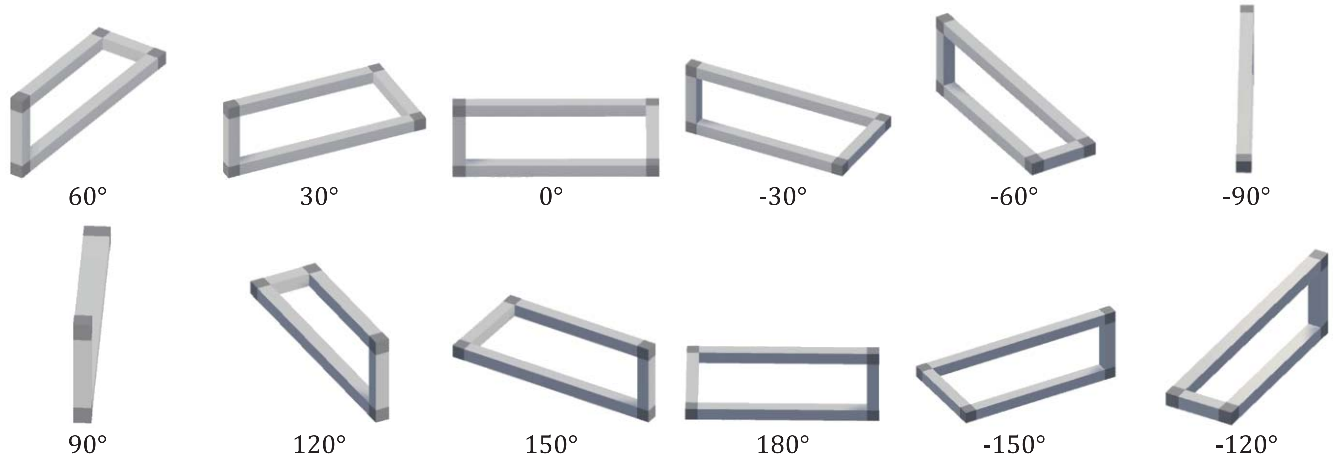


Fig.1 Our rendering result for each angle of viewpoint θ.

Abstract—**An impossible figure is a figure which looks realizable at first glance, but is not actually realizable as perceived by the human eyes. Conventional methods provide their own dedicated rendering systems to depict impossible figures. We propose a new modeling method of optical illusion objects for depicting toroidal impossible figures in general rendering environments.**

Keywords—Computer graphics; Toroidal impossible figures; Optical illusion objects; View-dependent modeling; Unity

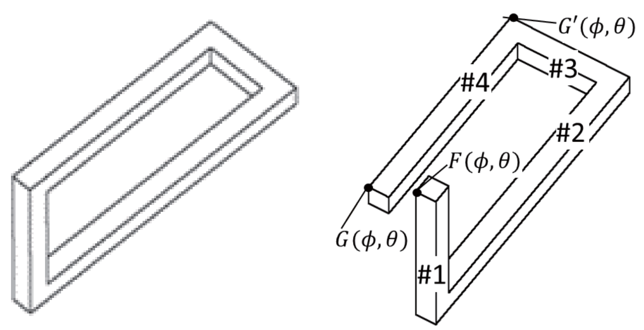
I. Introduction

An impossible figure is a figure which looks realizable at first glance, but is not actually realizable as perceived by the human eyes. An example of an impossible figure is an optical illusion object. An optical illusion object is a three-dimensional object made to be an impossible figure when viewed from a specific viewpoint. The viewpoint that realizes an optical illusion object is called as an illusion viewpoint. An optical illusion object is not perceived as an impossible figure when a viewpoint moves away from an illusion viewpoint.

By modeling 3D shapes of an optical illusion object in accordance with the motion of the illusion viewpoint, it is possible to animate the impossible figures. There have been proposed several methods which maintain even impossibilities by moving the illusion viewpoint over a wide range [1-3].

However, these methods can depict impossible figures only in dedicated rendering environments. There has been no mention of versatility for general rendering environments such as availability in game engines. Thus, a new method is needed.

In this paper, we deal with a square toroidal impossible figure with twist (Fig.2(a)) which is a representative impossible figure. We extend Nakatsu's method [3] that was limited to line drawing in previous study dealing with this figure, while keeping the advantages of the her method. Our proposed method enables Nakatsu's method under the more generic rendering environments.



(a) Toroidal impossible figure. **(b)** Optical illusion object model.

Fig.2 Nakatsu's target output and model for it.

II. Previous Work

In order to represent the square toroidal impossible figure with twist (Fig.2(a)), Nakatsu et al. [3] proposed a deformation method of the optical illusion object model according to the viewpoint and keeping the property as an impossible figure in a very wide range.

In this method, as shown in Fig.2(b), a shape model that becomes an optical illusion object is divided into a set of prisms, and by adjusting the length of prisms according to the viewpoint. As a result, the shape model looks to be connected as shown in Fig.2(a). Note that orthographic projection is assumed, and render impossible figures by line drawing.

Here, the viewpoint coordinates are expressed as (ϕ, θ) in the polar coordinate system using the latitude ϕ and the longitude θ. The viewpoint is always facing the optical illusion object. Since an orthographic projection is employed, the origin of the coordinate system and the distance to the viewpoint are not considered here.

Note that we change the coordinate system of the proposed method which is slightly different from the conventional method [3] for convenience of explanation.

In this method, each prism is stretched and moved as shown in Equation (1). So that, the point F (which is the end point of the prism #1 in Fig.2(b)) and the point G (which is the end point of the prism #4 in Fig.2(b)) are drawn at the same position on the orthographically projected image.

$$G(\phi,\theta) = \begin{pmatrix} r\sin\theta \\ -r\tan\phi \\ -r\cos\theta \end{pmatrix} + F(\phi,0) \qquad (1)$$

In this case, r is the initial length of prism #2.

Fig.3 illustrates this operation from the upper direction of prism #1.

In addition, as shown in Fig.4, the point G' is stretched by using the expansion control coefficient k. This gives an effect of natural impressions to the observer.

$$G'_X(\phi,\theta) = \frac{r}{k}\sin\theta + G'_X(\phi,0) \qquad (2)$$

In this case, $G'_X(\phi,\theta)$ represents the X-coordinate of G. Regarding the Y and Z coordinates, it follows the state of the

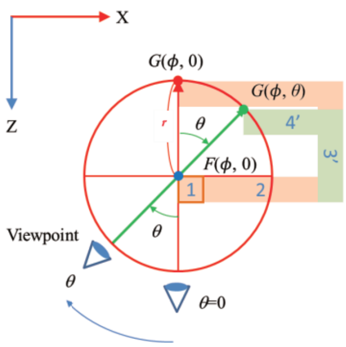


Fig.3 Move of point G.

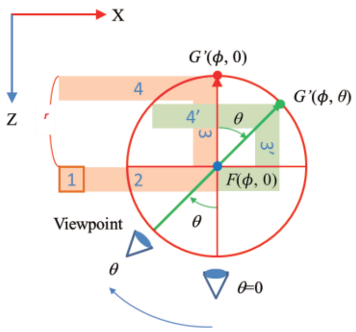


Fig.4 Move of point G'.

prism #4 after the point G has moved according to the Equation (1).

By the above operations, it seems that the point F and the point G are always connected from the viewpoint of the observer. In this paper, we call the part where the point F and point G are overlapped on the projected image and looks connected as the "pseudo-connection".

However, Nakatsu's method [3] rendered line-drawing images by their own application. When this method is applied to a general rendering environment and surface rendering, shading processing, etc. are performed, a plurality of problems which are not supposed in [3] occurs. For example, reversal of shielding relation at the corner part of the figure.

In [3], the authors do not mention about the shielding relationship on the drawing in the pseudo-connection part. The result image is obtained by their application which does not draw specific ridge lines of each model in the application that draws an impossible figure.

III. Proposed Method

In this paper, we extend Nakatsu's method to render impossible figures with shading and texture mapping by using Unity [4] as a general rendering environment.

A. Connection order change of the prisms to avoid the shape failure at shading

Fig.9(a-1) and (a-2) show shading results by Nakatsu's method. Fig.9(a-1) is the shading result where the prisms are rotated by $\theta = 45°$, in the positive direction around the prism #1 as a rotation axis. The result looks almost impossible figure. However, as shown in Fig.9(a-2), the shading result where the prisms are rotated by $\theta = -45°$, in the negative direction around the prism #1 as a rotation axis. The shielding relationship of the pseudo-connection becomes strange, and it cannot be said that it is established as an impossible figure.

This problem occurs in the range where $\phi\theta < 0$, $\theta \in [180°, -180°)$, $\phi \in [-90°, 90°]$. We call it "negative" range. On the projected image, the prism #4 should be connected to the upper end of the prism #1 and it must appear to be continuous at 90° between them. However, in the above range, the surface of the prism #1 appears on the front of the prism #4. This occurs because the prism #1 is always located closer

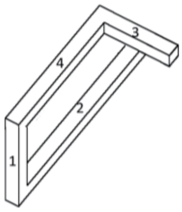


Fig.5 Our model in “negative” range.

to the viewpoint than the prism #4 by moving point G. However, on the projected image, the prism #4 is deformed so as to be extended from the prism #1 toward the near side.

Therefore, the order of the prisms is changed so that the pseudo-connection part always locates to the near as seen from the viewpoint. For example, when φ > 0 and θ < 0, after deforming prisms by the previous method, prism #4 and #3 are moved to the upper side of prism #1 as shown in Fig.5.

By this operation, in the "negative" range, the pseudo-connection part connects the prism #2 and #3. Since the two prisms are expanded from the pseudo-connection part on the viewpoint side, the above problem can be solved. The result is shown in Fig.6.

We call the procedure described in this sub-section A as Method A.

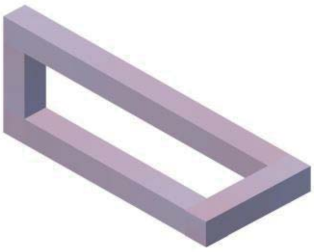
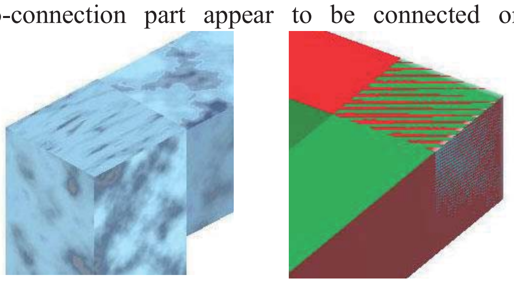


Fig.6 Result by applying Method A.

B. *Introduction of corner objects to reduce unnaturalness at shading*

By Method A, the issue of the shielding relation near the pseudo-connection part was solved. However, when drawing the surface of the optical illusion object model, new issues arise.

One of the issues is the texture of the surface which is not properly connected. As an example, consider a situation as shown in Fig.9(a-2). The prisms #4 and #1 constituting the pseudo-connection part appear to be connected on the projection image, but they are separated in the 3D space. Therefore, the prism #1 is located closer to the viewpoint than the prism #4. However, their textures are not connected as shown in Fig.7(a).



(a) Texture seam. **(b)** Z-Fighting.

Fig.7 Expansion of illusion corner.

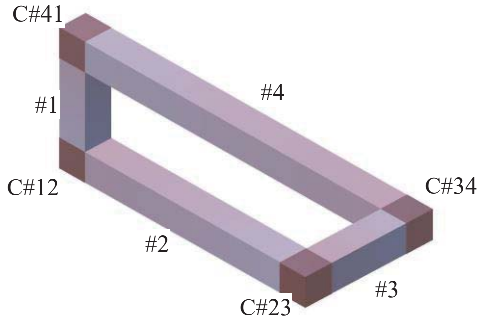


Fig.8 Corner objects.

Even if texture is monochromatic, in some environments, for example in Unity, although there are only light sources of ambient and directional, colors may differ due to difference in internal drawing calculation.

Furthermore, in the connecting part (connected on both the projection image and the 3D space), since the prisms have polygons that similar values in the Z-buffer, a phenomenon called "Z-fighting" occurs. Fig.7(b) is color arrangement to make this issue easy to understand.

In order to avoid these phenomena, the cubic parts where prisms overlap are made independent as corner objects (C#12, C#23, C#34, and C#41 in Fig.8). At that time, the length of the prism is shortened to the corner objects and the part not covered. The corner objects of the pseudo-connection part is always located closer to the viewpoint.

When assigning a color or texture different from the prisms to this corner objects, the corner objects strongly give the observer the recognition that "corner color and texture are different for this torus figure." As a result, there is an effect which reduces the unnaturalness of the pseudo-connection part, too.

By the above procedure, it became possible to naturally draw a square toroidal impossible figure with twist in the orthographic projection rendering environment which performs shading processing. Comparison and enlarged view between the previous method and proposed method when $\theta = 45°, -45°$ are shown in Fig.9.

IV. Results and Discussion

By using our proposed method, it becomes possible to render impossible figures appropriately while drawing the surface in a more general environment, and keep it following the movement of the viewpoint.

Fig.1 shows the rendering results at each angle by our proposed method. Our method maintains even impossibilities by moving the illusion viewpoint over a wide range. Several issues indicated in section III (Shape failure, unnatural pseudo-connection part) are resolved.

In addition, because it becomes possible to draw in a general environment, it is possible to easily operate materials of optical illusion objects. However, regarding the change of the light source, there is an issue that the reflection direction suddenly changes when moving viewpoint (such as crossing into "negative" range) if there is a light source very close to the optical illusion object. Further investigation is necessary for the allowable range of the light source.

V. Conclusion

Nakatsu's method one of the most effective and expressive representation methods for a square toroidal impossible figure with twist. In this paper, we extended Nakatsu's method, and propose a generic rendering method for the optical illusion object model to make it as an impossible figure under the more general rendering environments. The proposed method makes it possible to draw impossible figures in various 3D environments such as various game engines. This makes it possible to use as an asset inside the existing or new 3D scene.

Acknowledgement

We would like to thank all the members of Visual Computing Lab., Tokyo Denki University for their encouragement and fruitful discussion. This work was supported by JSPS KAKENHI Grant Numbers JP-17K19965 and JP-17H004436.

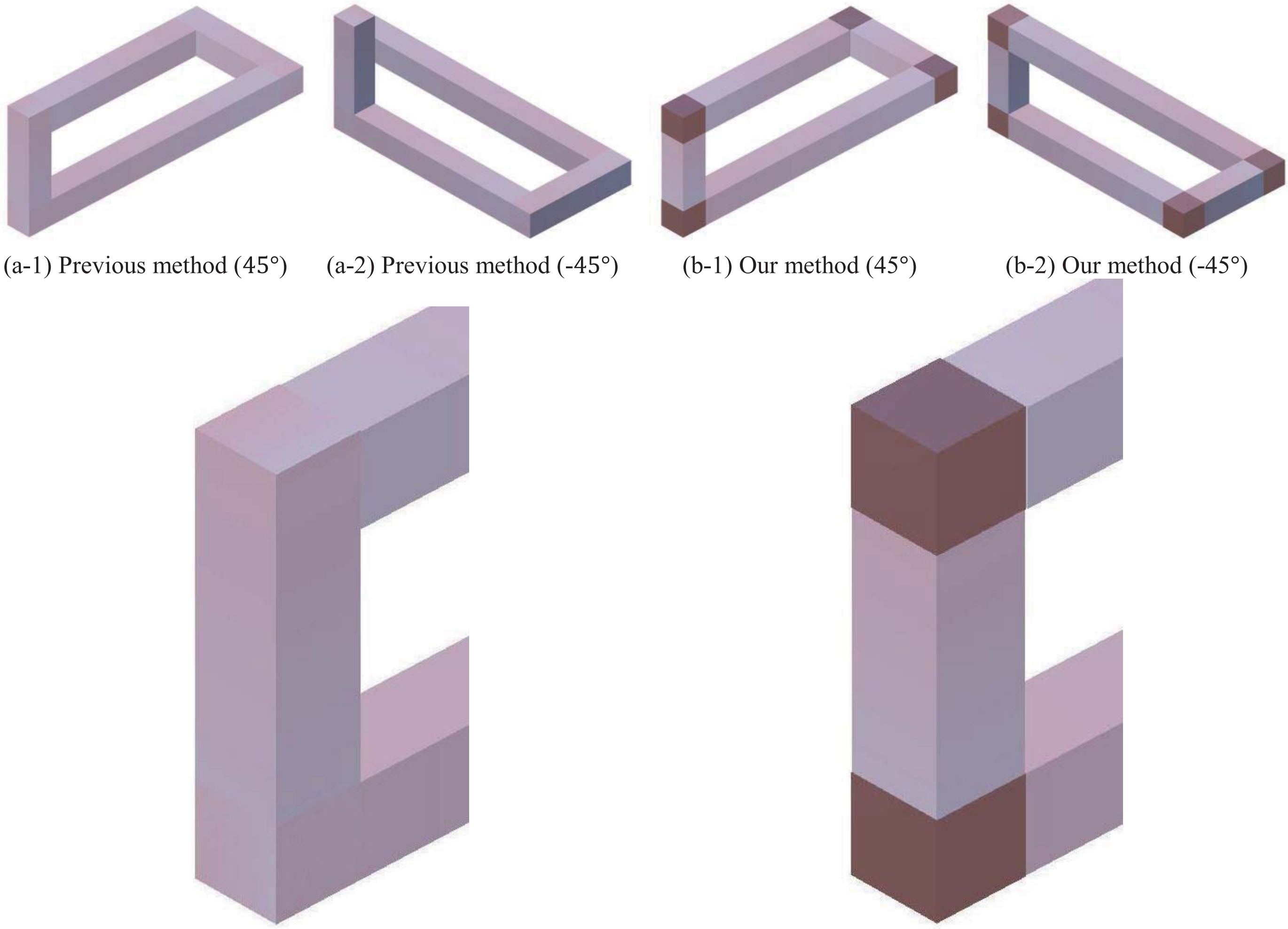
References

[1] J. Fujiki, S. Owada, "theRelativity," ACM SIGGRAPH ASIA 2008 Art Gallery, Dec. 2008.

[2] T. P. Wu, C. W. Fu, S. K. Yeung, J. Jia, C. K. Tang, "Modeling and Rendering of Impossible Figures," ACM Trans. On Graphics, vol. 29, no. 2, pp. 13:1-13:15, Aug. 2010.

[3] K. Nakatsu, T. Takahashi, T. Moriya, "A Stereoscopic Representation of Impossible Rectangle Twisted Torus Figure," ACM SIGGRAPH 2012 Posters, Aug. 2012.

[4] Unity, https://unity3d.com/



(a-1) Previous method (45°) (a-2) Previous method (-45°) (b-1) Our method (45°) (b-2) Our method (-45°)

(a-3) Previous method (Expansion) (b-3) Our method (Expansion)

Fig.9 Comparisons of our method and the previous method.