DOCTOR OF PHILOSOPHY

Modeling and Rendering Three-Dimensional Impossible Objects

Taylor, Ben

Award date:
2020

Awarding institution:
Bangor University

Link to publication

PRIFYSGOL
BANGOR
UNIVERSITY

School of Computer Science and Electronic Engineering
College of Environmental Sciences and Engineering

Modeling and Rendering Three-Dimensional Impossible Objects

Benjamin Adam Taylor

Submitted in partial satisfaction of the requirements for the
Degree of Doctor of Philosophy
in Computer Science

Supervisor Dr. Ik Soo Lim

May 2020

Acknowledgements

I would like to thank my supervisor Dr. Ik Soo Lim for his help and advice throughout my Ph.D., without his guidance this work would not have been possible. I would also like to thank my family and friends who have encouraged and supported me along every step of this journey, and for providing a much-needed distraction through the more difficult moments.

Abstract

Impossible Figures, such as the Penrose triangle or those created by M.C. Escher, are a form of optical illusion consisting of locally possible sections joined together to form a globally inconsistent structure. Through exploiting hidden deformations within the object's structure it is possible to model three-dimensional representations of these impossible figures. The hidden deformation methods used, however, result in corresponding deformations or anomalies in the object's cast shadow. Due to these deformed shape of the resulting shadows they are often excluded from the final rendering. By excluding cast shadows entirely the important contextual and positional information they contain about the scene is also lost.

This work presents a novel method of modeling three-dimensional versions of these impossible figures through the use of transparency. By manipulating the surface transparency we demonstrate the ability to simulate a range of impossible figures under new viewpoints. We also produce a screen space occlusion method for rendering impossible cast shadows automatically. By manipulating information stored within the depth buffer our algorithm produces impossible cast shadows for both possible and impossible objects. Providing a solution to casting visually appropriate shadows for impossible objects, along with other rendering features such as ambient occlusion. By operating in screen space our algorithm works without disrupting the existing rendering pipeline, as such we have implemented it within the Unity engine. To examine the effectiveness of our copycat shadow algorithm we conduct of visual perception experiments. Exploring the effect cast shadows have on the viewers perception of impossible objects.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Impossible figures are a form of optical illusion consisting of a collection of locally consistent possible sections connected in such a way to form a globally inconsistent structure. A common example of this is the Penrose triangle [64], each corner of the triangle being locally consistent, however, when viewed as a whole produces depth inconsistencies in the global structure and confuses our perception of the object. Examples of these impossible figures can be found in pictorial art, most notably in the work of M. C. Escher [19]. These figures are commonly depicted as two-dimensional images, either in the form of line drawings or using simple flat shading.

In the field of computer graphics work has been done to allow the creation of three-dimensional models of these impossible figures [15, 84]. Creating three-dimensional representations of these figures enables them to be viewed in a completely different way. When dealing with two-dimensional figures they are often depicted from a single ’illusion viewpoint’. Working with the three-dimensional objects opens up the possibility to view these impossible figures from new view directions. Previous visual perception research has examined our perception of these figures in two dimensions [23, 24] to explore how our visual system interprets and constructs mental images of them. However the ability to model more complex three-dimensional versions of these figures opens up opportunities to explore what effect rendering these objects in more complex scenes will have. One limitations that is evident when rendering these impossible objects however is the ability to render them along with convincing cast shadows. As impossible figures by definitions cant exist in the real work there is no answer as to the type of shadow they should cast.

Many existing methods of modeling these impossible objects use some form of deformation to simulate the impossibility. Manipulating the geometric structure of the object, this however results in corresponding anomalies in the object's cast shadow and shading. In current examples where we see renderings of these three-dimensional impossible models the cast shadows are completely excluded. However excluding all forms of cast shadows can not only create perceptual problems when viewing the image, as cast shadows convey important spatial and positional information, but also limits the use of these objects in rendered three-dimensional scenes. Without a cast shadow, an object stands out and does not look to be part of the scene.

We aim to introduce methods to render these three-dimensional impossible objects with convincing shading and cast shadows, opening up their use in full three-dimensional scenes, along with investigating how the introduction of these cast shadows impacts our perception of the impossible objects.

1.1 Modeling and Rendering Impossible Objects

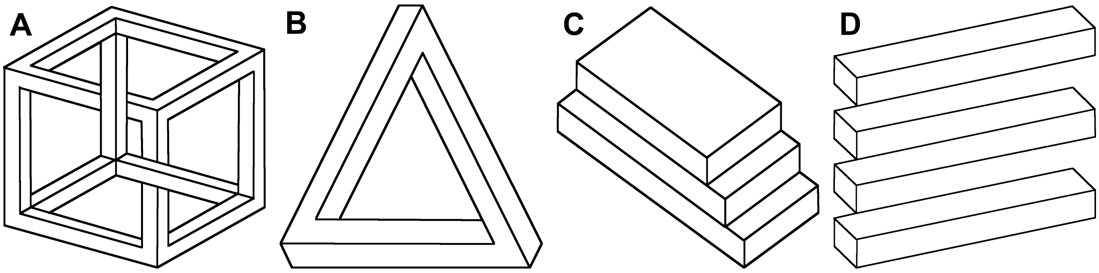


Figure 1.1: Examples of four common types of impossibility. (*A*) The Impossible Cube. (*B*) The Penrose Triangle. (*C*) The Impossible Staircase. (*D*) The Impossible Shelves.

An impossible figure is a type of optical illusion, consisting of a two-dimensional image that our visual system interprets as a projection of a three-dimensional object. The illusion is created due to inconsistencies in the structure of the perceived object that does not appear to be realizable in the real world [45]. Impossible figures can be divided into four main categories [17], where each figure is created through a different form of optical illusion:

- **Depth Interposition**. These illusions are caused by the structural inconsistency of the object due to irregularities in the depth ordering.

The Impossible Cube, Figure 1.1, where parts of the cube appear to be both in front and behind at the same time.

- **Depth Contradiction**. These illusions are caused by differences in the local three-dimensional information resulting in inconsistencies in the global object structure. The Penrose Triangle, Figure 1.1, where the three sides of the triangle join to form an impossible cycle.

- **Disappearing Normals**. These illusions are caused by the appearance of a twist on the surface of an object, at one side the surface may appear vertical whilst at the other, it may appear horizontal. Causing the object to have an inconsistent normal direction across the surface. The Impossible Staircase, Figure 1.1, where one object face gradually transitions into another.

- **Disappearing Space**. These illusions are caused by the silhouette of the object not being fully closed. The Impossible Shelves, Figure 1.1, where sections of the object seem to disappear completely.

Most impossible figures consist of one or more of these four types of illusion.

1.1.1 Modeling Impossible Objects

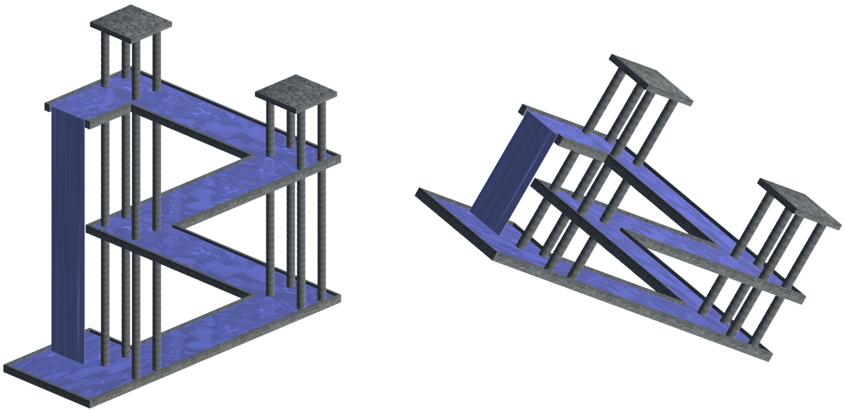


Figure 1.2: A three dimensional model of M. C. Eschers "*Waterfall*" [18] modeled using our transparency based modeling method. Our modeling method allows for easy shading and texturing along with generating novel viewpoints of the object.

Creating a three dimensional model of these impossible figures often begins with creating a three-dimensional possible representation and manipulating it in some way to create the illusion of impossibility. Taking advantage of the limitations of our visual system to correctly interpret depth in two-dimensional images. Although these seemingly impossible models (SIMs) create the illusion of impossibility they are completely possible three-dimensional models. One of the most popular methods to achieve this is through hidden deformations [15, 84]. These deformations to the geometry of the object are applied along the line of sight such that they are not visible in the final image. These manipulations to the object's geometry can cause anomalies when rendering these three-dimensional objects using traditional rendering and lighting equations, such as inconsistent shading and texturing, that must be corrected.

We provide an impossible object modeling method based on manipulating the surface transparency of an object. Our transparency based modeling method is able to simulate each of the four main types of impossibility shown in Figure 1.1 under a range of novel viewpoints. As our method does not manipulate the geometry of the object, shading and textures can be applied without the need for additional corrections. Figure 1.2 demonstrates how our transparency based modeling method can be used to create a three-dimensional representation of M. C. Escher's "*Waterfall*" [18].

1.1.2 Cast Shadows of Impossible Objects

Although work has been done to allow the creation of tangible three-dimensional models of these impossible figures, they are often rendered under very limited lighting conditions. Only containing basic shading and lacking any form of cast shadow. The techniques used to create the illusion of impossibility within these SIMs causes anomalies in their cast shadows, as such cast shadows are often completely excluded from the final rendered image. The lack of cast shadows within a scene can result in an unnatural look, reducing the realism of the scene. Without cast shadows it can be difficult to accurately understand the composition of the scene and the location and positioning of the objects within. Not being able to cast appropriate looking

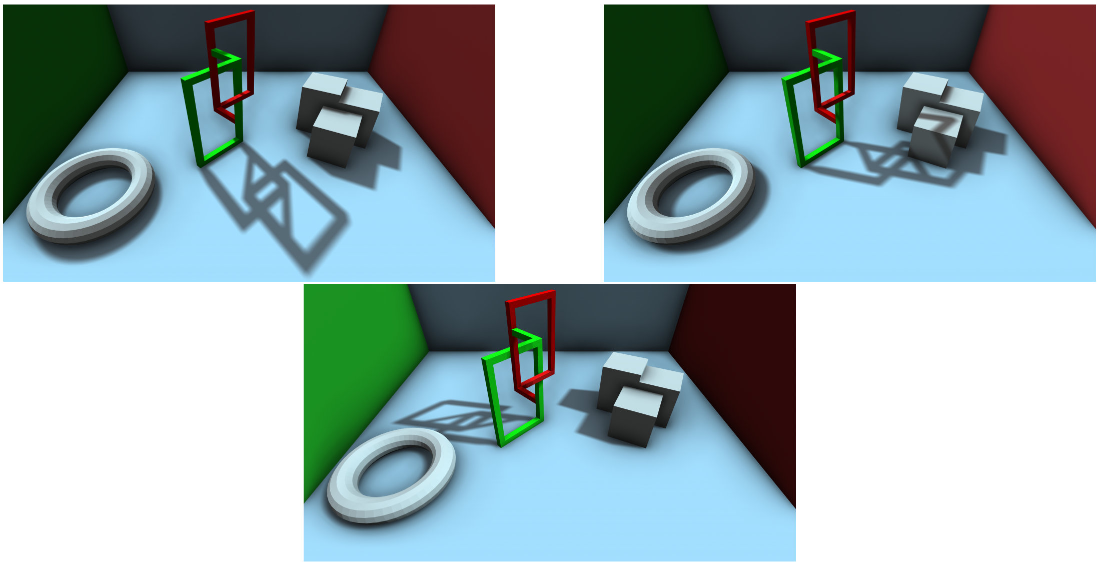


Figure 1.3: A scene containing a seemingly impossible model (SIM) casting a copycat shadow rendered using our screen space copycat shadow method. We are able to achieve a convincing soft shadow and ambient occlusion for the SIM.

shadows limits the use of these impossible models in three-dimensional scenes.

We suggest that a solution to casting appropriate looking shadows for these SIMs may come in the form of copycat cast shadows. The purpose of a copycat shadow is not to represent the true three-dimensional structure of the casting object, but rather to produce a shadow whose shape aims to replicate the visible profile of the casting object. Evidence of these copycat cast shadows has been found in paintings and artworks by many different artists [8]. Although copycat shadows are technically impossible they often go unnoticed by the observer.

We present a screen space method of generating copycat cast shadows automatically under a range of different light and view directions. Figure 1.3 shows a SIM rendered alongside other objects within a scene using our copycat shadow method. By projecting the stored depth information of the model onto a planar surface we are able to generate copycat shadows for any object, both possible and impossible. As our method operates on screen-space information it does not disrupt the existing rendering pipeline and can work alongside already established features such as ambient occlusion.

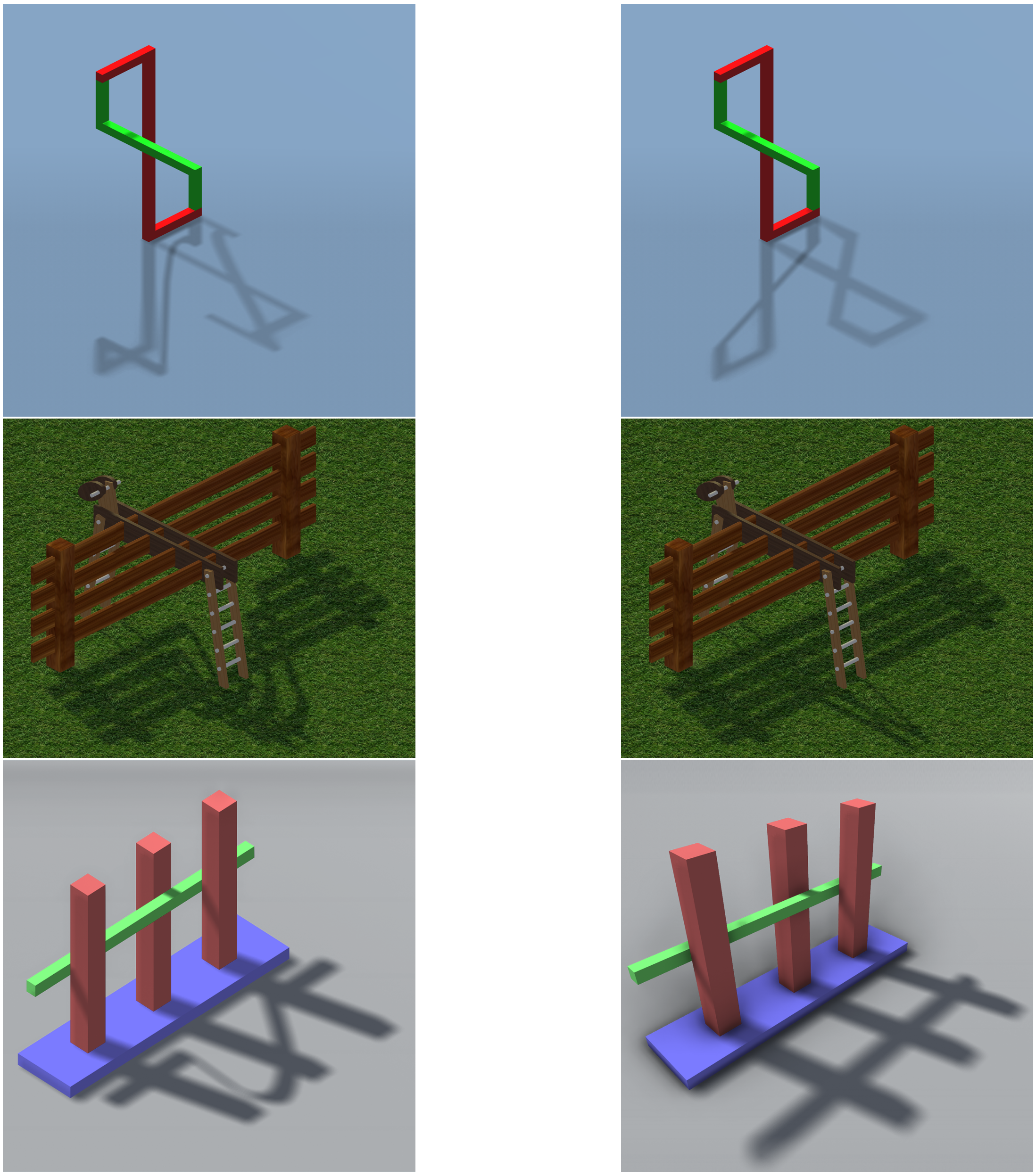


Figure 1.4: This figure demonstrates some of the we have been able to achieve using our copycat shadow algorithm. Where previous renderings of impossible objects like these are seen with no cast shadows, using our algorithm we have been able to render them alongside full visually convincing soft shadows. Also using other features such as multiple light sources and full ambient occlusion. (*left*) Shows the objects traditional cast shadows. (*right*) Shows copycat cast shadows generated using our algorithm.

Figure 1.4 demonstrates some of the more advanced result we have been able to achieve using our copycat shadow algorithm.

1.1.3 Perception of Impossible Objects

Impossible figures have commonly been used in visual perception research. The stimulus often consists of basic grayscale line drawings of these impossible figures as those were the resources easily available. With our ability to model and render SIMs in three dimensions we wanted to investigate what effect this had on our perception of the objects, in particular, what effect the new inclusion of cast shadows for these impossible objects would have. We performed a series of visual perception experiments to test our perception of these impossible objects and their normal cast shadows along with evaluating the effectiveness of our copycat shadow algorithm, exploring how our algorithm preforms against traditional cast shadows.

1.1.4 Aims and Objectives

This research aims to explore methods of modeling and rendering three-dimensional representations of impossible figures. It will investigate alternate methods of modeling impossible objects in three dimensions. Along with rendering these objects alongside cast shadows, opening up the possibility to use these impossible objects in more complex three-dimensional scenes. This work will also investigate what effect cast shadows have on a viewer's perception of these impossible models.

By the end of this research we hope to answer these two questions:

1. Do impossible 'copycat' cast shadows provide a viable solution to rendering visually convincing cast shadows for three-dimensional impossible objects?

2. What effect does the inclusion of cast shadows have on a viewer's perception of impossible objects?

Our objectives throughout this research are as follows:

- Explore the existing uses of impossible objects within computer graphics and pictorial art. Investigate the existing methods that have been implemented to model and render these impossible objects.

- Investigate the role of cast shadows in object perception.

- Design a new impossible object modeling algorithm based on transparency.

- Design an algorithm for automatically generating copycat cast shadows. Where previously these shadows are created using manual post-processing.

- Implement our algorithms using existing tools so that they are easy to use without the need for any dedicated rendering environment.

- Analyse the effect cast shadows have on a viewer's perception of impossible objects through a series of visual perception experiments.

1.2 Contributions

The main contributions of this research consist of:

1. We provide a new transparency based method for creating seemingly impossible models. Our approach manipulates the surface transparency of object sections to create the illusion of impossibility. One of the key contributions is the ability to simulate a wide range of different impossibility illusions under novel viewpoints. Along with working under both orthographic and perspective projection. As our method does not manipulate the actual geometric structure of the object, shading and texturing can be applied without anomalies. This has been implemented using existing tools, avoiding the need for a dedicated rendering environment.

2. We provide a fully automatic method of generating copycat cast shadows. By projecting object depths in screen space our algorithm can automatically generate copycat shadows for any object, both possible and impossible, under a range of view and light directions. As our algorithm operates on screen-space data it does not disrupt the existing rendering pipeline and can be used alongside features such as perspective projection and ambient occlusion. By using copycat shadows we demonstrate how it is possible to render visually appropriate cast shadows for three-dimensional impossible objects. This work shows the potential for impossible objects to be rendered convincingly in three-dimensional scenes. This has been implemented using existing tools, avoiding the need for a dedicated rendering environment.

3. We provide evidence from visual perception experiments that copycat shadows generated using our screen space algorithm appear more convincing and visually appropriate than the traditional cast shadows for impossible objects. Along with evidence that the inclusion of cast shadows alongside impossible objects aids our visual system in understanding the structure of these objects.

1.3 Thesis Outline

Chapter 1. Introduction. Introduction, background, research aims, and contributions.

Chapter 2. Related Work. We review the previous related work in both computer graphics, shadow rendering and impossible object modeling, and visual perception, perception of impossible objects, and cast shadows.

Chapter 3. Transparency Based Impossibility Modeling. We present a transparency based method for modeling three-dimensional impossible objects.

Chapter 4. Screen Space Copycat Shadows. We present a screen space method for automatically generating copycat cast shadows. Demonstrating how these shadows can be used to cast visually appropriate shadows for three-dimensional impossible models.

Chapter 5. Visual Perception of Impossible Models. We present results from a series of visual perception experiments carried out to test how the inclusion of cast shadows affects our visual perception of impossible objects.

Chapter 6. Conclusions and Future Work. We reflect on the goals of this thesis as well as outlining possible future research directions.

Chapter 2

Related Work

There has been much research on impossible figures and cast shadows individually, however not directly in relation to one another. Visual perception of impossible figures and our understanding of cast shadows have been studied by cognitive scientists and psychologists separately but not their direct impact on one other. Methods of modeling three-dimensional versions of these impossible figures have been developed in computer graphics but are often rendered using basic lighting and no cast shadows. In this chapter, we discuss the previous work on impossible objects and cast shadows, including their history and use by artists such as M. C. Escher, modeling and rendering methods introduced in computer graphics, along with their use in psychology to better understand human visual perception.

2.1 Impossible Objects

When viewing a two-dimensional image our visual system often interprets the image as a projection of a three-dimensional figure. For example, when viewed, Figure 2.1a would most commonly be interpreted as a projection of a three-dimensional box. Less likely would be the interpretation of a more planar figure, such as two squares connected by diagonal lines or a square surrounded by irregular two-dimensional geometric shapes.

This feature of our visual systems insistence on viewing two-dimensional objects as projections of three-dimensional shapes can lead to interesting problems, or by another name optical illusions. For example, Figure 2.1b appears to be a three-dimensional object. However, the object itself seems

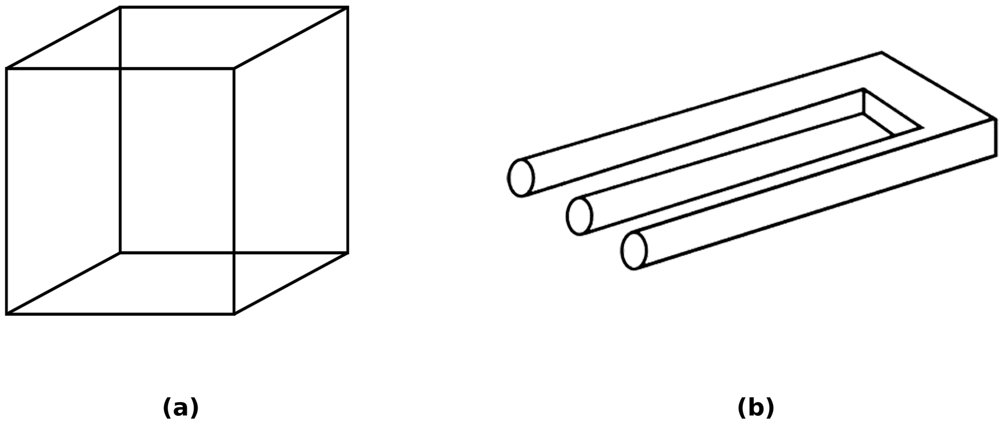
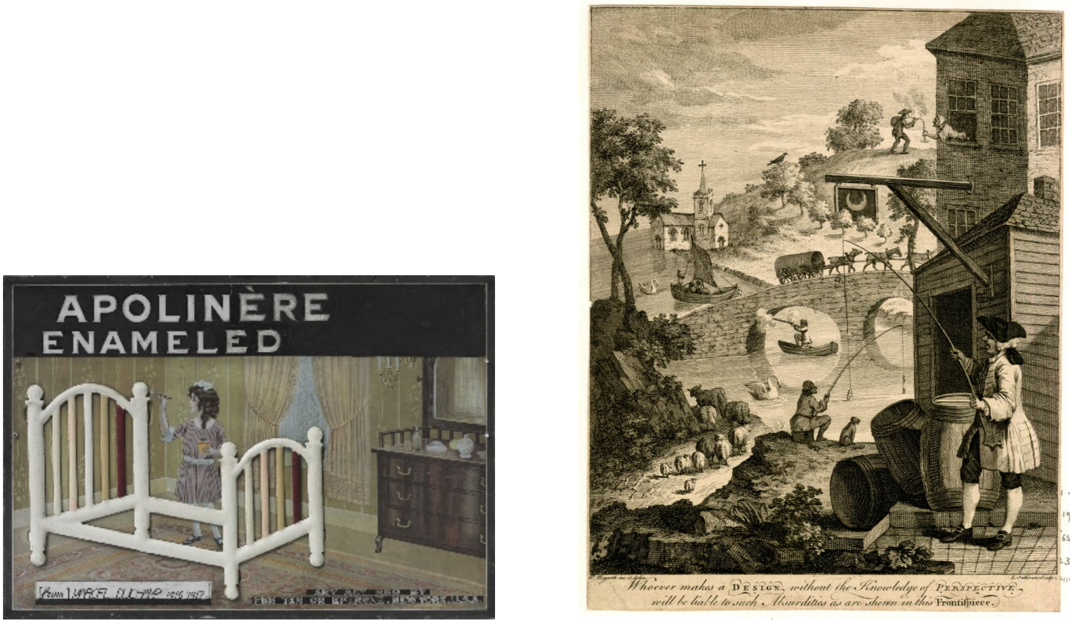


Figure 2.1: 2D line drawings give the visual impression of a 3D object. (*a*) The Necker Cube, a series of 2D lines that create the illusion of a 3D cube when viewed. (*b*) The Impossible Trident, a line drawn impossible figure whose shape contradicts itself. When focusing on the left side it appears to have three distinct forks, but when focusing on the right side it only appears to have two.

to change its properties based on how it is viewed. When focusing on the left side of the object it appears to have three distinct prongs, however when focusing on the right side of the object it only appears to have two.

If our visual system were to interpret this image as a two-dimensional figure then there would be no confusion as it is an entirely possible configuration of three circles joined together by connecting lines. However as our visual system attempts to construct a three-dimensional object from the image, the irregularities and inconsistencies within the structure of the figure make it impossible.

To create the impression of three-dimensions a variety of visual cues are used, including interposition or occlusion, height, perspective, size, and shading. These impossible figures take advantage of these visual cues and at the same time break them. Giving viewers the illusion of a three-dimensional object which under closer inspection is revealed to be impossible. Interestingly although the impossibility can become apparent after viewing the figure, the initial impression of a three-dimensional object remains even after it has been contradicted. "These pictures work so well because they obey the pictorial rules in local regions but defy them globally." [81]


APOLINÈRE ENAMELED
Whoever makes a DESIGN without the Knowledge of PERSPECTIVE will be liable to such Absurdities as are shewn in this Frontispiece.

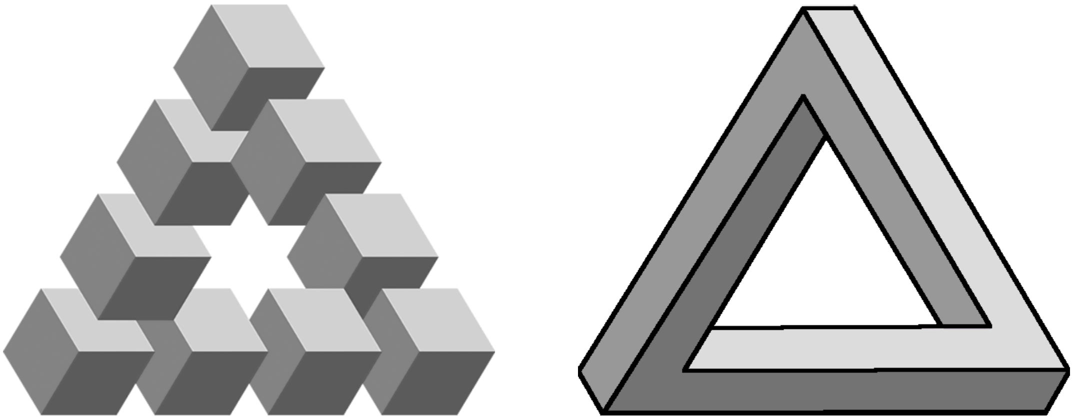
(a) Apolinère Enameled, Marcel Duchamp, 1916-17

(b) Satire on False Perspective, William Hogarth, 1754

Figure 2.2: Early examples of impossible figures used by artists. (*a*) Depicts a bed where the frame includes conflicting perspectives. (*b*) Contains numerous perspective effects, including the sign moored to two buildings, one in front of the other, with beams that show no difference in depth.

An early example of an impossible figure can be seen in Figure 2.2a, "Apolinère Enameled" by Marcel Duchamp, sometimes referred to as Duchamp’s impossible bed. The image depicts a girl painting a bed frame, the frame itself deliberately includes conflicting depth interpositions to create an impossible object. Another early example of the use of impossibility can be seen in Figure 2.2b, William Hogarths "Satire on False Perspective". This work shows a scene that contains many deliberate examples of irregular or inconsistent perspective effects. If we focus on the particular element of the sign, we can see that it is attached to two separate buildings one in the foreground and the other in the background. However, the beams of the sign itself show no difference in depth. This could be considered an early example of one of the most notable impossible figures, the impossible tribar or Penrose triangle.

The impossible tribar was first created by Swedish artist Oscar Reutersvärd in 1934 (Figure 2.3a). His version consisted of a series of cubes viewed in parallel projection, with each cube appearing to overlap the previous cube,

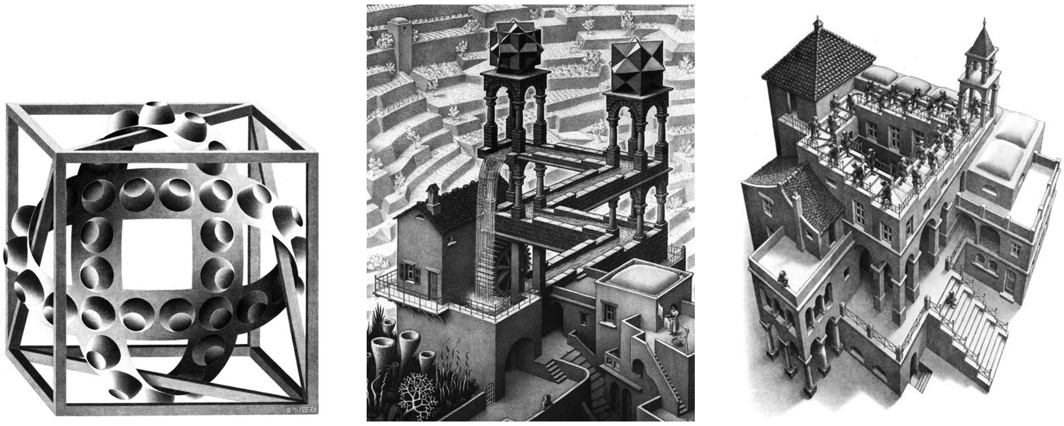


(a) Reutersvärd Triangle

(b) Penrose Triangle

Figure 2.3: Versions of the impossible triangle.

creating an infinite impossible loop. This impossible tribar was independently developed and popularised by Roger Penrose [64], who drew the figure as three rectangular bars connected at right angles to create a closed figure (Figure 2.3b).



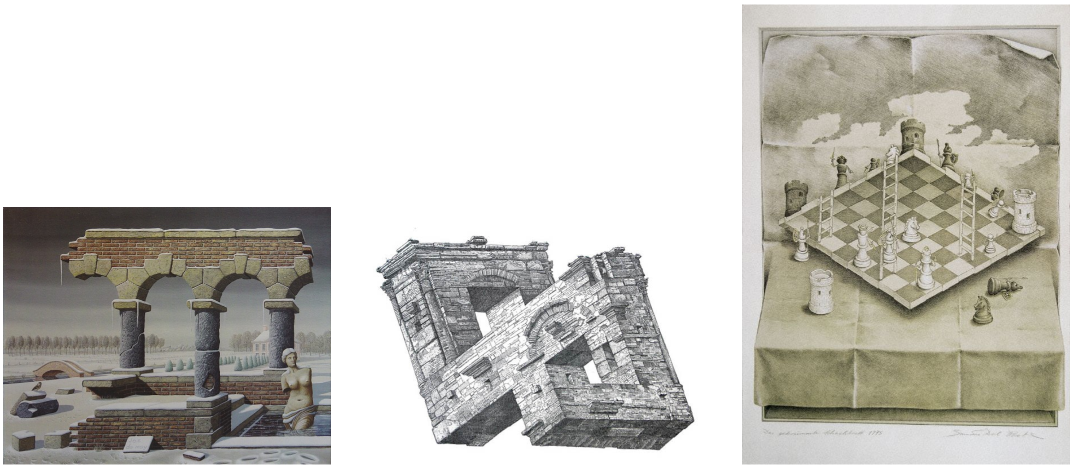
(a) Cube With Magic Ribbons, M. C. Escher, 1957

(b) Waterfall, M. C. Escher, 1961

(c) Ascending and Decending, M. C. Escher, 1960

Figure 2.4: Lithographs by M. C. Escher containing various examples of impossible figures.

Examples of the impossible triangle can be found in more mainstream artworks, in particular those of Dutch artist M. C. Escher [19] who produced many drawings featuring irregular perspective gradually working towards true impossible objects. In 1957 his first drawing containing a true impossible object was created, "Cube with Magic Ribbons" shown in Figure 2.4a, depicting two interlocking bands wrapped around the frame of a necker cube. Whilst Penrose and Reutersvärd focused mainly on designing shapes and objects that existed outside conventional geometric laws, Escher managed to incorporate these shapes into his paintings and artworks.

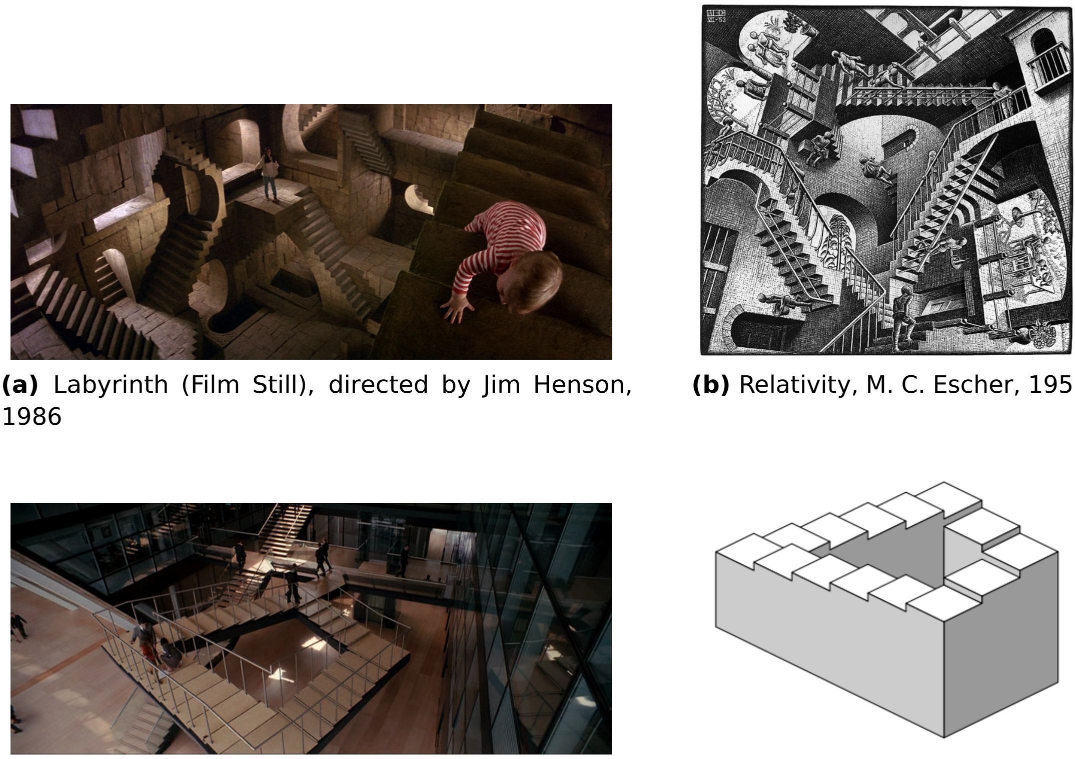


(a) Winterzicht van de onvoltooide Pergola, Jos De Mey, 1980 **(b)** Fal (Wall), István Orosz, 2006 **(c)** The Warped Chessboard, Sando Del Prete, 1975

Figure 2.5: Examples of pictorial art containing impossible figures.

Further examples of impossible objects and structures can be found throughout Escher's works including Figure 2.4b "Waterfall", whose construction depicts two instances of the impossible tribar, and Figure 2.4c "Ascending and Descending", whose inner courtyard is encircled by a never-ending staircase. Escher's work did much to draw the attention of the public to impossible objects and inspire further artists to experiment with impossible figures. Artists such as Jos de Mey, István Orosz, and Sandro del Prete all use examples of impossible objects within their work, as shown in Figure 2.5.

The use of these impossible objects has also extended outside of static two-dimensional paintings and drawings with examples being seen in both film and video game media. An early example of this can be seen in the 1986 film 'Labyrinth' shown in Figure 2.6a, where the characters can be seen exploring a castle containing an elaborate set of staircases themed after Escher's artwork "Relativity" (Figure 2.6b). In the 2010 film 'Inception', we see an example of the characters traversing a set of impossible stairs (Figure 2.6c). To achieve this effect the film-makers were tasked with replicating the impossible stairs (Figure 2.6d) in the real world. Although it may seem that these impossible figures cannot really exist in three-dimensional space they can be simulated through the clever use of forced perspective and hidden cuts



(a) Labyrinth (Film Still), directed by Jim Henson, 1986

(b) Relativity, M. C. Escher, 1953

(c) Inception (Film Still), directed by Christoper Nolan, 2010

(d) Penrose impossible stairs.

Figure 2.6: Examples of impossible objects in film.

and deformations in the structure of the object. To simulate the impossible effect of the stairs they must be viewed in such a way that disguises the disconnected nature of the staircase.

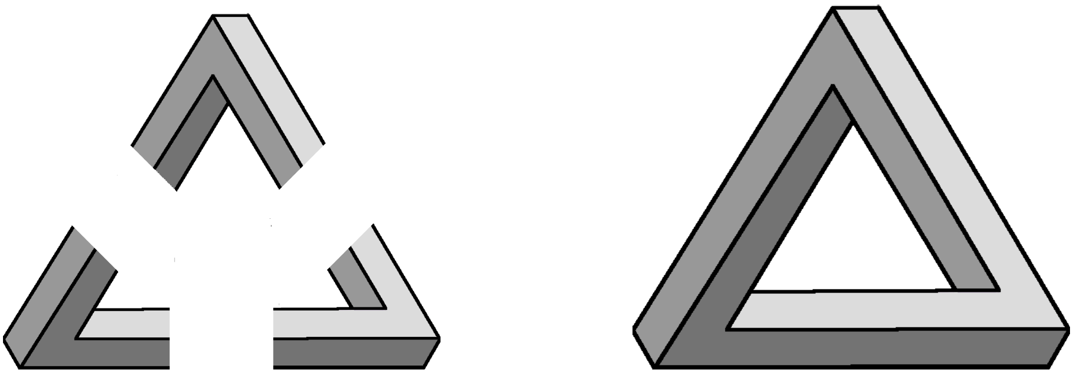
Further examples of these impossible objects existing in the real world can be found in the work of Kokichi Sugihara [74] who describes a set of impossible objects that can be modeled in three dimensions using paper templates. These three-dimensional objects still maintain their illusion of impossibility when viewed from specific viewpoints. Video games such as "Echochrome" and "Monument Valley" have implemented the use of these impossible objects in an interactive setting. Tasking players to navigate through levels designed and based on impossible objects. Players have the freedom to change the view of the object along with manipulating the object itself in order to create a pathway for their character.

2.1.1 Classification of Impossible Figures

Impossible figures can be seen in many forms, from the simple geometric style drawings of Reutersvärd to the elaborate scenes of Escher. The foundations of these objects, however, are much the same, connecting locally possible sections in such a way that makes the object globally impossible. While this principle may be the same for all impossible objects, the way in which it is executed can be very different. Take the impossible triangle (Fig 2.3) and the impossible trident (Fig 2.1b) as examples. The impossible triangle crates the illusion of impossibility through its connected nature, where the order in which the object is connected creates a contradiction in-depth where sections of the object appear to be both in front and behind one another. The impossible trident, on the other hand, creates the illusion of its impossibility through the open ambiguous nature of the structure, where parts of the object seem to disappear creating an open silhouette.

To better understand how these impossible figures work different classification methods have been devised. Sugihara [74] attempts to classify these objects using an edge labeling technique, extended from Huffmans [36] edge labeling algorithm. Each edge is labeled based on its function, an edge which forms a ridge, an edge which forms a valley and an edge which forms a silhouette. These so-called ’labelabble’ figures create the impression of a three-dimensional object when viewed. According to Sugihara for a figure to be impossible it must be labelable (each edge must be assigned a label) whilst not producing a physically correct polyhedron. Under this categorization method, only those figures with a realizable three-dimensional structure are considered true impossible figures, this excludes such figures as the impossible trident (Fig 2.1b) as the edges of the object are ambiguous making it ’un-labalable’.

Alternate methods of categorization focus on the type of impossibility contained within the object. Thro [78] identifies two distinguishable types of impossible figures, impossible solids, and spatially impossible objects. Impossible solids create their illusion though changes in the contours of the object, where lines go from delineating an inside edge to an outside edge for example. This creates an open-ended object that lacks a continuous edge, such as the impossible trident. Spatially impossible solids create their illusion though inconsistencies in the depth of the object, giving the impression that the object lies on a single plane at the same time as it lies on separate planes, such as the impossible triangle.



(a) Locally possible sections.

(b) Globally impossible structure.

Figure 2.7: Although the global structure of the Penrose triangle is impossible, it is created by joining together locally possible sections.

For both these categories of impossibility, there may be an infinite number of different impossible figures. Kulpa [45] suggests that although there may be many different impossible figures they can be constructed using a basic set of building blocks. Using the Impossible triangle as an example, take each corner of the triangle viewed individually (Fig 2.7a) there is no impossibility to be seen. However, when these corners are connected we get a globally impossible construction (Fig 2.7b). The impossible triangle falls within a specific class of figures introduced by Cowan [13] as cornered toruses, otherwise known as multi bars or impossible polygons. This subset of impossible figures are created using a number of rectangular sections (or bars) connected at their ends to form n-sided polygons. Kulpa [45] expands this classification by identifying the four basic corner configurations used in the creation of these multibar objects. Using this basic set of corner configurations it is possible to construct a near-infinite number of impossible multi bars. For tribars specifically, like the impossible triangle, Kulpa [45] identifies 24 possible combinations of these corner sections that result in a unique impossible tribar. The same principle can be extended to any n-sided polygon. A similar method is also introduced for the impossible solids class of objects, such as the impossible trident.

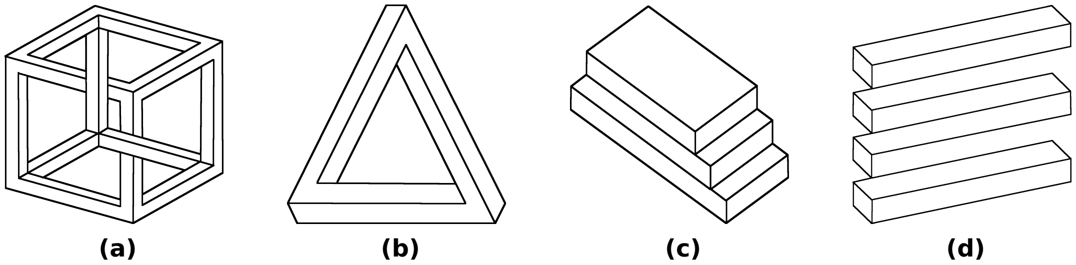


Figure 2.8: Examples of four common types of impossibity. (*a*) The Impossible Cube. (*b*) The Impossible Triangle. (*c*) The Impossible Staircase. (*d*) The Impossible Shelves.

Ernst [17] describes four key types of impossible illusion based on depth and spatial inconsistencies within the objects. Figure 2.8 shows an example of each of the four identified impossible classes:

- **Depth Interposition.** These illusions are caused by the structural inconsistency of the object due to irregularities in the depth ordering. The impossible cube, Figure 2.8a, where parts of the cube appear to be in front and behind at the same time.

- **Depth Contradiction.** These illusions are caused by a difference in the local three-dimensional information resulting in inconsistencies in the global object structure. The impossible triangle, Figure 2.8b, where the three sides of the triangle join to form an impossible cycle.

- **Dissapearing Normals.** These illusions are caused by the appearance of a twist on the surface of the object, at one end the surface may appear vertical whilst at the other, it may appear horizontal. Causing the object to have an inconsistent normal direction across the surface. The impossible staircase, Figure 2.8c, where one object face gradually transitions into another.

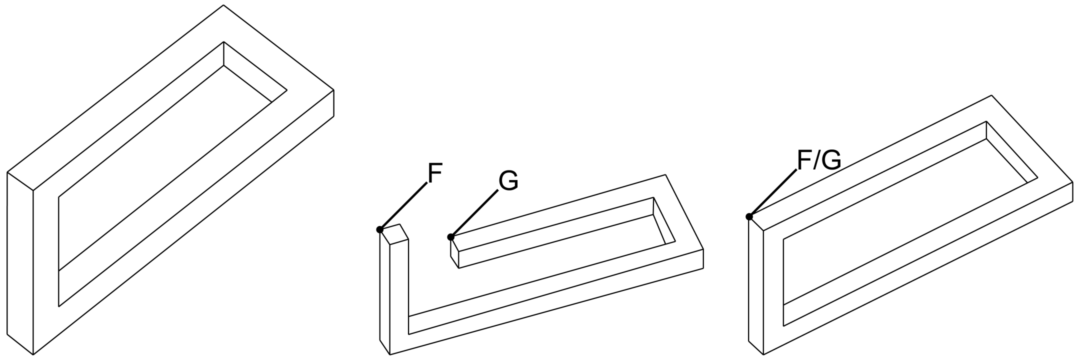
- **Disappearing Space.** These illusions are caused by the silhouette of the object not being fully closed. The impossible trident, Figure 2.8d, where sections of the object seem to disappear completely.

Each impossible figure can fall into one, or more, of these categories. By categorizing objects based on the nature of their impossibility for our work we can better understand whether different impossibility types require different solutions when it comes to modeling and rendering their three-dimensional representations.

2.1.2 Computer Generated Impossible Objects

To aid in creating new and more complex versions of these impossible figures Uribe [80] proposed a set of triangular tiles each containing a colored side or corner of a rectangular bar. Using sets of these tiles joined together to create different constructions of the impossible multi bars. These are however still two-dimensional representations of these impossible figures. Whilst cases of impossible figures can be seen in pictorial art throughout the last century and beyond, their presence in more modern media is still relatively new. With cases such as Monument Valley and Inception, we see these impossible figures being brought to life in three-dimensional form, as opposed to the traditional two-dimensional drawings and paintings.

We often view impossible objects from an incidental viewpoint or an ’illusion viewpoint’. As our view of the object changes and we move further away from this ’illusion viewpoint’ the true nature of the structure is revealed. It becomes apparent that the structure may contain hidden disconnections or deformations. A benefit of creating these objects in three-dimensional space is the ability to manipulate and view these impossible objects from different angles. Khoh and Kovesi [42] generated novel views of impossible figures by using two complementary halves, where one half of the three-dimensional model is created through inverse transformations in the image plane. This method allows for unrestricted viewpoints, however, the final rendered object can suffer from severe distortion. Owada and Fujiki [61, 62] implement a constraint solver to seamlessly connect separate three-dimensional possible sections in a projected two-dimensional domain. By maintaining connections between user labeled edges their system is able to render line drawn impossible figures from novel viewpoints without losing the visual continuity of connected lines. Savransky at al. [70] proposed a similar approach to modeling seemingly impossible three-dimensional models through transformations to disconnected locally possible sections, such as those proposed by Kulpa [45], creating the illusion of a connected three-dimensional impossible model when projected onto the 2D image plane.



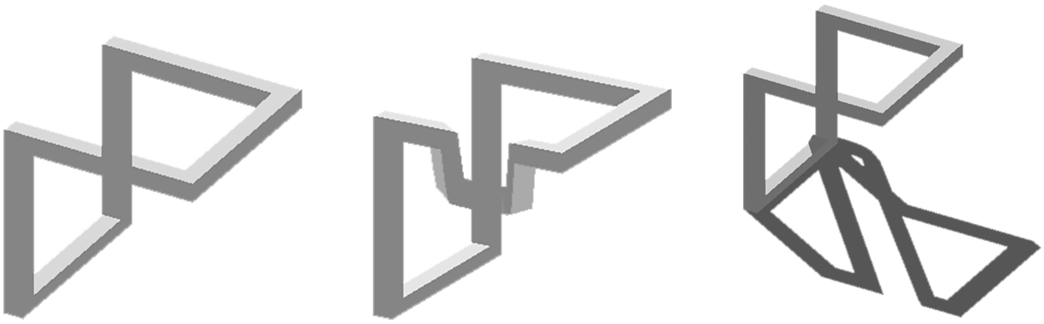
(a) The original 'illusion viewpoint' of the impossible object.

(b) When the viewpoint is changed points F and G become misaligned.

(c) Sections of the object are stretched and scaled to realign points F and G and recreate the illusion of impossibility.

Figure 2.9: This figure is a reproduction of the work by Nakatsu et al. [56]. Beginning with a disconnected possible model their method creates the illusion of impossibility by scaling and stretching sections of the object such that points of the object are aligned when projected orthographically.

In many cases although these objects appear impossible they are actually possible objects that exploit visual tricks to create the illusion of impossibility. One common way this is done is by giving the viewer the impression that the object is a single connected structure when viewed from it's 'illusion viewpoint'. When viewed from an alternate viewpoint it is revealed that the structure contains a hidden disconnection. To allow for these disconnected impossible objects to be viewed from novel viewpoints and maintain their illusion of connectivity Nakatsu et al. [56] propose a transformation based solution. Given a disconnected object, shown in Figure 2.9, their method creates the illusion of connectivity by scaling and stretching the sections of the object such that points G and F are drawn at the same position when the object is orthographically projected onto the image plane. Chiba et al. [11] further extends this to allow the objects to be rendered with both shading and textures. This method, however, is only capable of handling a single type of impossible objects, specifically, they showed only one example of an impossible multibar in their publications.



(a) Impossible object viewed from the ’illusion viewpoint’.

(b) An alternate view of the object reveals the hidden structure.

(c) The object rendered alongside its traditional cast shadow.

Figure 2.10: An Impossible object model created using Elber’s [15] deformation technique. Deformations are applied along the line of sight such that they are hidden from view. These hidden deformations cause corresponding deformations in the objects traditional cast shadow.

An alternate method of creating the illusion of impossibility within these objects is through hidden line of sight deformations within the structure of the object. Elber [15] proposes a deformation based modeling method that manipulates the geometry of the object along the current line of sight to create a connected seemingly impossible model. Figure 2.10 shows an example of an impossible object modeled using Elber’s line of sight deformation method. This illusion is created due to the apparent orthogonality of the object. When we view the same object from an alternate viewpoint, away from the ’illusion’ viewpoint, the deformations within the structure are revealed. Wu at al. [84] propose an alternate deformation based method, where locally possible three-dimensional sections are connected through hidden deformations to give the illusion of a globally impossible three-dimensional model. Their method is able to simulate all four types of impossible objects. Table 2.1 summarizes the characteristics of the approaches designed to model and render impossible objects.

When dealing with deformation based approaches problems can occur when applying shading and textures to the object. The deformations applied to the geometry of the object also change the way the shading and textures behave on the surface of the object. Further steps must be taken in order to achieve appropriate looking textures and shading for these objects. Tsuruno [79] presents the Mimetic Surface Colour and Texture (MSCTA) method

Method	Classes	Modeling Approach	Geometric Input	Viewpoint	Projection	Rendering	Environment
Savransky et al.(1999)	A, B	3D Tiling	3D Mesh	Restricted	Parallel	Shaded	Dedicated
Khoh and Kovesi (1999)	A	Edge Alignment	3D Lines	Unrestricted	Parallel	Line Drawn	Dedicated
Uribe (2001)	A	2D Tiling	2D Tiles	Fixed	Parallel	Shaded	Dedicated
Owada and Fujiki (2008)	A, B, C ,D	Edge Alignment	3D Mesh	Restricted	Perspective	Line Drawn	Dedicated
Wu et al. (2010)	A, B, C, D	Deformation	3D Mesh	Restricted	Perspective	Shaded, Textured	Dedicated
Elber (2011)	A, B	Deformation	3D Mesh	Restricted	Perspective	Shaded, Textured	Dedicated
Nakatsu et al. (2012)	B	Transformation	3D Lines	Restricted	Parallel	Line Drawn	Dedicated
Chiba et al. (2018)	B	Transformation	3D Mesh	Restricted	Parallel	Shaded, Textured	General
Our Method	A, B, C, D	Transparency	3D Mesh	Restricted	Perspective	Shaded, Textured	General

Table 2.1: A comparison of existing impossible object modeing and rendering methods. Classes indicate the type of impossible figures that the methods can handle - 1:Depth Interposition, 2:Depth Contradiction, 3:Dissapearing Normals and 4:Dissapearing Space. See Figure 2.8 for examples.

to produce naturally shaded and appropriately textured three-dimensional impossible models. By utilizing geometry from a disconnected undeformed model of the object, it is possible to recolor the surface of the deformed impossible model and remove any anomalies. These deformation based methods produce extra steps that we aim to avoid in our own transparency based modeling method. As our method does not manipulate the geometric structure of the object we are able to shade and texture the objects without additional steps whilst also being able to accurately reproduce objects containing each of the four types of impossible illusion.

From Table 2.1 it can be noticed that only Khoh and Kovesi [42] provide truly unrestricted views of their impossible objects. Under specific viewpoints, however, the objects can undergo severe distortions, especially at some ’degenerate’ viewpoints where the width of the object is scaled to near zero. All other methods implement a restricted set of viewpoints, under which appropriate looking impossible objects can be generated. One theory for this is that impossible figures are only feasible from a finite set of viewpoints. It can also be noted that each of the previous modeling methods operates using their own dedicated rendering environments, only Chiba et al. [11] have implemented their modeling algorithm using a general rendering environment. One goal of our work was to be able to implement each of our algorithms without the need for dedicated environments, as such, both our transparency based modeling algorithm and copycat cast shadow algorithm have been implemented using the Unity Engine, similar to Chiba et al. [11]

Whilst Tsruno's [79] method accounts for the surface shading of the deformed object it does not account for how these hidden deformations and disconnections will affect the objects cast shadow. Instead to compensate for the deformed cast shadows that were cast by the rendered surfaces in his work, Tsruno purposefully darkened the background of the images so as the shadows became less noticeable. This is a common occurrence throughout the previous modeling methods, objects are presented lacking any form of cast shadow. The illusions used to create the appearance of an impossible object that are hidden from the observers' viewpoint may be revealed in the objects cast shadow. These hidden disconnections and deformations result in anomalies within the corresponding cast shadow, as seen in Figure 2.10c. These anomalies may detract from or break the objects illusion of impossibility and as such are simply excluded from the rendered images. Hori [32] demonstrates the use of impossible objects in virtual reality, evaluating how the impossibility may behave if viewed in the real world. To create the most realistic impression of these objects they are rendered alongside their cast shadows, as they would be in the real world. This creates a higher sense of realism, however, only a single simple object is presented. When viewing more complex impossible objects the cast shadows may appear unnatural for viewers. We attempt to solve this problem of casting visually appropriate shadows for impossible objects by implementing a screen space occlusion algorithm, allowing the objects to be rendered in more complex and realistic scenes.

2.2 Rendering Cast Shadows

Simulating global illumination effects has long been a focus of computer graphics research, as such a large body of work exists on the range of methods that have been devised. We refer the reader to surveys by Ritschel at al. [67] and Hasenfratz et al. [31] for a better overview of the subject. Here we focus on some of the more relevant and real-time methods.

Shadows are a key part of graphics rendering, dramatically improving image realism [22]. Without them, scenes often feel unnatural or flat. Cast

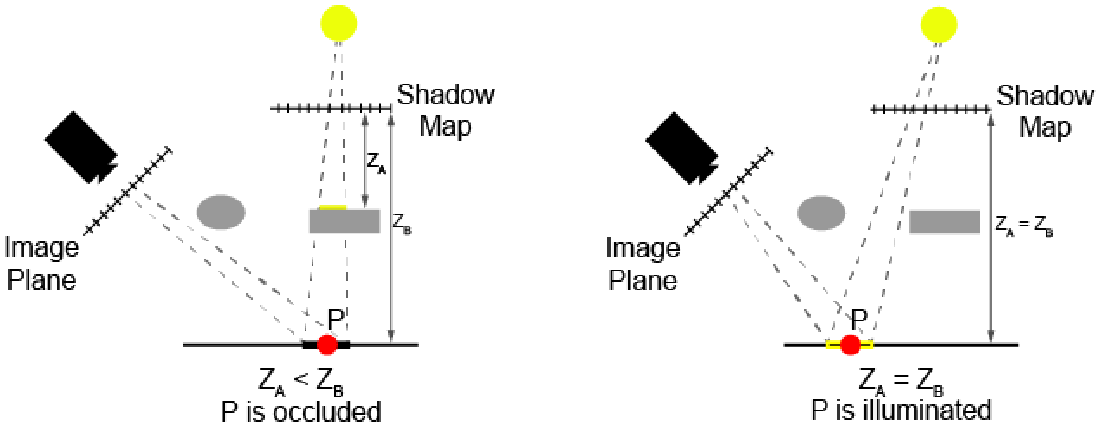


Figure 2.11: Shadow mapping overview. A shadow map is generated from the point of view of the light source, occlusion is then calculated by comparing the depth information from the camera view direction to those stored within the shadow map. If the depth from the view direction is greater than that stored in the shadow map then the point is occluded.

shadows aid in our understanding of three-dimensional environments, object position, and size along with geometric information. While soft shadows provide a heightened sense of realism to an image. With improvements in graphics hardware and cast shadow algorithms, it is possible to render convincing soft shadows in real-time. One common technique of this is the use of shadow maps [83]. Shadow mapping works by identifying areas of the scene that are hidden from the light source, by first computing a view of the scene from the point of view of the light source. A second step is then taken to render the scene from the point of view of the camera. By comparing both depth results it is possible to compute those areas in the scene that are occluded. Figure 2.11 demonstrates a simple example of the principals behind shadow mapping to calculate occlusion. One of the main advantages of the shadow mapping method is its efficiency, especially when compared to geometric based methods. Due to their speed shadow maps can provide fast efficient methods for generating cast shadows.

The basic shadow mapping techniques cannot produce soft shadows, therefore techniques have been developed that allow these algorithms to render soft shadows. One of the more popular of these algorithms is Percentage Closer Filtering (PCF) [66]. When sampling the collected buffers to calculate occlusion instead of only considering the current pixel data, the neighborhood around the pixel (known as a kernel) is also considered. The percentage of

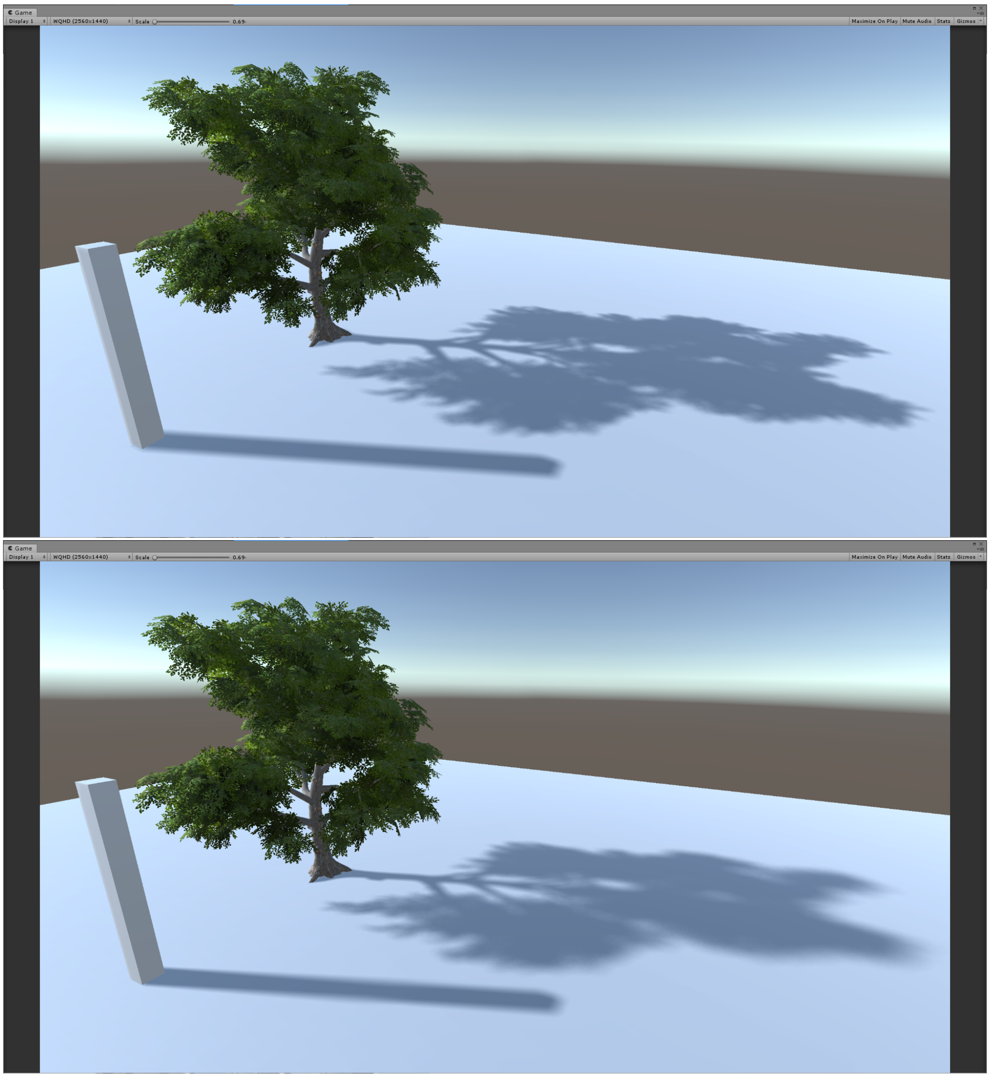


Figure 2.12: (*Top*) Uniform soft shadows, the penumbra of the shadow remains constant. (*Bottom*) Percentage Closer Soft Shadows, varying shadow penumbra. Softer shadows are observed as the distance from the occluder increases.

these neighboring pixels that are also occluded is calculated. If the value is 0 then the sampled pixel is illuminated, if the value is 1 then it is fully occluded. Any value between 0 and 1 means the pixel lies in the penumbra of the shadow, the partially shaded outer region of the shadow, and the percentage value determines how dark the pixel appears. While PCF can produce visually satisfying results the inclusion of the kernel increases the number of times the shadow map is sampled, impacting the speed of the algorithm. To further increase the realism of this method the Percentage-Closer Soft Shadow [21] (PCSS) algorithm creates contact hardening shadows by varying the PCF kernel size at each pixel in relation to the size of the light source and distance from the occluding object. This results in cast shadows that become sharper as objects contact one another and softer the further apart they are, see Figure 2.12.

As algorithms like PCF and PCSS require multiple samples from the texture map to work. Gumbau et al. [30] describe the approach of Screen Space Soft Shadows (SSSS). Their approach employs the use of a separable Gaussian kernel, applied in screen space. This kernel is used to smoothen and blur the standard shadow from the observers' point of view to create the shadow penumbra. As their kernel is separable the number of samples needed to create the shadow penumbra is lower compared to other soft shadowing approaches, resulting in a higher performance of their algorithm.

For our implementation of cast shadows, we focused on the process of screen space occlusion. As impossible objects are only impossible from the observers' current viewpoint the appropriate occlusion method would also need to apply along the view direction. Instead of using original object geometry to calculate occlusion, screen space occlusion creates a simplified reconstruction of the scene geometry through the use of frame-buffers. These buffers contain information for each pixel in the image space, position, normal, and material. Using this information it is possible to calculate an occlusion value for each pixel. This method was first implemented to calculate real-time ambient occlusion [1, 2, 53]. Ambient occlusion is a global lighting method used to increase the realism of a scene by calculating how each point in the scene is affected by ambient lighting. Figure 2.13 demonstrates an example of ambient occlusion, ambient occlusion becomes most noticeable within the corners of objects within a scene. Screen Space Ambient Occlusion (SSAO) aims to compute the amount of occlusion for each point in the image by sampling the surrounding points from the buffered information. By constructing a sampling area around the image pixel it is possible to compute whether a point is occluded by any geometry by testing its depth against the buffered depth information [47, 63]. The benefit of this screen space implementation is its speed, as the method is independent of the scene geometry it operates the same way for each of the pixels of the image allowing a real-time imple-

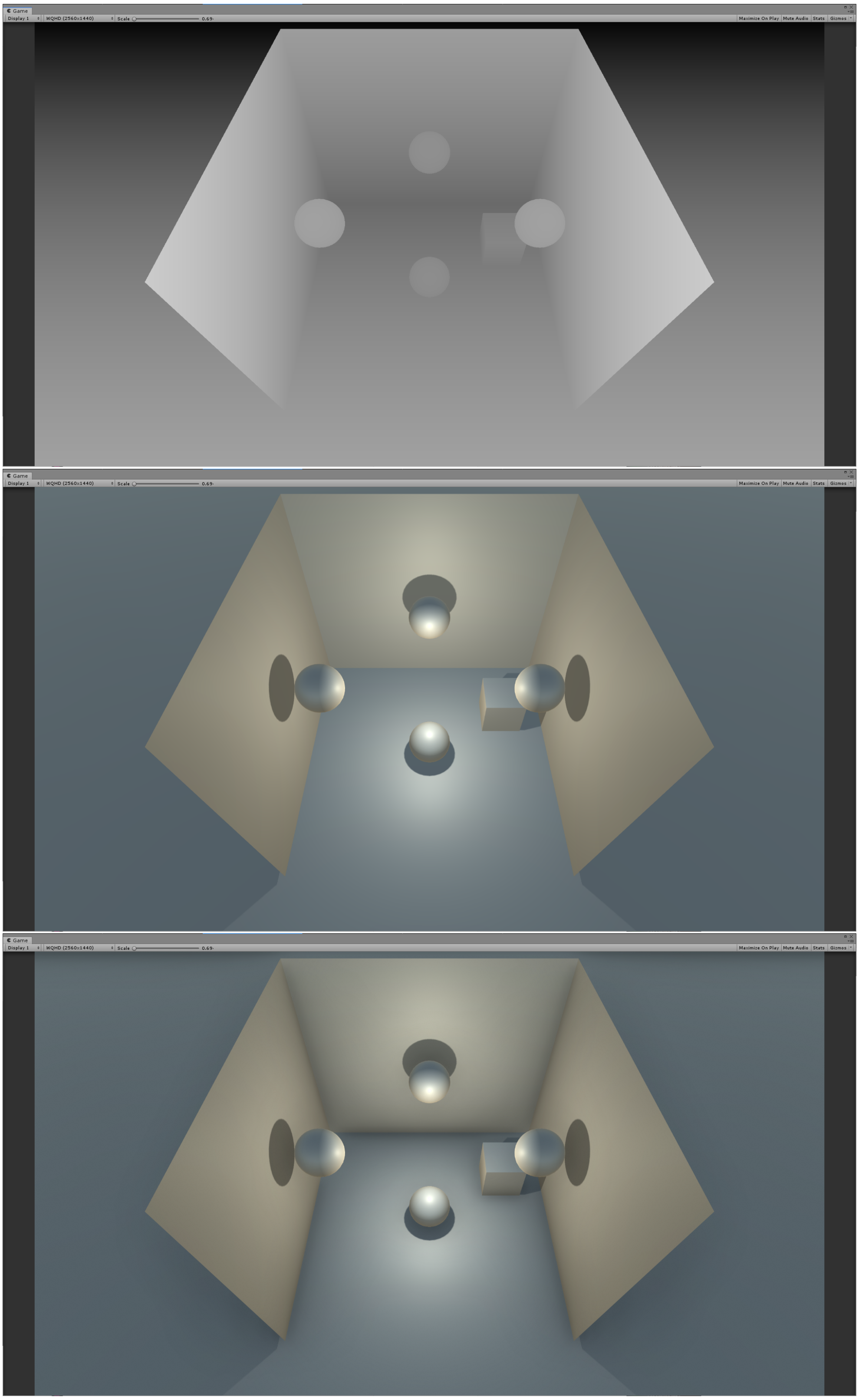


Figure 2.13: (*Top*) Depth buffer for the scene, darker pixels indicate a greater depth. (*Middle*) Scene rendered with no ambient occlusion. (*Bottom*) Scene rendered with ambient occlusion. Ambient occlusion darkens areas where objects are close together, such as corners, creating a more realistic lighting effect.

mentation. Due to this balanced trade-off between speed and computational cost, this effect is popular in real-time rendering and games [55, 73, 76].

This screen space method of occlusion has been extended to implement other global illumination effects [58, 59, 57, 72]. In particular, methods of directional lighting have been implemented to make use of the advantages of operating in screen space. Ritschel et al. [68] describe a method of approximating dynamic global illumination in image space, including the ability to calculate ambient occlusion, directional occlusion, and indirect color bleeding between objects. McGuire and Mara [52] present an efficient solution for screen-space three-dimensional ray tracing to render a scene. To calculate directional occlusion in screen space their method marches a sample point along a three-dimensional ray for a bounded distance. By projecting each three-dimensional sample point into screen space it can be classified as a ray hit if the point is behind the depth buffer at that pixel. By sampling ray intersections in screen-space, they are able to achieve a high-quality approximation of the scene at an efficient rate compared to using a full geometric representation of the scene. This method of ray tracing against the depth buffer can be used to implement features such as ambient occlusion, reflection, refraction, and even full-screen space path tracing.

Our occlusion algorithm aims to exploit the advantages of screen space methods by only operating on the buffered information. Doing so we are able to manipulate the stored depth information at an interactive rate to change the resulting cast shadow. By operating in screen space we are also able to implement our shadow algorithm without changing the existing rendering pipeline. Implementing our shadow algorithm using screen space data allows us to incorporate it within existing rendering environments allowing users to take advantage of the exiting tools they use, avoiding the need for a dedicated rendering environment.

Screen space methods are not exclusive to global illumination effects [37, 40]. Due to the speed and efficiency of screen space operations, many Non-Photorealistic Rendering (NPR) effects have been implemented. For a wider overview of these, we refer the reader to existing surveys [46, 48, 71]. One common NPR effect implemented in screen space is the process of drawing outlines or features lines [38, 54]. These outlines aid in emphasizing the edges or creases of an object's structure. As we often find impossible figures represented as simple line drawings, the use of these outlines on three-dimensional impossible objects may produce interesting results. We also consider our screen space shadow algorithm a form of NPR rendering as the aim is not to produce the most photorealistic result, rather we aim to produce impossible copycat shadows.

2.3 Perception of Cast Shadows

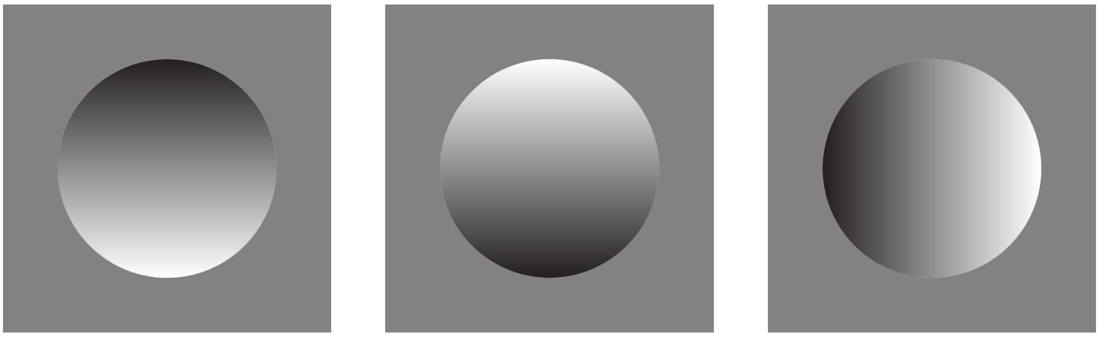


Figure 2.14: Example of stimulus used by Ramachandran [65]. Due to the human visual systems assumption that light comes from above the gradient within the first two circles creates a different interpretation when viewed. (*Left*) Concave and (*Middle*) Convex. The (*Right*) circle becomes more ambiguous as the human visual system is not used to interpreting light coming from horizontal directions.

Shadows and shading have been shown to have a large effect on a viewer's perception of a scene adding both a heightened sense of realism and conveying important spatial and positional information. Knill et al. [44] and Ramachandran [65] demonstrate that from object shading alone our visual system can interpret important features such as the shape of the object. Ramachandran [65] demonstrates using shaded two-dimensional circles, a sample is shown in Figure 2.14, how our visual system perceives three-dimensional concave cavities and convex spheres. Due to our brains' assumption that light comes from above in these images, as in the real world, our visual system is tricked into interpreting these flat shapes as three-dimensional objects. When the shading is applied horizontally these shapes become more ambiguous as our brain is not used to interpreting light directions from a horizontal angle. This work demonstrates that from surface shading alone viewers are able to perceive three-dimensional depth within a two-dimensional image, an important fact to consider when looking at deformation based impossible objects. Although the sections of these impossible objects may appear flat from the observers' viewpoint, inconsistencies in the shading of the surface due to the hidden curvatures within the structure of the object may reveal its true shape, detracting from the objects impossible illusion. By leaving the geometry of the object unchanged our transparency based modeling method does not cause inconsistent surface shading.

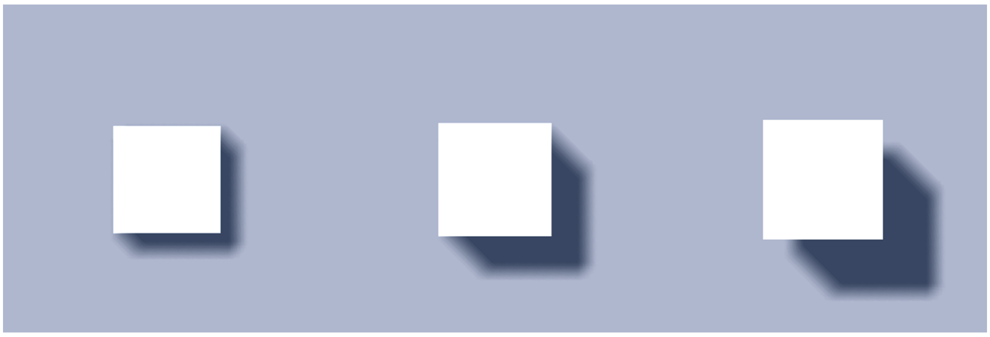


Figure 2.15: As the shadow moves further away from its casting object our visual system interprets the object as getting closer in depth.

Cast shadows have also been shown to have an impact on a viewer's perception of object depth and positioning within a scene. Casati [6] reviews the different types of information that can be conveyed through cast shadows. The presence of a cast shadow within a scene but not its casting object indicates the presence of hidden casting objects, either positioned outside the field of view or occluded by another object in the scene. From cast shadows, the viewer can infer the light source location and intensity, or even the presence of multiple light sources. The motion of a cast shadow carries information about the three-dimensional structure, depth, position, and motion direction of its casting object. This is also shown in work by Kersten et al. [41] and Mamassian et al. [50]. The position of a cast shadow in relation to its casting object gives the viewer important cues as to the depth of the object within the scene, see Figure 2.15.

Wanger [82] demonstrates how cast shadows can inform viewers of an objects' geometric shape by tasking viewers to solve a shape matching test. Viewers were shown a set of uniquely shaped objects that appeared identical when viewed from their base. These objects were rendered using different shadow conditions, a similar scene we have created is shown in Figure 2.16.

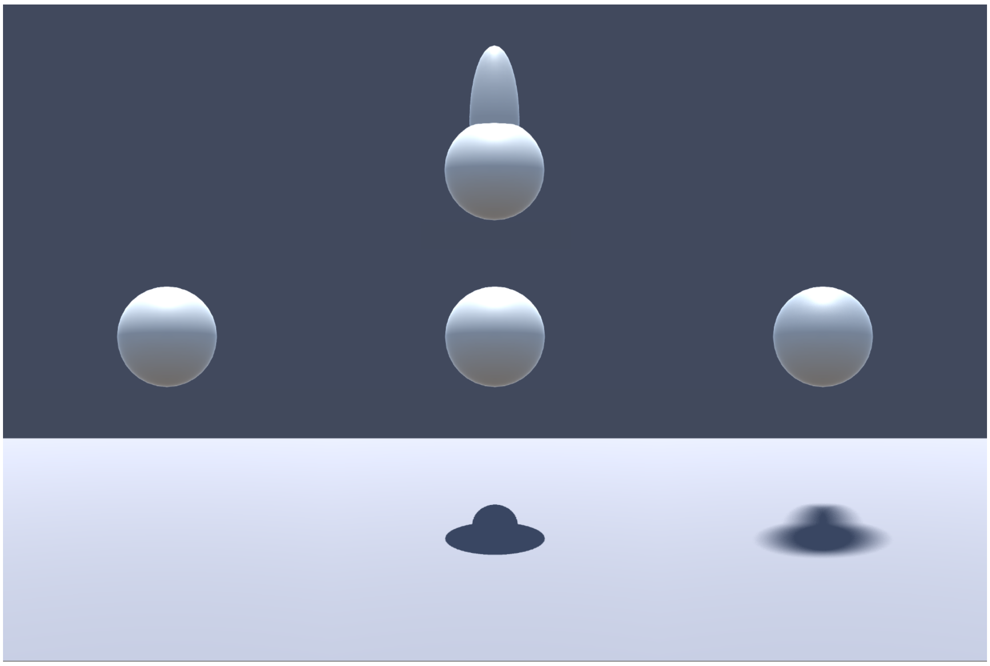


Figure 2.16: A similar example to stimulus used by Wanger [82]. The three lower objects in this scene are identical to the top object, only viewed from below. By exploring different strengths of soft shadows Wanger demonstrates that hard shadows allow viewers to correctly estimate the shape of an object compared to softer shadows.

They were then given a test object which was identical to one of the other objects only rotated to be viewed from the side. Viewers were tasked with matching the test object to its corresponding shape in the set of other objects. It was discovered that sharper shadows have a positive effect on a viewer's ability to correctly estimate the geometric structure of its casting object. Similar results were also found by Castiello [9], where a viewer's perception of object shape is hindered when the objects are presented alongside incorrect cast shadows. This feature of cast shadows may be the reason why in previous impossible object modeling research we see the models rendered without any form of cast shadows. The hidden geometric shape of the impossible model results in deformed cast shadows which can reveal the shape of the object to the viewer and detract from the illusion of the object. Our cast shadow algorithm aims to solve this problem by casting shadows based on the visible profile of the object rather than its full three-dimensional shape. Conflicting results, however, have been reported by Braje et al. [3] where experiment results suggest that object recognition is not affected by the presence of cast shadows, with cast shadows being filtered out at an early level of visual processing. The authors report however this may be dependant on the type of stimulus used which contained redundant information, due to the simplicity and familiarity of the object used, which could reduce the importance of the cast shadows.

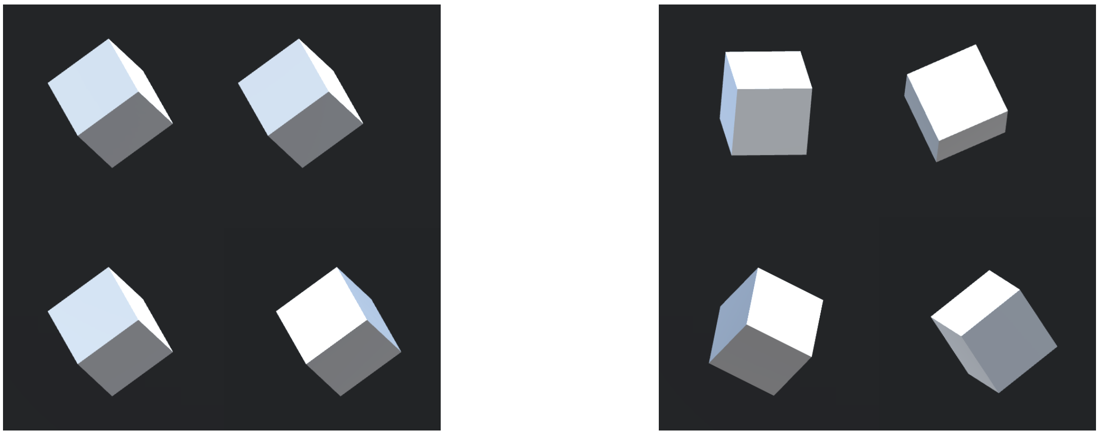


Figure 2.17: Example images, similar to the stimulus used by Ostrovsky et al. [60]. *(*Left) Objects in homogeneous orientation. *(Right)* Objects in heterogeneous orientation. In both images, the bottom right object is illuminated from a different light direction to the other objects. Under heterogeneous orientation this illumination inconsistency becomes more difficult to detect as shown by Ostrovsky et al.

Although cast shadows have been shown to play an important role in aiding viewers to correctly interpret spatial and positional relationships of objects within a scene, the human visual system can also be insensitive to ambiguous cast shadows. Psychological research suggests that the human visual system uses a coarse representation of cast shadows, discarding the finer details. Ostrovsky et al. [60] demonstrate the human visual system's insensitivity to oddly illuminated objects and scenes. Their first study involved the presentation of a set of identical objects randomly orientated with consistent shadows and shading, but with one object from the set illuminated from a different direction. The previous studies they expand on suggest that the illumination inconsistency was easily noticeable, due to a pop-out effect [16, 75]. Ostrovsky et al. show this may have been due to the homogeneity in the orientation of the object set. When the set of objects are orientated in a heterogeneous fashion the illumination inconsistency is more difficult to perceive. Figure 2.17 shows a similar setup to those used by Ostrovsky et al. under both homogeneous and heterogeneous orientations. We draw inspiration for one of our visual perception experiment designs from the one used by Ostrovsky et al. with the aim to investigate how sensitive viewers are to impossible objects among a set of possible object distractors.

A second experiment performed by Ostrovsky et al. [60] investigated a viewer's ability to detect inconsistent illuminations within digitally manipulated images. Results showed that viewers were insensitive to detecting the modified images with near chance level performance. Furthermore, viewers within this test were explicitly told to look for illumination inconsistencies within the images, making it plausible that the unprimed subject's performance would be even lower. Similar results were found more recently by Farid and Bravo [20] who demonstrated observers' sensitivity to inconsistent illumination in scenes was only seen when the combination of light sources came from opposite sides of the room (the shadows were cast in opposite directions). Presenting evidence that the human visual system is not capable of detecting simple inconsistencies in the position of a light source within the scene, only under extreme circumstances. Sattler et al. [69] aim to take advantage of the human visual systems' insensitivity to inconsistent illumination for the purposes of computer graphics. Evaluating the required level of detail needed within a cast shadow to produce results that are acceptable to the human visual system. With their final aim being to use simplified object models within their scene to reduce the complexity of their rendering algorithm.

Evidence of the human visual systems' insensitivity to these inconsistent illuminations can be seen when looking at artistic depictions of the real world. Throughout history, artists have explored the use of perspective, color perception, and visual illusions. Often creating their works under what Cavanagh [10] refers to as an 'alternate physics'. Instead of adhering to physical rules of lighting and shadows that are present in the real world, artists often relax these constraints. Manipulating the shadows and shading in their work to fit their artistic style or create a more visually appealing piece of art. Jacobson and Werner [39] investigated how sensitive the human visual system is to the appearance of cast shadows. Using a visual search experiment where viewers were tasked with detecting scenes that contained an 'odd' cast shadow, this being a shadow that was cast in an opposite direction to the other cast shadows in the scene. The results of the experiment indicate that viewers were insensitive to the inconsistencies in cast shadows.

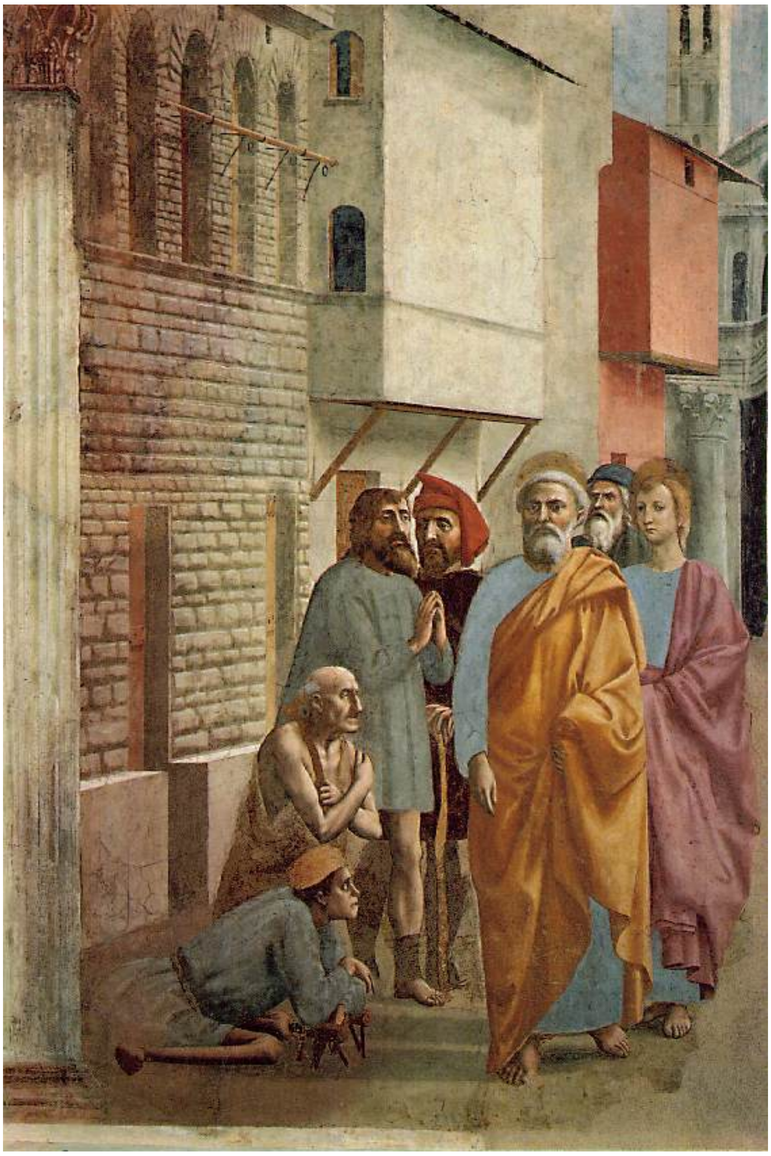


Figure 2.18: St Peter Healing the Sick with his Shadow, Masaccio, 1426-1427. Example of pictorial art containing impossible shadows. In this scene, both the cast shadow and the characters play an important part in the narrative. So as not to obscure the character in the foreground the artist has chosen to use an impossible shadow, where the shadow can be seen on the ground but is transparent when casting on the character in the foreground.

Supporting their claims by highlighting the use of shadows in pictorial art where artists are seen to carefully arrange the cast shadows in their scene so as not to obscure focal points in their work. Figure 2.18 shows the scene from Masaccio's 'St Peter Healing with His Shadow'. The scene is shown as Saint Peter's shadow is passing over the figure in the foreground, both elements are critical to the narrative of the scene. To ensure no critical elements are obscured by the shadow Masaccio has made the shadow visible on the ground but transparent on the figure of the man. Whilst this is impossible in the real world it does not appear evident or jarring to the viewer in this work.

Casati [8] highlights a specific shadow phenomenon present in pictorial art called copycat shadows. These shadows do not aim to accurately represent the geometric structure of the object, rather their shape mimics the visible profile of the casting object. Although these copycat shadows are impossible cast shadows that would not be found in the real world, only under limited light and object positions, they are often a simplification compared to the objects traditional cast shadow. This simplification along with the similarity between the shape of the shadow and its casting object means when viewing these copycat shadows observers may find it easier to firstly, correctly identify the dark area as a shadow, and secondly, correctly match the shadow to its casting object. This visual perception problem of associating a shadow with its corresponding casting object, known as the shadow correspondence problem, can be difficult within a complex scene or where the object and shadow are far apart. Solving this problem correctly is important to gain an accurate understanding of the scene, in particular determining the location and position of objects within the scene. Casati suggests that copycat shadows may make this task easier due to the natural similarity between the shape of the object and the shape of the copycat shadow.

Mamassian [49] investigates the human visual systems' effectiveness at solving the shadow correspondence problem when impossible shadows are used. Suggesting that the human visual system uses a coarse representation of the scene to solve this correspondence problem. Relying on a wider view of the shapes within a scene rather than the finer details. This result supports the case for copycat shadows as our visual system may resort to pattern matching solutions when viewing these copycat shadows, solving the correspondence problem based on the similarity between the shape of the object and copycat shadow. Our shadow algorithm allows the creation of these copycat cast shadows automatically, where previous methods would have relied on manual post-processing effects. Due to the nature of copycat shadows ignoring the complex three-dimensional shape of their caster and aiming to replicate the visible profile, along with the insensitivity of the human visual system to impossible shadows, we aim to use copycat cast shadows as a solution to producing compelling, visually appropriate cast shadows for impossible objects.

2.4 Perception of Impossible Objects

Impossible figures have often been a focus of visual perception research in an attempt to better understand how the human visual system constructs three-dimensional objects from two-dimensional images. Dror and Ivey [14] utilized impossible objects within their work to test how three-dimensional objects are stored by the human visual system, using either a local or global representation when performing mental rotations of the objects. Impossible objects were chosen as they are not encoded as global representations, or at least it is more difficult to do so compared to possible objects. This is due to the impossible feature within the design of the figure that makes their structure locally possible but globally impossible. Results suggest a local representation of the object is stored in visual memory as participants performed comparably for both possible and impossible figures.

These impossible figures have also been used to estimate the order in which the human visual system processes features within a scene. Friedman and Cycowicz [29] suggest that the human visual system processes the global structure of the scene prior to examining the finer details. Their test involved measuring the object recognition time for both possible and impossible figures, tasking participants to label objects as left or right facing. While their results show participants being able to effectively distinguish between the two types of figures, participants performed much slower for impossible figures, suggesting the impossibility is examined during late hight levels of the visual process. Work by Freud et al. [24, 25] supports these findings. Suggesting the differences between object types only emerge after the initial recognition process. Where both possible and impossible objects are processed using the same early-stage visual processes at first, with the impossibility of the object only being processed at a later stage.

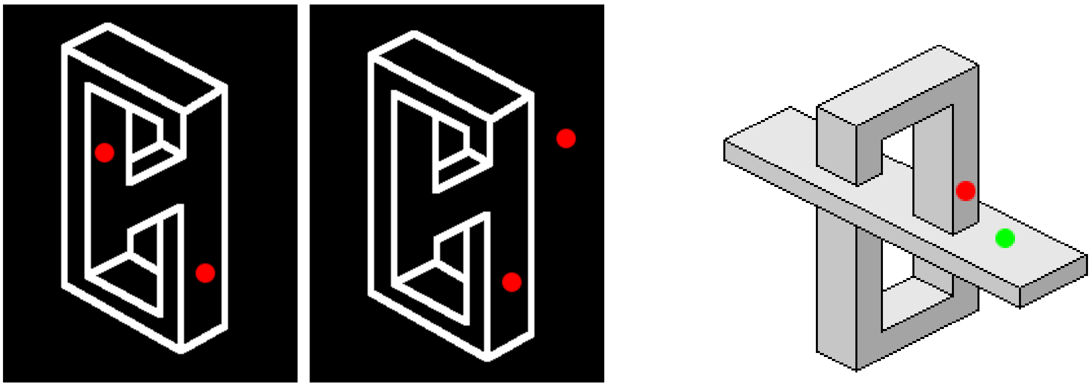


Figure 2.19: Replicated stimulus used by Freud et al. [27, 26]. Exploring depth perception within impossible objects by superimposing fixation markers onto the object’s surface. These experiments inspire the design of the visual perception experiments we conducted.

Despite being impossible in their constriction these impossible figures still contain valid depth cues when viewed, enabling them to be interpreted by the human visual system as a projection of a three-dimensional object. Freud et al. [27] explore how the human visual system processes two-dimensional figures in order to construct three-dimensional mental representations. By superimposing fixation markers on the surface of two-dimensional figures participants were asked to perform same/different classifications on the locations of the fixation markers, see Figure 2.19 for an example of the stimulus used. Their results suggest that initially both possible and impossible figures are processed in a similar way, using a ’gist of the scene’ with the finer details not being encoded. Only at a second higher-level stage is the detailed information of the scene encoded, resulting in slower processing of impossible objects.

In later work Freud et al. [26] use a similar method of superimposing fixation markers onto the surface of impossible objects, with participants being asked to determine which marker is closer in depth. When designing our perception experiments we aimed to use methods that had already been established, this would ensure we had a proven basis for our experiment design from which to compare our results. For our second perception experiment, we aim to follow this design created by Freud et al. [26]. Which also inspired the design of our third and fourth experiments.

Although both cast shadows and impossible figures are prevalent subjects of visual perception research the two have not been combined previously. It should also be noted that the stimulus figures used in previous research were two-dimensional impossible figures, created through line drawings and basic shading. However with our ability to both model and render three-dimensional versions of these impossible figures we are able to perform visual perception experiments using the three-dimensional models. Allowing us to view these objects from new viewpoints that differ from their traditional ’illusion viewpoint’. We are also able to fully render these objects alongside cast shadows, allowing us to explore how these two factors behave together. As impossible figures are inherently impossible there is no way to know the exact shadows they should cast, we conduct a series of visual experiments using both traditional cast shadows and our copycat shadow algorithm.

Chapter 3

Impossible Object Modeling

In this chapter, we present a new method for modeling three-dimensional representations of impossible figures through the use of transparency. Given a regular three-dimensional model, our approach simulates the effects used to create the illusion of impossibility within an object by manipulating the surface transparency of the object sections. We present an algorithm to calculate the surface transparency based on distance from a user-supplied axis, with the ability to adjust factors such as the rate of change of transparency. We demonstrate how transparency can be used to simulate a range of different impossibility types, including those created through depth ambiguities and surface manipulations. By implementing the use of object scaling we demonstrate how it is possible to model these impossible objects from novel viewpoints whilst maintaining the illusion of impossibility. Finally, we implement our modeling algorithm within the Unity engine avoiding the need for any form of dedicated rendering environment.

3.1 Introduction

The process of modeling three-dimensional representations of impossible figures has received attention in the past, with many previous methods focusing on using hidden deformations to create the illusion of impossibility within an object [15, 84]. These deformation based methods manipulate the structural geometry of the original object. Our method does not change the geometric structure of the original object but instead alters its surface transparency. By manipulating the surface transparency of an object we are able to simulate the illusion of impossibility. Figure 3.1 summarises the steps

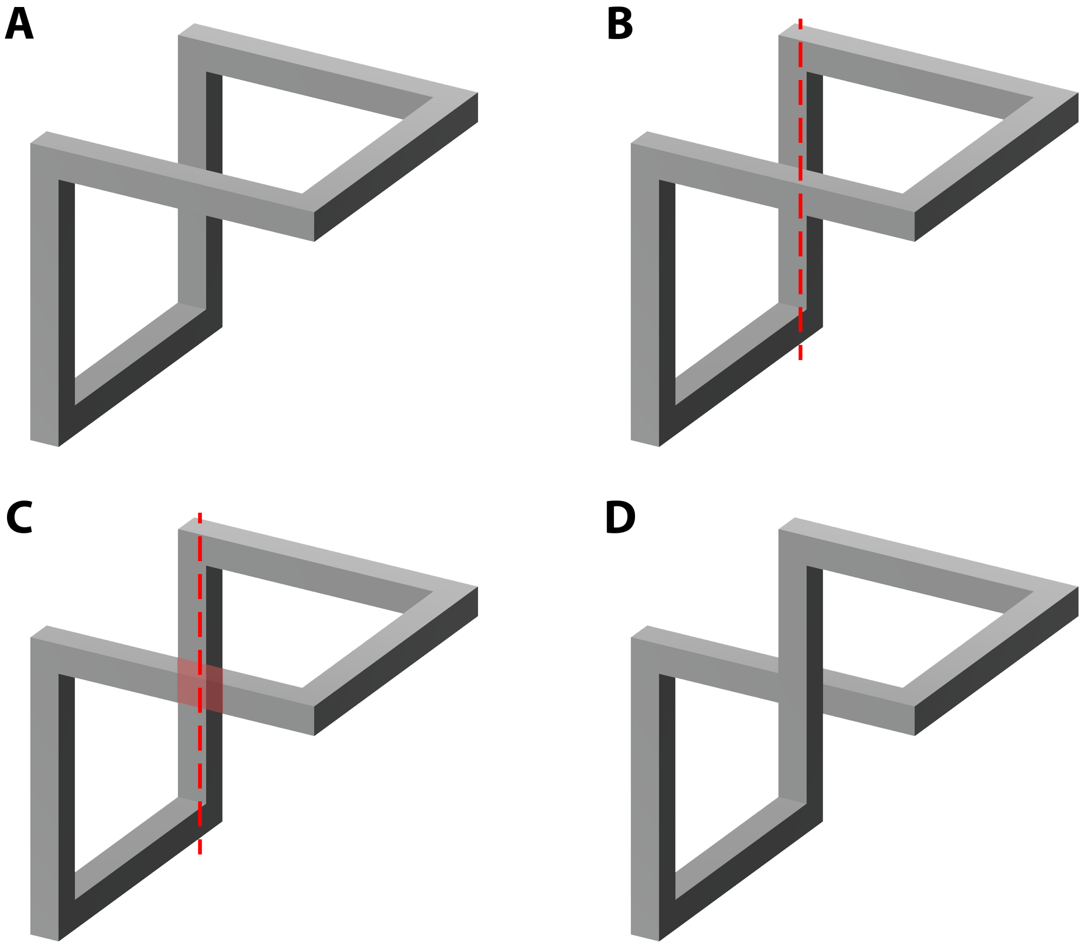


Figure 3.1: Applying transparency to an object to create the illusion of impossibility. (*A*) The original three dimensional object. (*B*) Chosen transparency axis, highlighted in red. This axis is in line with the vertical bar at the back of the object. (*C*) Transparency applied to the foreground horizontal bar, radiating outwards with the axis at the center of transparency. The section of the object where the transparency is manipulated is highlighted in red. (*D*) Transparency creates the illusion of an impossible object, making the horizontal bar in the foreground appear as if it travels behind the vertical bar at the back.

involved in our transparency algorithm. Beginning with a three-dimensional possible representation of the desired impossible figure an axis must be supplied, around which transparency can be applied to the object's surface. This transparency axis generally follows the orientation of one section of the object, in the example object the desire is to make the horizontal bar at the forefront of the object appear to be behind the vertical bar at the back of the object. As transparency is manipulated based on the distance from the supplied axis, radiating away with the axis at the center of transparency, the orientation of this supplied axis matches the orientation of the vertical bar at the back of the object. Surface transparency for the foreground horizontal bar is then calculated around the supplied axis.

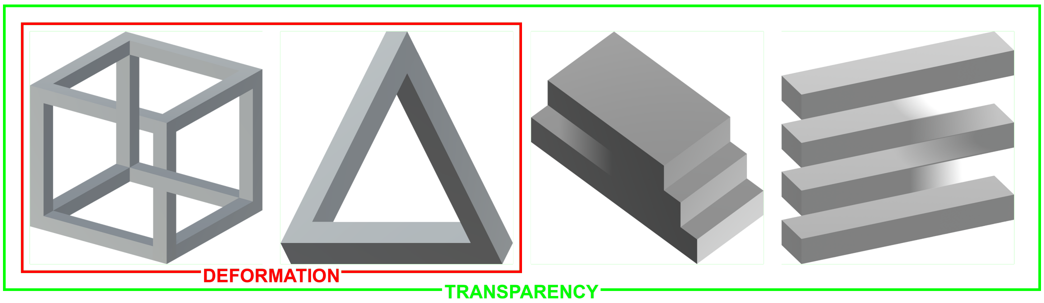


Figure 3.2: Examples of the four main classes of impossibility. The Impossible Cube, The Penrose Triangle, The Impossible Staircase and The Impossible Shelves. Those able to be modeled using deformation-based techniques [15] are outlined in red. Using our transparency method it is possible to simulate all four types of impossibility, outlined in green.

Our transparency modeling method is similar in its function to the line of sight deformation method proposed by Elber [15]. Both cases perform similarly, where the effect is applied radiating outwards around a user-specified line/axis. The strength of the effect at a point on the surface of the object is determined by its distance from this user-specified axis. The ability to manipulate such factors as the region of influence and rate of decay of transparency can be done by changing values in our modeling formula. Both effects are also able to cover a similar range in view directions. The main advantage of using our transparency method over the deformation based method is the ability to simulate a wider range of impossibilities. Figure 3.2 shows the four key types of impossibility, depth interposition, depth contradiction, disappearing normals, and disappearing space. Using Elber's deformation based modeling method it is only possible to simulate the first two types of impossibility, those created through depth illusions. Using our transparency based method it is possible to simulate all four types of illusion. By varying the rate of change of transparency we are able to achieve the instant transition from transparent to opaque needed for depth-based impossibilities, and achieve the linear change in transparency needed for the disappearing sections based impossibilities.

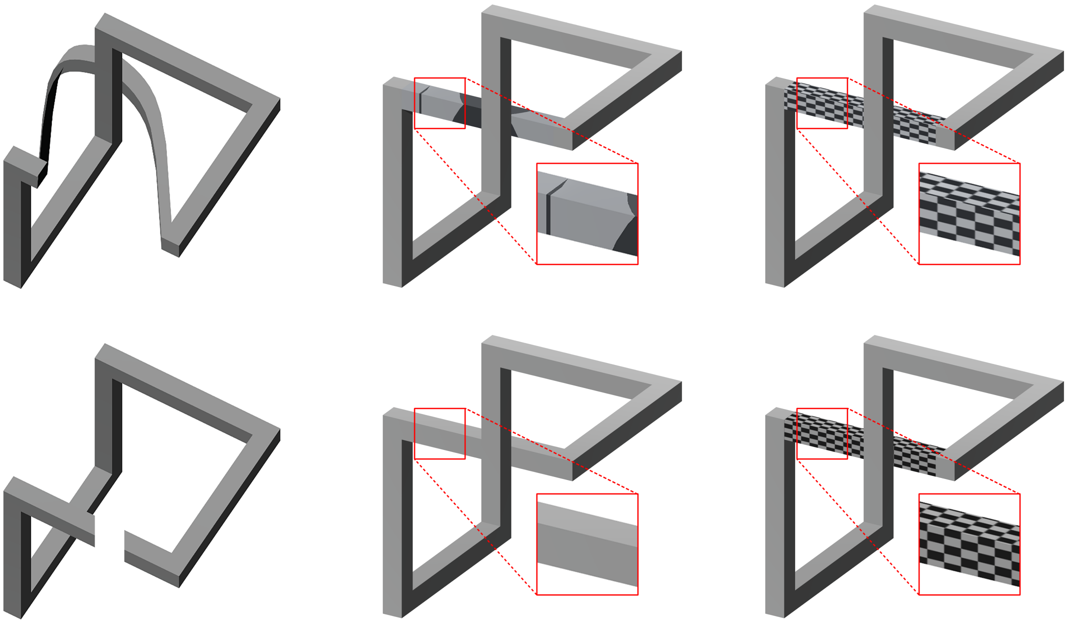


Figure 3.3: (*top*) An object modeled using deformation-based methods. Discontinuities in the shading and texturing can appear along the deformed section of the object. (*bottom*) The same object modeled using our transparency method. The transparency is applied to the same section of the object without affecting the shading and texturing. (*left*) Shows both objects from an alternate viewpoint, here we see why the deformation based model has inconsistencies in its shading and texturing.

One of the main novelties of our transparency method is being able to simulate these impossibilities whilst leaving the original geometry of the object unchanged. Deformation-based methods [15, 84] apply hidden manipulations to an object's structure along the viewing direction, that can only be seen when the object is viewed from an alternate viewpoint. These deformations change the three-dimensional structure of the object, whereby it loses its orthogonality. Although the deformations are not visibly evident in the shape of the object, the change in shape reveals itself through inconsistencies in the object's surface shading and texturing. To correct these inconsistencies additional steps must be taken that involve sampling or interpolating the surface normals of the object prior to applying the deformation [79]. This additional step means either an additional undeformed version of the object must be stored or additional calculations must be performed to compute the original surface normal. Both solutions result in more memory or calculations that our transparency based method aims to avoid. The benefit of using transparency to simulate impossibility is that the geometric structure of the object remains unchanged, meaning no additional operations are needed to produce accurate surface shading or texturing. Where the object structure is manipulated, to handle novel viewpoints (viewpoints that differ from the objects initial 'illusion viewpoint'), this can be done through scaling the object rather than through deformations/distortions. Figure 3.3 shows how applying a deformation to an object structure causes problems with the conventional methods of shading and texturing, whereas our transparency method works without additional changes.

3.2 Transparency Based Impossibility Algorithm

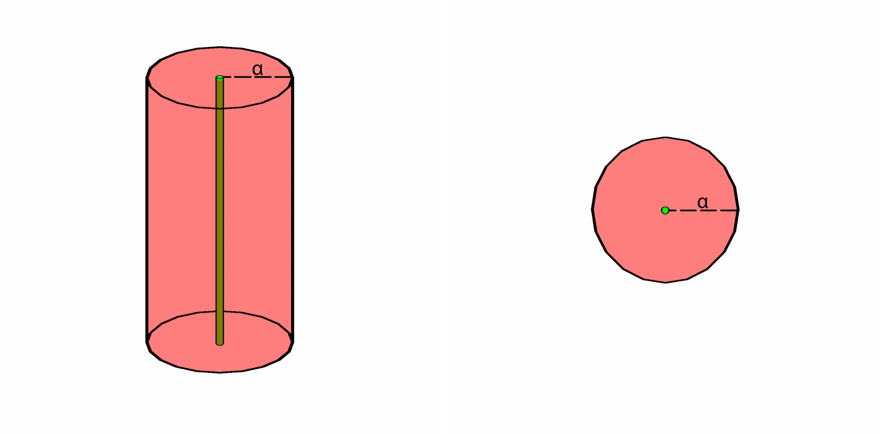


Figure 3.4: Visual demonstration of how transparency is calculated. Transparency axis (*green*) is at the center of transparency (*red*). Transparency effect radiates away from axis up to a set distance threshold.

In order to simulate the illusion of impossibility using transparency, we must begin with a regular three-dimensional model, one that represents the general shape of the impossible figure we wish to recreate. Our algorithm is then used to manipulate the surface transparency for specific sections of the object that create the figures impossible nature. Transparency is calculated based on the distance from a user-supplied axis, radiating outwards with the axis located at the center of transparency. Figure 3.4 demonstrates how transparency is applied around a specified axis.

Our algorithm provides a method of calculating new transparency values for each point $P(x, y, z)$ on the surface of an object, based on its distance D from a user-specified axis. Given the center of transparency and an axis along which to apply the transparency, it is possible to calculate a corresponding line in the XY plane. Using Eq. 3.1 we can calculate the distance of any point on the object surface (x, y) to the line, where A, B, C are the coefficients of the line in the XY plane.

$$D = \frac{|A(x) + B(y) + C|}{\sqrt{A^2 + B^2}} \tag{3.1}$$

This will alter the transparency along the entire object surface, to limit this we supply a region of influence R to set the distance at which the transparency will no longer take effect. Along with a rate of change modifier k between 0 and 1 to alter the distance at which the change in transparency begins to take effect. By using these threshold values we can normalize the distance D to calculate the transparency value (α) at the current point using Eq. 3.2. Where $R_{min} = R * k$. Clamping values in the range of 0 to 1.

$$\alpha = \frac{D * R_{min}}{R - R_{min}} \tag{3.2}$$

Using the rate of change modifier k it is possible to change where the linear transition of transparency begins. Along with being able to achieve an instant transition between fully transparent and opaque, when $k = 1$. Figure 3.5 demonstrates how manipulating the rate of change modifier k can affect the resulting transparency of an object. This freedom to manipulate the rate of change of transparency enables the ability to simulate a wide range of impossible illusions. The impossible cube illusion, Figure 3.2, requires sections of the object appearing to be simultaneously in front and behind another section. By using a sharp transition from transparent to opaque we can achieve this and maintain the illusion that every section of the object is a solid continuous surface. Whereas the impossible staircase, Figure 3.2, requires a more gradual transition, one surface of the object fading into another, using a more gradual linear change in transparency here creates this illusion.

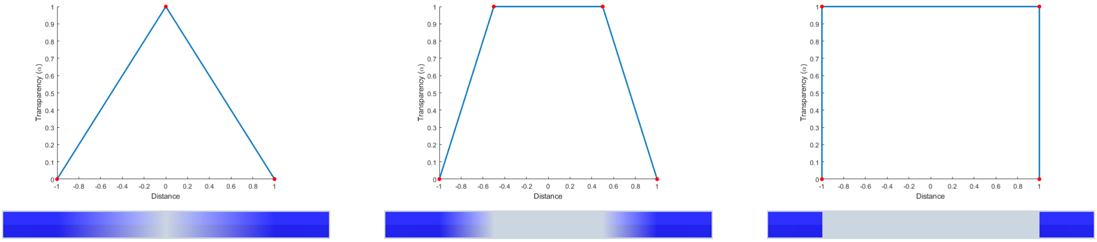


Figure 3.5: Plots of transparency effect (*top*) with corresponding visual effects (*bottom*). (*left*) $k = 0$ creates a linear transition of α from center of transparency ($distance = 0$) to maximum range of influence (R). (*Center*) $k = 0.5$, α remains at max until distance value becomes greater than k. Increasing the area at which the object is fully transparent. (*Right*) $k = 1$, creates an instant transition from fully transparent to opaque. Giving the effect of a clean cut through the object.

3.3 Novel Viewpoints

The ability to work under a range of novel viewpoints is an important factor to consider when modeling impossible objects. Impossible figures are often only seen from a single static viewpoint or 'illusion viewpoint', when creating a three-dimensional representation of these figures it is often the case that when viewed from an alternate viewpoint the impossibility illusion is broken. To allow our transparency based impossible models to be viewed from novel viewpoints we divide them into two categories of illusions. Those created by depth ambiguities, such as the impossible cube and the Penrose triangle. The second category being those created through disappearing sections, such as the impossible staircase and the impossible shelves.

3.3.1 Depth Based Impossibility

For those impossible objects created through depth interposition or contradiction, our transparency based modeling method behaves similarly to the line of sight deformation method proposed by Elber [15]. When supplied with a single axis of transparency our modeling system can handle a limited range of viewpoints without any additional changes, see examples (A) and (B) in figure 3.6. For the impossibility to be successful within this example the supplied axis must be parallel to the vertical/horizontal section at the back of

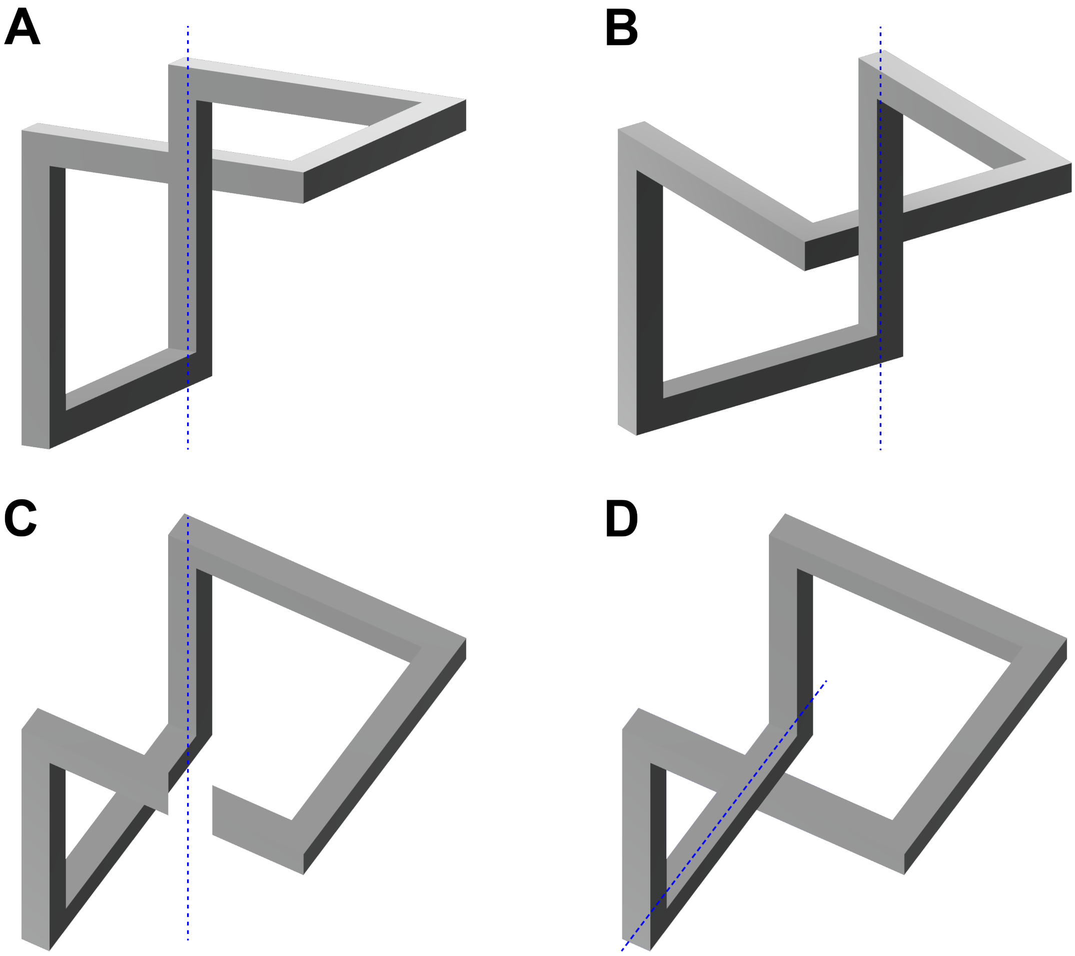


Figure 3.6: Changing axis of transparency for novel viewpoints. Images (A) and (B) show how a single transparency axis (highlighted by the dotted line) can be used for a range of viewpoints. Image (C) shows an example where the current transparency axis fails. By changing the axis of transparency an impossible object can still be achieved at this viewpoint as shown in (D).

the object, the transparency of the foreground section is then altered to give the illusion it passes behind the rest of the object. Once the viewpoint moves outside the acceptable range the currently specified axis of transparency fails, as in example (C) from Figure 3.6. At this point, a new axis must be specified, as in example (D) from Figure 3.6. This is the same behaviour we see with the line of sight deformation method. A single deformation value can work under a limited range of viewpoints, outside of which the deformation algorithm must be updated.

3.3.2 Disappearing Sections Based Impossibility

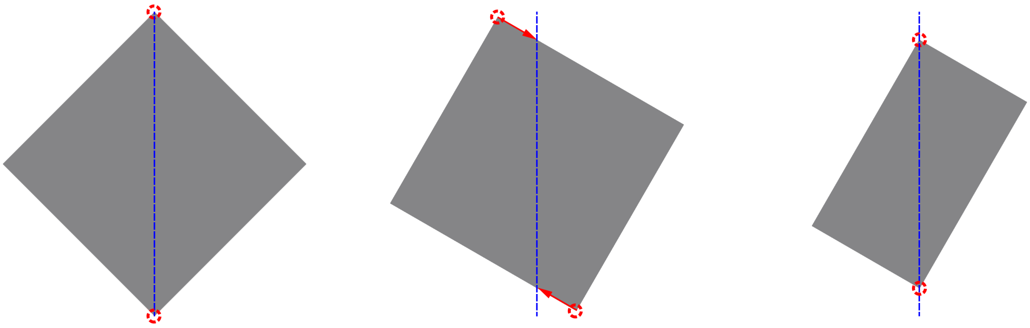


Figure 3.7: Calculate scaling to correct misalignment of edges. (*left*) Initial position, corners highlighted in red are selected as anchor points. (*center*) The object is rotated, such that the anchor points are no longer aligned. Distance from the anchor points to the original alignment line is calculated. (*right*) All object vectors are offset by the calculated distance to realign the anchor points.

For those impossible objects created through disappearing space or disappearing normals, a different problem must be overcome when viewed at novel angles. These objects rely on the correct alignment of the object edges to create the illusion of impossibility. Where one section of the object seems to fade away into another. As the viewpoint is moved away from the initial 'illusion viewpoint' this alignment is broken, as is the impossibility illusion.

By using a scaling based method our algorithm is able to compensate for this misalignment of edges under novel viewpoints. Figure 3.7 shows a basic example of how an object can be scaled to maintain alignment of the important points. Two points are chosen as anchor points, from which the line connecting them can be calculated. This will be the alignment line, the anchor points will be scaled such that they remain in position on this line. As the viewpoint is changed the distance between anchor points and alignment line (D) can be calculated. This operation is done in world space such that as the object is scaled, it maintains its orthogonal structure. The new position for each of the object vertices can then be calculated using $P = P + (V * D)$. Where P is the current vertex position and V is the vector along which to scale. Figure 3.8 shows how this operation can be applied to an impossible object, maintaining the illusion of impossibility at novel viewpoints. This use of scaling allows us to solve for novel viewpoints of these impossible objects, whilst maintaining the orthogonality of the original object.

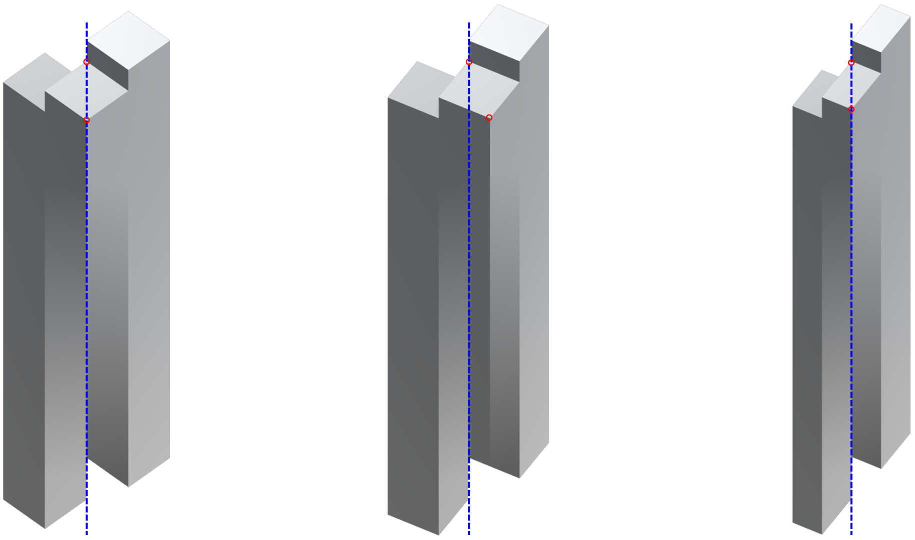


Figure 3.8: Scaling to solve novel viewpoints for a transparency based impossible object. The impossibility of the object is only achieved when the front and back corner are in alignment, along the dotted line. When the object is viewed from a different angle, the alignment is lost. To compensate for this the object is scaled until the two points are back in alignment.

3.4 View Range

To measure the performance of our transparency based modeling method we examined the range of viewpoints at which we could successfully model an impossible object. We sampled the hemisphere above the objects, 360 degrees around the azimuth and 90 degrees elevation, sampling a total of 32,400 view directions. This range was chosen as we imagine the object is standing on a flat surface and cannot be viewed from below. This will be the same range used to test our copycat cast shadow algorithm. Figure 3.9

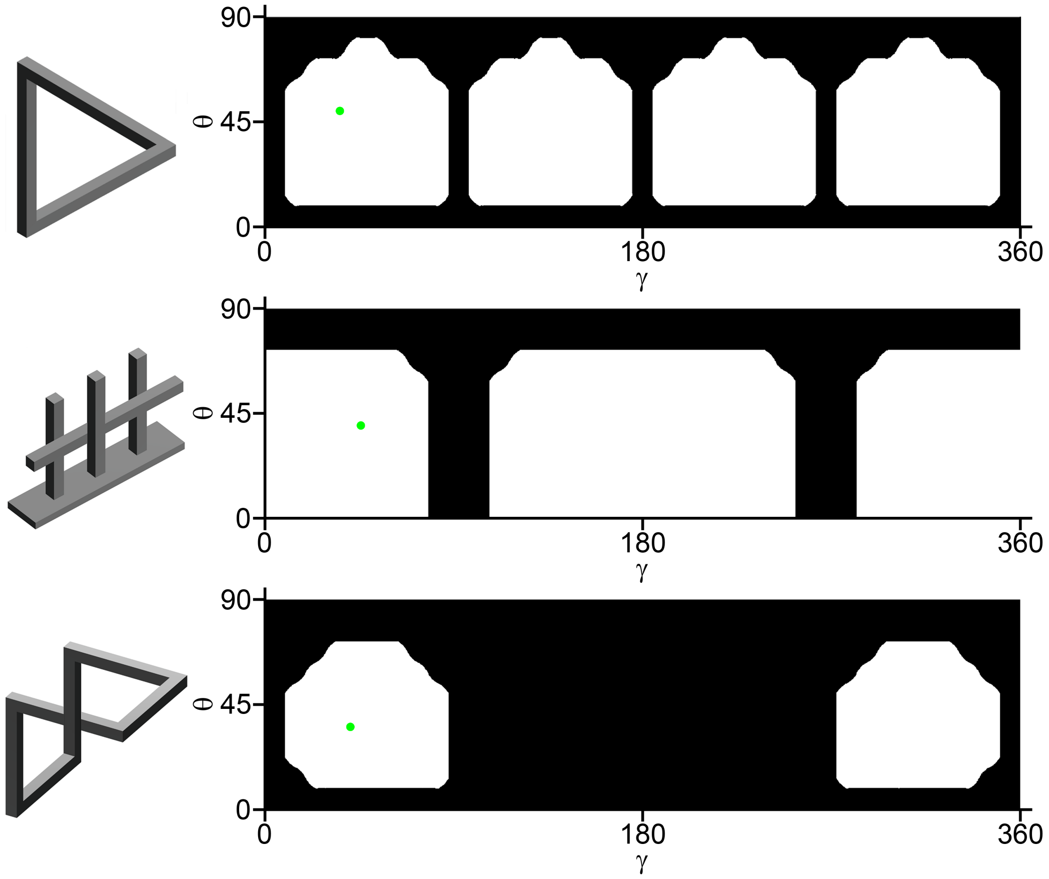


Figure 3.9: Success rate of our transparency algorithm for a range of view directions. γ and θ indicate the azimuth and altitude (in degrees) of the view direction, this range covers the hemisphere directly above the object. At each viewpoint, we examine whether our algorithm successfully models an impossible object (white) or fails to model an impossible object (black). The green point on each graph indicates the position of the objects original ’illusion viewpoint’ shown on the left.

shows the results for three impossible objects. The graphs presented show the entire range of sampled viewpoints, where white indicates our modeling method was successful, and black indicates viewpoints at which our modeling method failed.

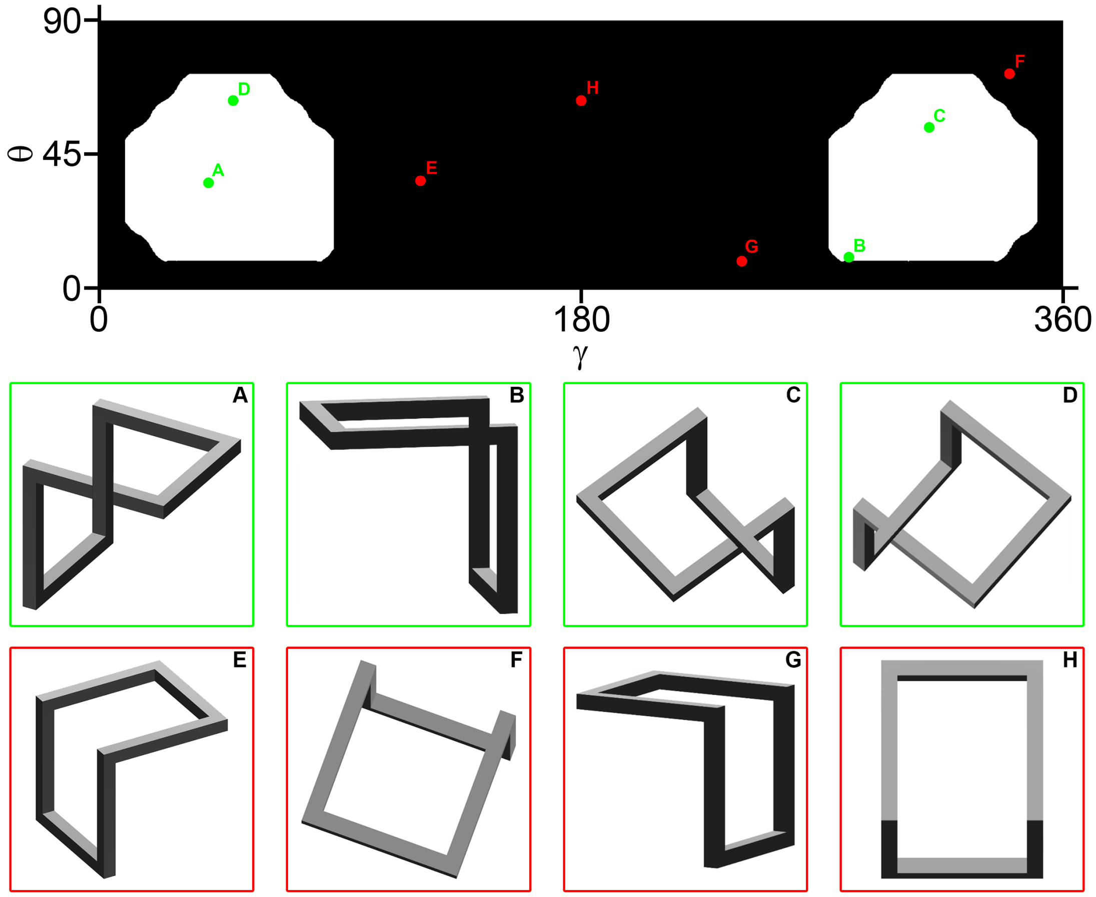


Figure 3.10: Examples of successful and failure cases for our transparency based modeling algorithm. Green points on the graph correspond to the four successful cases, whilst red points correspond to the four failure cases. The successful cases all share the feature that sections of the object overlap one another, this feature is needed to successfully simulate the impossible illusion. This suggests that view limitations are influenced by the structure of the object itself, making some objects more difficult to model than others.

At each viewpoint, we examined the structure of the object to determine if our modeling method had successfully achieved an impossible object. This was done by manually choosing sections of the object which must overlap for our modeling method to be able to simulate impossibility. Figure 3.10 shows an example of this. Each of the viewpoints highlighted in green contains an overlap in the object structure and those highlighted in red contain no overlap. For these objects, the basis of their impossibility relies on overlapping sections of the object, where one section appears to be both in front and behind another at the same time. If this overlap is not present in a viewpoint our transparency modeling method cannot resolve this issue.

From the graphs shown in Figure 3.9 we can see that the successful view ranges vary depending on the object. With the successful range of viewpoints varying from around 65% to 27%. This result seems to support the theory Wu et al. [84] proposes, that the view range may not only be limited by the chosen modeling method but also the structure of the object itself. Making some objects more difficult to model than others. The three objects were chosen due to their presence in the paper by Elber [15], each of these objects can also be modeled using his deformation based method. Both Elber's deformation based method and our transparency based method operate based on a transfer function, as such this deformation based method is the closest comparison to our own method. We performed the same view range tests for the objects modeled using Elber's deformation method and achieved the same results, indicating that our modeling method is able to successfully perform over the same view ranges and that both methods are affected by the base structure of the object itself.

We have demonstrated that our modeling method can successfully model and render impossible figures at viewpoints far away from the initial 'Illusion viewpoint' and that at those viewpoints where our modeling method fails it does not distort or deform the resulting object. In these cases our method merely renders the object without the impossible illusion, in other words, it resorts to rendering the possible version of the object. If this is not acceptable we may use the plots generated in Figure 3.9 to restrict the viewing positions for each object, for example using simple thresholding, to those viewpoints where our modeling method is successful.

3.5 Results

Using our transparency based impossibility modeling we are able to simulate all four categories of impossibility illusions. Figure 3.11 demonstrates the use of our transparency based method to model each of the four main

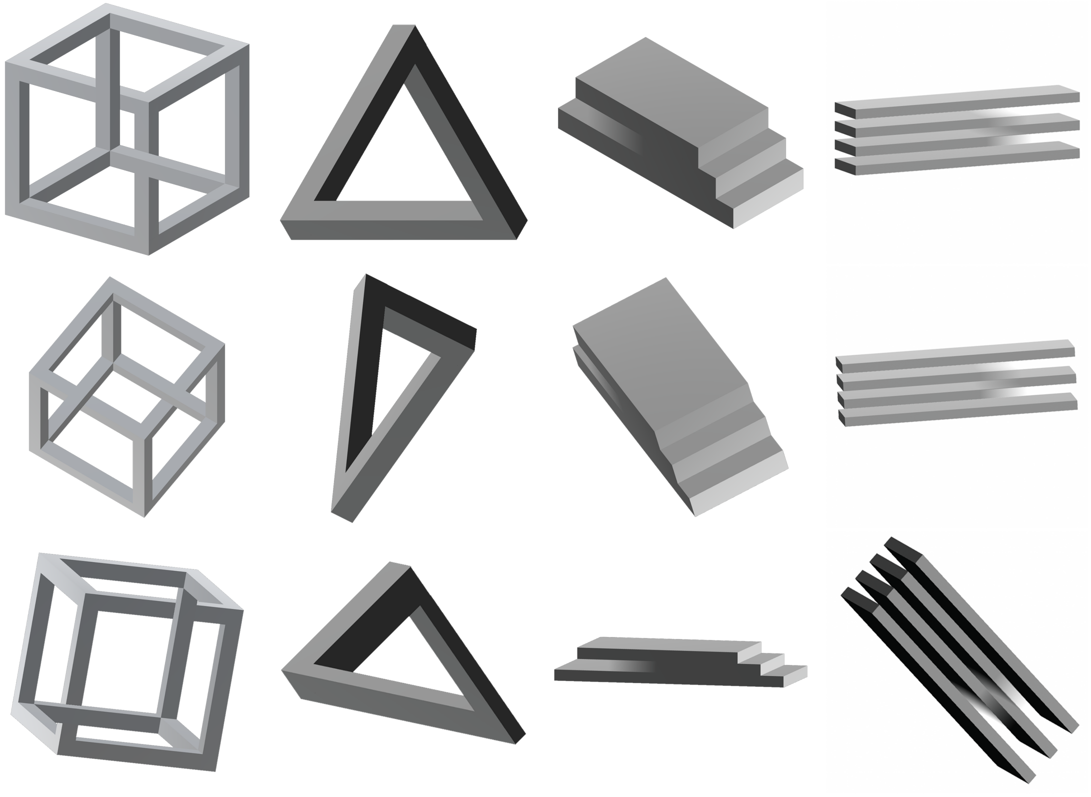


Figure 3.11: Four types of impossible object, each modeled using transparency and viewed from novel view directions. (*From left - right*) Impossible Cube - Depth Interposition based impossibility. Penrose Triangle - Depth Contradiction based impossibility. Impossible Stairs - Dissapearing Normal based impossibility. Impossible Shelves - Dissapearing space based impossibility.

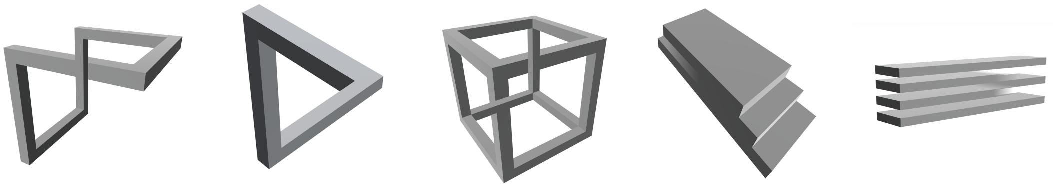


Figure 3.12: Impossible objects modeled using our transparency based algorithm viewed under perspective projection.

types of impossibility illusion, depth interposition, depth contradiction, disappearing normals, and disappearing space. Each of these impossibilities is recreated using our algorithm to alter the surface transparency for sections of the object, the actual geometric structure of the object remains unchanged. When viewed at novel viewpoints away from their initial 'illusion viewpoint' the objects are scaled along the desired axis to maintain the alignment of edges and their illusion of impossibility.

As our transparency method does not alter the geometry of the three-dimensional model it is useable for both orthographic and perspective impossible models, as shown in Figure 3.12. Figure 3.13 demonstrates the construction of Escher's "Waterfall" using our transparency modeling method. This structure belongs to the depth contradiction class of impossible objects, being made up of three Penrose triangle objects connected by multiple pillars. Figure 3.13 demonstrates the segmentation, rendering, and novel views of the object. Transparency is applied to each of the colored sections that create the illusion, with each section's axis of transparency being depicted as a dotted line in the corresponding color. Textures have been applied to each section of the object in the final rendered image. The third image in Figure 3.13 shows the object from a novel viewpoint, a different view from its initial 'illusion viewpoint'. This is achieved using our scaling method, notice that while the object may appear flatter at this view, it is not skewed and the orthogonality of the structure is maintained.

3.6 Implementation

An important goal of our research was to enable our modeling method to work within already established modeling and rendering environments.

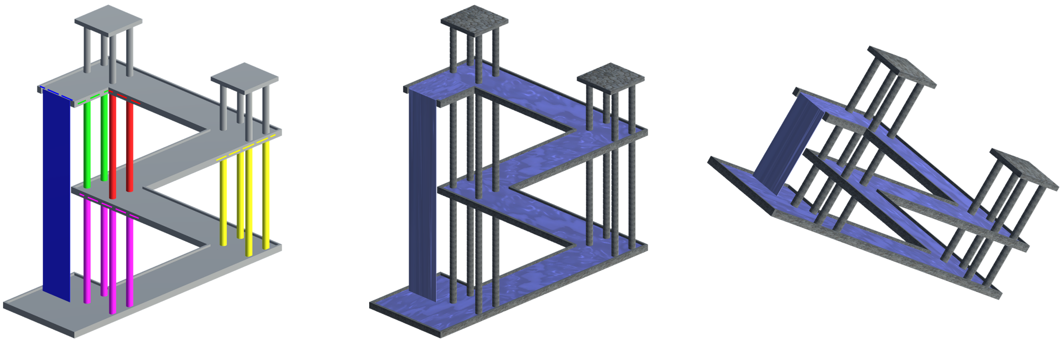
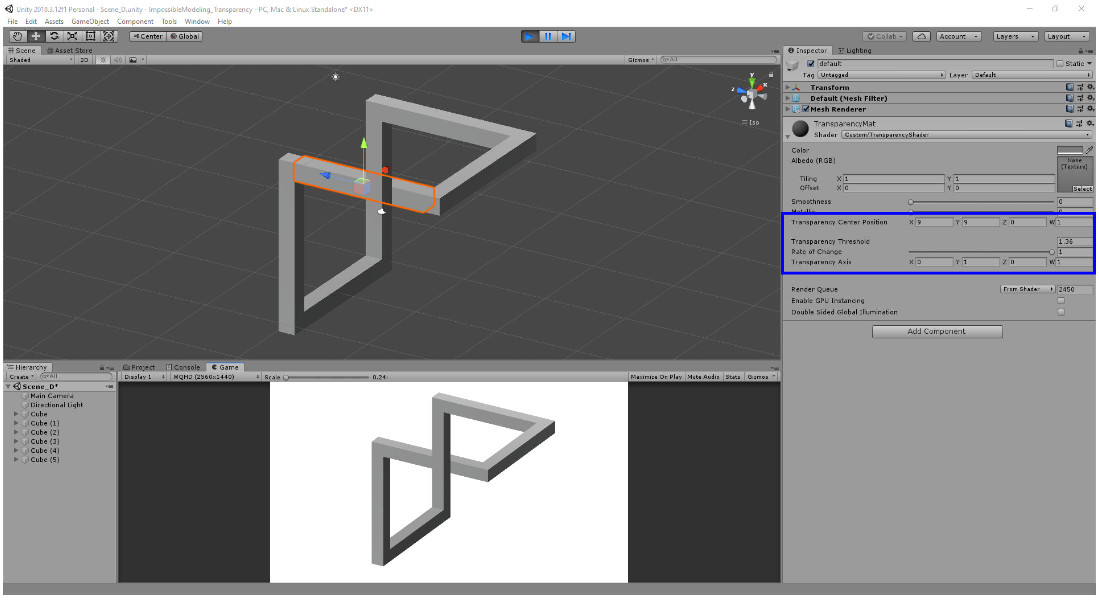


Figure 3.13: A three dimensional representation of "*Waterfall*" by M. C. Escher [18] created using our transparency based modeling method. (*left*) Markup, transparency is applied to each of the colored sections with each sections transparency axis indicated by a dotted line in the corresponding colour. (*middle*) Textured version of the impossible object. (*right*) Novel view of the object, scaling is used to preserve the impossibility illusion and the objects orthogonality.



Unity 2018.3.12f1 Personal - Scene_D.unity - ImpossibleModeling_Transparency - PC, Mac & Linux Standalone* <DX11>

Scene_D*
- Main Camera
- Directional Light
- Cube
- Cube (1)
- Cube (2)
- Cube (3)
- Cube (4)
- Cube (5)

Display 1 | WQHD (2560x1440) | Scale 0.24

☑ default □ Static
Tag Untagged Layer Default

Transform
Default (Mesh Filter)
☑ Mesh Renderer

TransparencyMat
Shader Custom/TransparencyShader

Color
Albedo (RGB) None (Texture)
Tiling X 1 Y 1
Offset X 0 Y 0
Smoothness 0
Transparency Center Position X 9 Y 9 Z 0 W 1
Transparency Threshold 1.36
Rate of Change 1
Transparency Axis X 0 Y 1 Z 0 W 1
Render Queue From Shader 2450
Enable GPU Instancing □
Double Sided Global Illumination □

Figure 3.14: Demonstration of our transparency algorithm using the Unity Engine. Our algorithm is applied to the object as a custom shader, allowing users to make use of Unity's existing features without the need for a dedicated rendering environment.

As was discovered in our background research all of the existing modeling methods we discussed, with the exception of the work by Chiba et al. [11], require their own dedicated rendering environment. By creating our modeling method to work within more general rendering environments we hope to make the process of modeling and rendering these impossible objects easier for the user, opening impossible objects up to more mainstream uses.

We implemented our transparency based modeling method within the Unity Engine, similar to Chiba et al. [11]. Figure 3.14 shows our modeling method working within Unity along with highlighting the required user input. Constructing the basic object shape is done using Unity's existing methods and geometric shapes. The goal of the user here would be to construct the shape of their desired impossible object without the impossible illusion, in essence, they are modeling a possible version of the desired impossible object.

Our transparency based effect is applied through a custom transparency shader attached to the material of the object. This allows the user to maintain the use of material colors and textures as they would with the base Unity material, whilst also applying our transparency function. The current user input needed is highlighted on the right of Figure 3.14. Our algorithm requires the position at the center of transparency and an axis along which to apply the transparency effect to be supplied. They are also required to input the distance threshold and rate of change of transparency. These inputs are used to solve equation 3.1 within the custom shader. This shader then changes the surface transparency of the object based on distance from the center of transparency. Using the rate of change slider the user can achieve both a smooth linear transition in transparency and an instant transition between transparent and opaque.

By implementing our transparency method as a custom shader within Unity it can be more easily utilized by a wider range of users, compared to if it were using its own dedicated environment. Users also have access to the built-in modeling and rendering tools already available within the engine. Although we have currently only implemented our modeling method using the Unity engine, we predict that it is possible to implement using other mainstream modeling and rendering environments.

3.7 Discussion

We have described a transparency based modeling solution for creating three-dimensional models of impossible figures. The main advantage of this method over existing deformation methods its ability to recreate all four types of impossible objects, without manipulating the underlying geometry of the object. By maintaining the original geometric structure we do not have to compensate for anomalies in the shading and texturing of the objects that are caused by deformation based modeling methods. We have demonstrated the ability of this transparency based modeling method to recreate a range of impossible objects, both simple and complex, in orthogonal and perspective projection. We have also created our algorithm to be used within general rendering environments, making it more accessible and mainstream for users without the need for any dedicated rendering environments.

One of the main limitations to the current transparency modeling algorithm presented is its ability to simulate an intersection within an object structure, shown in Figure 3.15. The illusion within this impossible figure is created through one section of the object that appears to intersect with another section. This cannot currently be achieved using our transparency algorithm. Figure 3.15 shows an example of an impossible illusion created through object intersection. Using our transparency based method the object still contains impossibility based on depth interposition, however, we are not currently able to simulate the intersection of the object sections. Future work should be done to expand our transparency method to allow for such features as object intersection. In our current implementation, a user must supply a single axis around which transparency is calculated. Extending this with additional axes may allow for the simulation of an intersection, see figure 3.15. Supplying additional axes that only allow for transparency to be applied when the sample point is above/below.

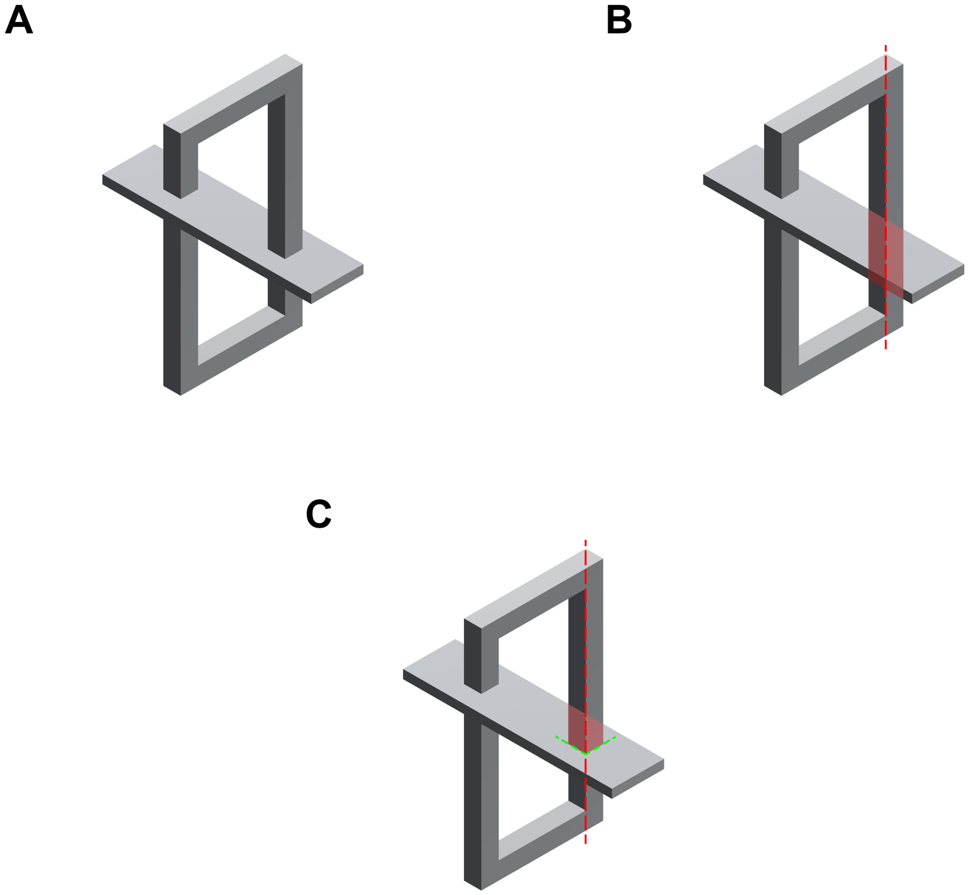


Figure 3.15: Impossibility illusion created through object intersections. (*A*) Object modeled using line of sight deformation. (*B*) Object modeled through transparency. Since transparency is applied along the entire specified axis it is not possible to create the appearance of an intersection. (*C*) Example of how multiple axes may be used to create the illusion of intersection. Original transparency axis highlighted in red additional axis highlighted in green.

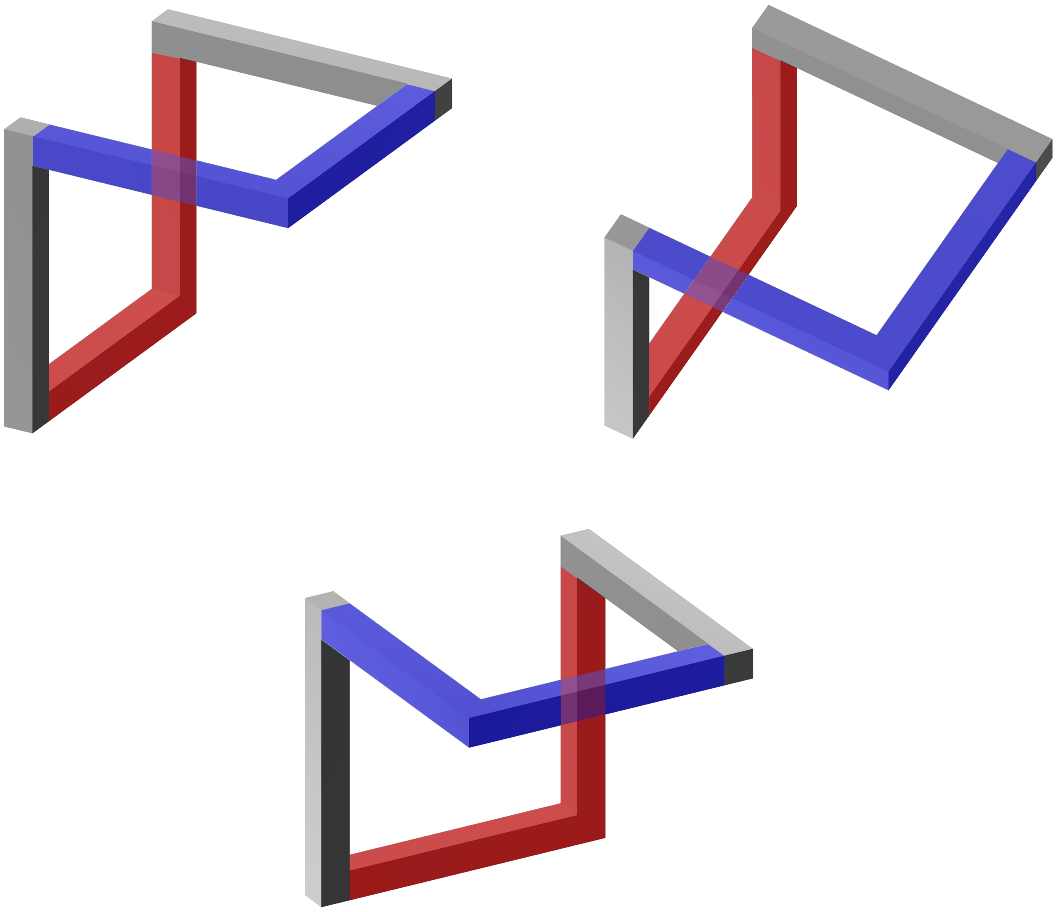


Figure 3.16: Transparency based on object overlap. Two sections of the object are selected during the modeling phase (*red and blue*). Transparency is then applied to any point on the surface of the object where the chosen sections overlap.

When examining the range of viewpoints under which our algorithm is successful we see that it is affected by the structure of the object, in particular, whether specific sections of the object overlap. The current method of applying transparency to the surface of an object operates using a transfer function where the strength of transparency radiates away from the user-specified axis, along with a distance value and rate of change modifier. A possible change could be made to apply the transparency to areas where chosen sections of the object overlap. This may make our algorithm require minimal user input. During the construction of the object the user could label two sections, one blue and one red, an example is shown in Figure 3.16. For each pixel in the image, a ray can be used to test whether both sections of the object are intersected. If so the transparency of the section closest to the viewer is altered. Future work should be done to look at whether this overlap method may work for different types of object impossibility, such as those created by disappearing space and normals.

Chapter 4

Impossible Object Rendering

This chapter introduces a new screen space occlusion algorithm to produce copycat shadows for three-dimensional impossible models. By projecting the model onto a planar surface in screen space our algorithm ignores the hidden geometrical complexities within the object model. Resulting in an impossible shadow whose shape is comparable to the visible shape of its casting object, a copycat shadow. Although they are technically impossible, results show that copycat shadows often appear much more convincing than the object's traditional cast shadow. Through the use of these copycat shadows, we demonstrate how it is possible to fully render these seemingly impossible models with shadows and shading without detracting from the illusion of impossibility. We implement our algorithm using the Unity Engine, avoiding the need for any dedicated rendering environment.

4.1 Introduction

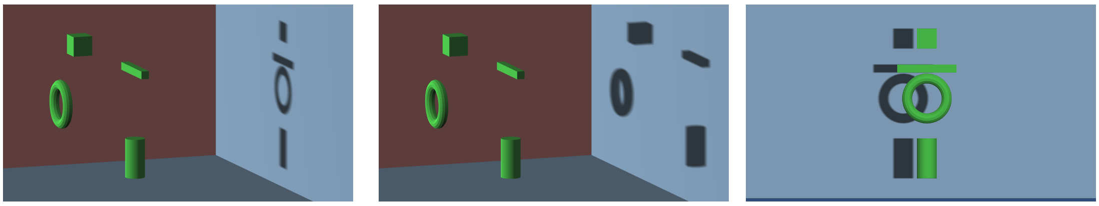


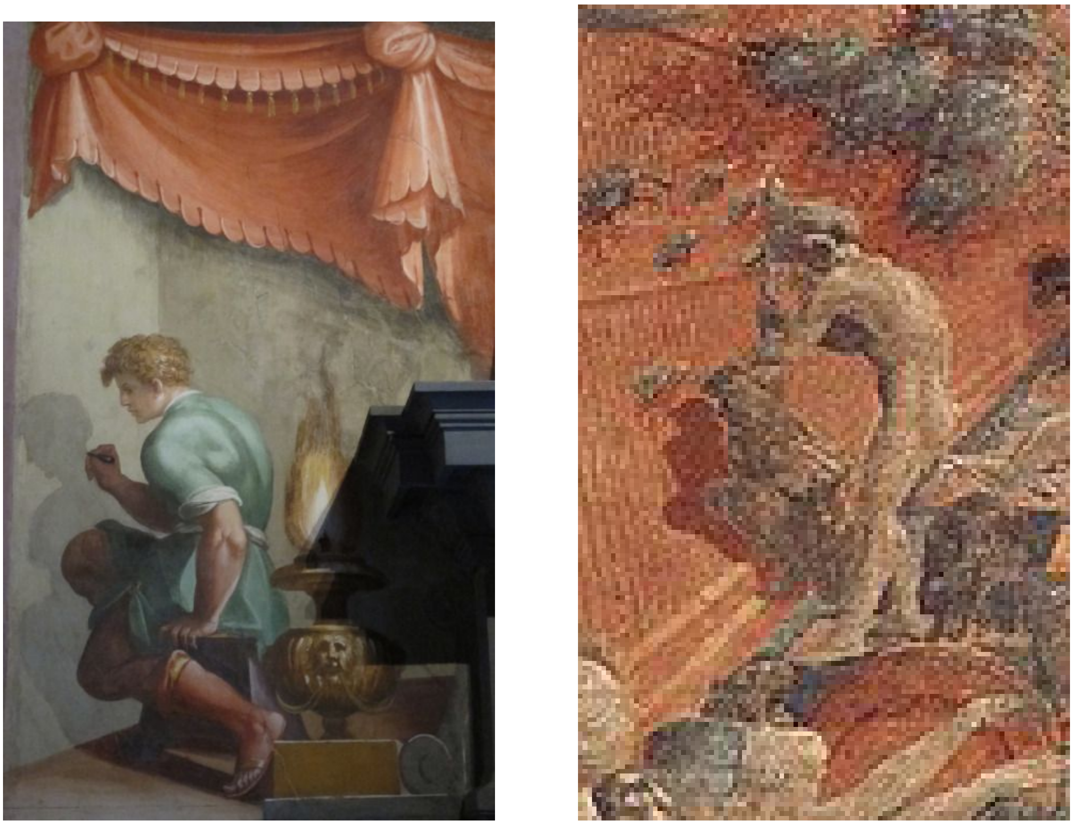
Figure 4.1: Scene reproduced from Casati [8] containing four objects alongside their normal cast shadows (*left*) and their copycat cast shadows (*middle*). These copycat cast shadows aim to replicate the visible shape and position of the objects. (*right*) Shows the scene when viewed from the front, revealing the true positioning of the objects.

Previous computer graphics research into the process of modeling three-dimensional impossible figures [84, 15, 61] has focused on challenges such as rendering the objects from new and novel viewpoints, where traditionally we see impossible figures as static two-dimensional images viewed from a single viewpoint. When we see these three-dimensional impossible models rendered using existing methods they are shown in very basic images. Most commonly we see them alone without any other objects in the scene, rendered using basic shading and lacking any form of cast shadow. This is attributed to the hidden deformations and manipulations needed to create these seemingly impossible models (SIMs) affecting the resulting shadows and shading of the object. To overcome problems of inconsistent and anomalous shading that is caused by the deformations used to model SIMs, existing methods [79] resort to sampling or interpolating new surface normals for the object to compensate for the deformations applied to the object geometry. A solution for generating cast shadows for these SIMs has not been explored in previous research. Cast shadows are excluded completely from renderings of these SIMs. Where they are addressed in previous work they are purposefully hidden [79]. This is due to the hidden deformations used to simulate the impossible nature of the object causing corresponding deformations in the objects cast shadow when rendered using traditional lighting methods.

Cast shadows add realism to a computer-generated scene, without them the scene can look flat and two dimensional. Cast shadows portray meaningful spatial and contextual information and are used by our visual system to resolve three-dimensional ambiguities within the two-dimensional image, such as object shape and position. They also play a large role in the recovery of an object's three-dimensional shape and structure along with object recognition. This poses a problem when using the objects traditional cast shadow of a SIM as the hidden deformations within the SIM are revealed, resulting in deformed and oddly shaped cast shadows that do not correspond to the apparent orthogonality of the casting object. However completely excluding cast shadows from the scene withholds important visual cues from the viewer regarding the composition of the scene.

We propose a solution to this problem of generating visually appropriate cast shadows for these SIMs without revealing their true structure by taking advantage of a technique used by artists knows as copycat shadows [8]. The purpose of a copycat shadow is not to represent the true three-dimensional structure of the casting object, but rather to produce a shadow whose shape aims to replicate the visible profile of the casting object. Figure 4.1 shows a set of four objects replicated from the work by Casati [8] demonstrating the objects traditional cast shadows alongside their copycat cast shadows. Comparing the two shadow versions we can see that the shape of the copycat cast shadow is much more similar to the visible shape and position of the objects within the scene. It is suggested that although humans use shadows as a method to retrieve important structural and positional information some inaccuracy within cast shadows is not only tolerable but can be preferable in cases [7, 39]. Although copycat shadows are impossible shadows that would not be found in the real world, only under very limited light and object positions, they are often a simplification compared to the object's normal cast shadow. This simplification along with the similarity between the shape of the shadow and its casting object means when viewing these copycat shadow viewers often find it easier to firstly, correctly identify the dark area as a shadow, and secondly, correctly match the shadow to its casting object [8]. This visual perception problem of associating a shadow with its casting object, known as the shadow correspondence problem, can be difficult within a complex scene or where the object and shadow are far apart. Solving this problem correctly is important to gain an accurate understanding of the scene, in particular determining the location and position of objects within the scene. Copycat shadows may make this task easier due to the natural similarity between the shape of the object and the shape of the copycat shadow. Mamassian [49] suggests that the visual system uses a coarse representation of the scene to solve this correspondence problem. Relying on a wider view of the shapes within a scene rather than the finer details, supporting the case for copycat shadows. When faced with objects casting impossible shadows it was found that participants did not notice the impossibility and were able to effectively solve the correspondence problem.

Artists have taken advantage of this tolerance to enhance their work rather than adhering to what may be considered physically correct. By manipulating such aspects as shadows and shading artists are able to maintain a sense of space and light within their painting without sacrificing their artistic intent [10]. Artists have demonstrated the ability to utilize light and shading to highlight focal points in their scene, producing lighting inconsistencies that although impossible are not evident or jarring to the viewer. Casati [8] examined works from a range of different painters throughout history and found cases where a copycat strategy has been used to draw the cast shadows for specific objects within the scene. Casati highlights the paintings seen in Figure 4.2, where the shadows cast on the wall by the characters in the scene do not seem to resemble realistic cast shadows. Instead, the outline of the shadows is almost identical to the outline of the characters casting them.



(a) The Origin of Painting, Giorgio Vasari, 1573

(b) A section from Flood and Waters Subsiding, Paolo Uccello, 1447-1448

Figure 4.2: Early examples of copycat style shadows in pictorial art. Each of these shadows seems to have a rather simplified shape, resembling the silhouette of its casting object.

In this chapter, we introduce a new screen space occlusion algorithm that maintains the core concepts of calculating occlusion in screen space but through manipulations of the stored depths allows us to generate copycat shadows. Our occlusion method aims to take advantage of the human visual systems' tolerance to shadow shape and utilize copycat shadows as a method of producing visually appropriate cast shadows for a range of three-dimensional impossible objects. As our occlusion algorithm operates in screen space we are able to perform per-pixel operations on the stored depths and project them onto a planar surface. By doing so our algorithm removes the objects three-dimensional depth information, creating a cast shadow based on the silhouette of the object's visible profile. Enabling us to generate visually appropriate copycat shadows for both possible and impossible objects.

4.2 Casting Copycat Shadows

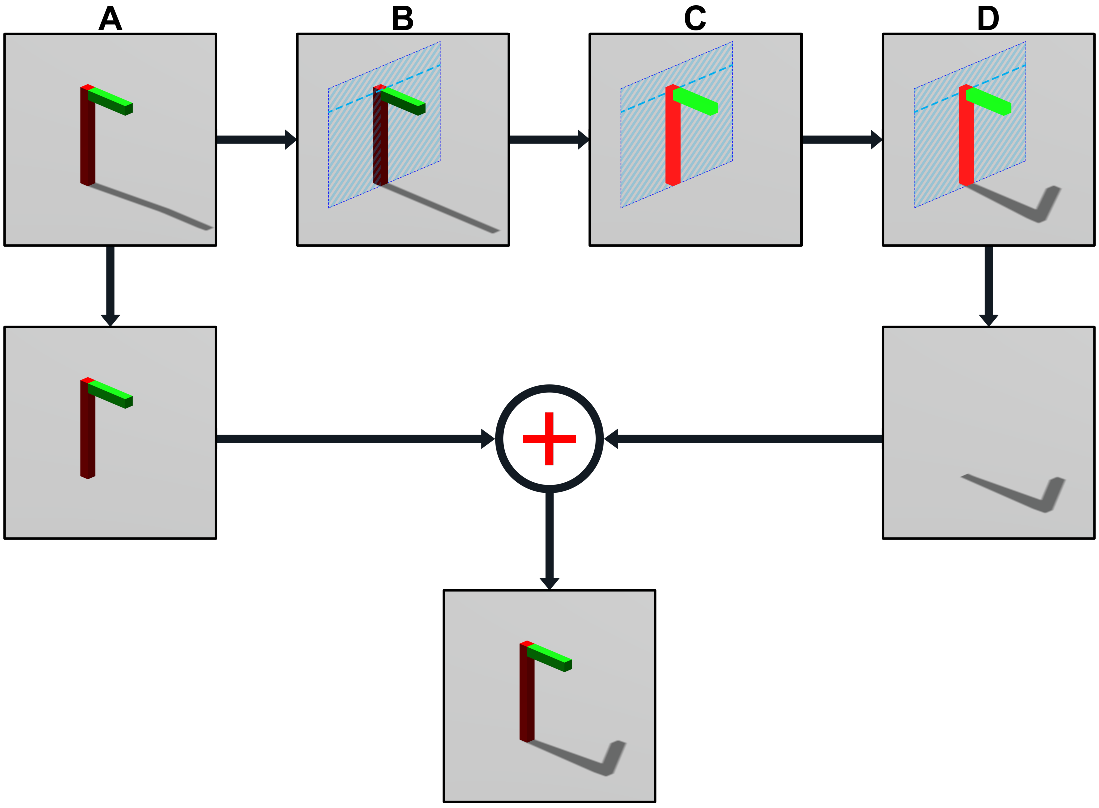


Figure 4.3: Our algorithm creates copycat cast shadows through manipulating stored depth information. Projecting the depth information such that the object is flattened onto the surface of a plane. The copycat shadow of the original object is obtained by using the shadow cast by this flattened version of itself.

Our copycat shadow algorithm works by manipulating stored depths, projecting them onto the surface of a planar object. This process essentially flattens the object, creating a silhouette of the object from which a shadow can be cast.

A visual overview of our algorithm is presented in Figure 4.3:

- (4.3 A) Beginning with an initial three dimensional object casting its traditional shadow.

- (4.3 B) A plane is constructed in three dimensional space. From our research, the positioning of this plane often lies parallel to one surface of the object. This plane will not be rendered in the final image.

- (4.3 C) Our algorithm projects the depth information of the three-dimensional object along the view direction, onto the surface of the chosen plane. This essentially flattens the object without change the visible shape.

- (4.3 D) A cast shadow is then calculated using the objects projected depth information. This resulting shadow is a copycat cast shadow.

To achieve the appearance of a three-dimensional object alongside a copycat cast shadow our algorithm operates in screen space. This means the original geometry of the object is not lost. In the final image, the original three-dimensional object is seen with its cast shadow being generated using the projected depth information.

When using the original depth information for the three-dimensional object we can see that the shape of the cast shadow is noticeably different from the visible shape of the object, see Figure (4.3 A). When rendering the flattened version of the object we lose any shading on the object as we are removing much of the three-dimensional information, however, a copycat cast shadow is produced, see Figure (4.3 D). This copycat shadow appears very similar in shape to the object. By combining these two results, the three-dimensional object and the flattened copycat shadow, it is possible to render an object which maintains correct shading and three-dimensional appearance but also casts a copycat shadow. To achieve this our algorithm calculates new depths for the object in screen space prior to calculating occlusion. Once new depth information has been calculated the process of calculating occlusion can be done using existing screen space occlusion methods. We implement our algorithm within the Unity engine, using the built-in rendering pipeline to calculate occlusion. Since depth information is manipulated on a per-pixel basis it is possible to create a scene using both traditional and copycat cast shadows. By projecting the depths onto a planar surface it is possible to remove the complex three-dimensional information hidden within the object. The resulting cast shadows appear more similar in shape to the visible profile of the casting object rather than its true three-dimensional structure.

4.2.1 Create Projection Surface

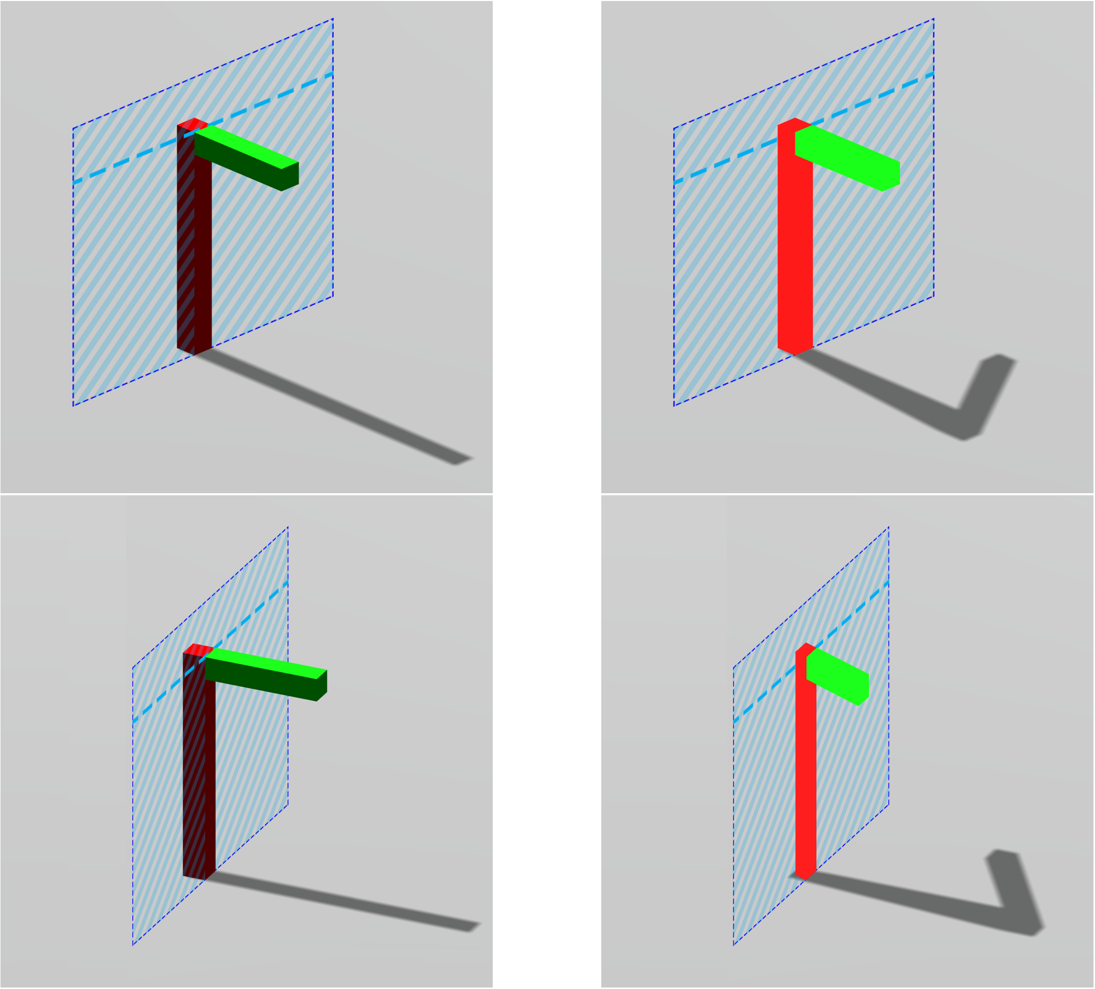


Figure 4.4: Projecting the object onto the planar surface. (*left*) Original three dimensional object, parts of the object are behind while others are in front of the plane in three dimensional space. (*right*) The object has been projected or flattened such that it lies on the surface of the three dimensional plane. Using this flattened object shilouette creates a copycat cast shadow.

Calculating new depth information for objects involves projecting the information stored within the depth buffer onto the surface of a plane. This projection plane must be specified by a point on its surface and its normal vector, from this the surface coefficients are calculated. The surface coefficients will then be used to calculate new depth information at each sampled pixel during the rendering pass. The projection plane itself is not visible within the final rendered image. Figure 4.4 shows an example of the location of this projection plane in relation to the object model.

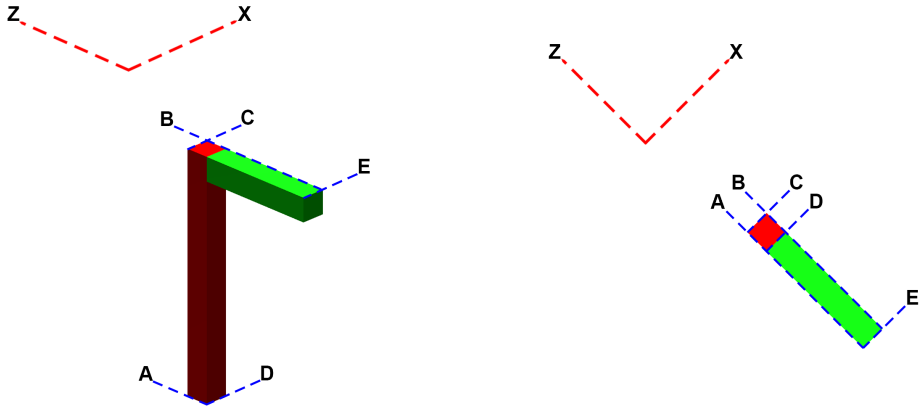


Figure 4.5: Choosing the best projection plane for an object. Each possible projection plane location runs parallel to the objects surface, from which the position and normal direction can be sampled.(*left*) Object viewed from the front. (*right*) Object viewed from above. Each of the possible projection planes are highlighted by a blue dotted line.

To create copycat cast shadows that appear connected to the casting object, the chosen projection plane should be positioned near where the object is in contact with the ground surface. This can be done by choosing a projection plane that lies parallel to one of the faces of the object making it possible to sample its surface normal and position from the object itself. Figure 4.5 shows an example of an object highlighting each of the possible orientations of the projection plane that lie parallel to the surface of the object. For this example, the projection plane normals lie either along the X or Z axis. To determine the plane that will give the best results image moments [4, 77, 43] are used to measure the similarity between the shape of the object and its copycat shadow.

The purpose of image moments are to describe the shape of an object in an image, based on features such as area, centroid, and orientation. The most basic form of moments, often referred to as raw moments, can be used to describe simple image properties such as area and the centroid. Moments are generally classified by the order of the moments, indices p and q of the moment M_{pq}. Given a binary image $I(x, y)$ we can use the zeroth-order moment, Equation 4.1, to calculate the area of the white pixels within the image. Whilst the first-order moment, Equation 4.2, can be used to calculate the centroid. Where M_{10} and M_{01} are the sum of the coordinates of all white pixels along the x and y axis respectively. Giving us a very basic description of the image.

$$M_{00} = \sum_x \sum_y I(x, y) \tag{4.1}$$

$$\{\bar{x}, \bar{y}\} = \left\{ \frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}} \right\} \tag{4.2}$$

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q I(x, y) \tag{4.3}$$

The problem with using these moments alone is that they are sensitive to the x and y positions of the pixels. To calculate moment values that are invariant to translation there are central moments, Equation 4.3. These central moments are used to describe features of the image, such as area, in relation to the centroid. For our purposes, we chose to use the set of 7 higher-order moments called Hu moments [33, 34]. These moments are translation, scale, and rotation invariant. Allowing us to get accurate image descriptors for our impossible object shape and shadow shape regardless of transformation differences.

To begin with, we segmented the image into its separate components, the object, and its cast shadow. Once we segmented the two desired features we then extracted the contours of each, see Figure 4.6. By using the contours rather than the full image data we are able to generate image moments based solely on the outline of the two segments. This eliminates any effect the different pixel intensities/values may have on the resulting image mo-

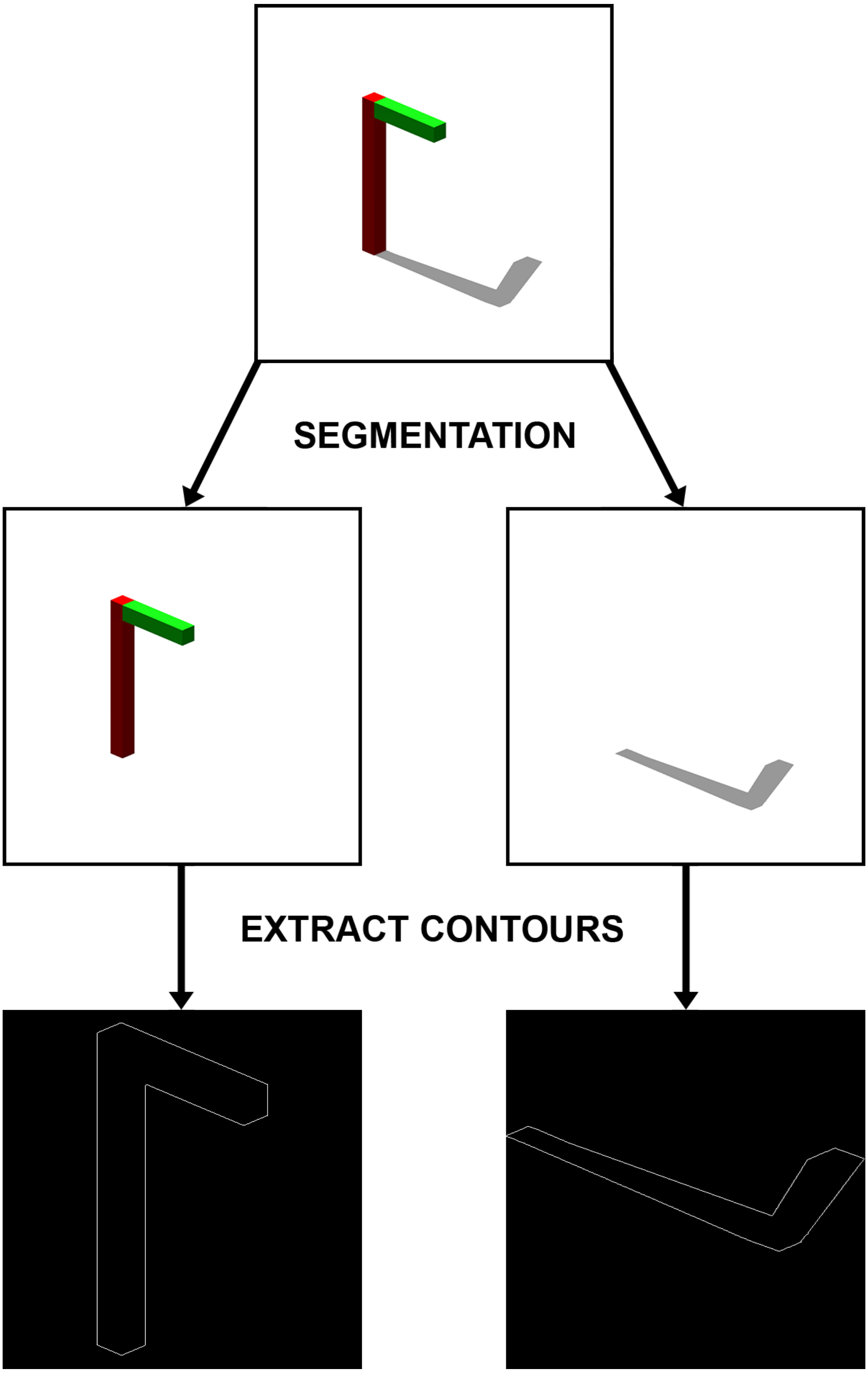


Figure 4.6: Figure demonstrating the process of preparing the rendered image to use in our image moment measurement. Beginning with the original rendered image we extract the two features we wish to compare, the object and its cast shadow. We then extract the contours of the objects and scale them to be the same ratio. By doing so we are able to achieve more accurate moment results.

	Hu Moment Reults							Distance
Object	1.0075	2.83179	3.25371	4.4581	-8.4508	-5.9547	8.47915	0
Shadow A	0.96421	1.95633	7.98568	8.08654	16.1239	9.1527	-17.2443	0.920419
Shadow B	0.981501	1-99931	7.44426	8.21651	16.2045	9.45006	-16.1905	0.862874
Shadow C	1.08238	2.28688	3.92057	4.86408	-9.61839	-6.38873	-9.30184	0.671305
Shadow D	1.05607	2.22451	3.88113	4.76441	-10.211	-6.79328	-9.08841	0.620877
Shadow E	1.05064	2.21251	3.88881	4.50906	8.8351	5.82953	-8.88474	0.719094

Table 4.1: Hu Moment results for the object in Figure 4.7. The distance measure is calculated by comparing the Hu moment results for each shadow version to those of the original object.

ments. We then normalized the size and aspect ratio of the two segments, doing so allows us to better generate moment values that are invariant to transformations. To generate the moment values themselves we used the built-in functions from the OpenCV library. Using the functions within the OpenCV library we can generate a set of Hu moments for each of the image segments. To measure how close both shapes are in appearance we calculate the distance between both sets of image moments, with lower distances indicating a higher similarity, shown in Table 4.1.

Figure 4.7 shows the results for each of the possible projection planes highlighted in Figure 4.5. The moment results are normalized between 0 and 1, where 0 would be a perfect match and 1 is the highest difference found among those projection planes. For this object, we found the projection planes with normals orientated along the Z-axis to give the best results, with plane D giving the highest object/shadow shape similarity.

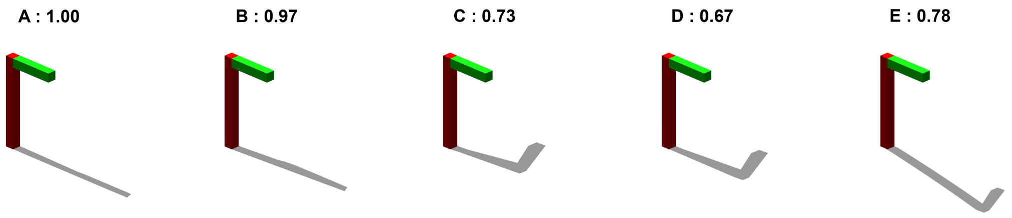


Figure 4.7: Alternate Projection Plane Results. Each object labeled with the projection plane location (A - E) from Figure 4.5. Along with the image moment result, calculated by comparing the image moments between cast shadow and object shape (normalized between 0 and 1) where the lowest value represents the highest object/shadow similarity. For this object the optimal projection plane would be D as it has the lowest image moment result.

4.2.2 Select Depth Information to Project

A key step in our screen space copycat shadow algorithm is to identify which objects within the scene should be projected in order to cast a copycat

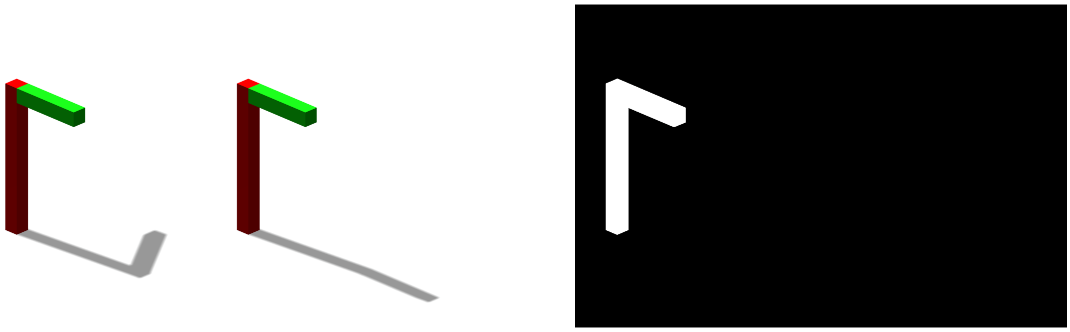


Figure 4.8: Specifying which objects within a scene should be projected. (*left*) Scene containing two identical objects, the object on the left has been chosen to cast a copycat shadow the object on the right is casting its traditional shadow. (*right*) The projection buffer for this scene. Only those depth values associated with the object on the left in the scene will be projected, this is highlighted by the white pixels in the buffer. All other depth information (black pixels) will not be projected.

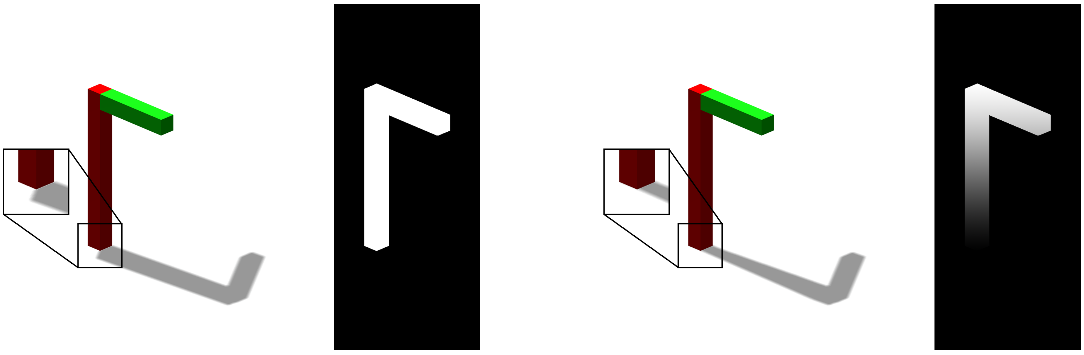


Figure 4.9: (*left*) Object has been fully projected onto the surface of the projection plane. This produces anomalies at the base of the object where it appears to be floating above the ground plane. (*right*) Gradually increasing projection strength with the distance from the base of the object. This generates copycat shadows whilst maintaining the objects connection to the ground plane. Alongside each object there is an example of the projection buffer where white pixels correspond to full projection and black to no projection. Pixels that lie in the gradient between black and white correspond to the projection strength applied, calculated using 4.4.

shadow. During the process of creating the scene a sentinel value, true or false, should be assigned to each object in the scene. This information will be stored within a new buffer, we have titled this the projection buffer, such that it can be easily sampled alongside depth information. The value at each pixel within this new buffer will determine whether the depth at that pixel should be manipulated in order to cast a copycat shadow, only the depth information belonging to those objects whose value is true will be projected. During the rendering pass, each time the depth buffer is sampled the projection buffer is sampled at the same pixel location. The value returned from the projection buffer will be used to decide if the sample depth should be projected prior to calculating occlusion. An Example of the new projection buffer is shown in Figure 4.8, only one of the objects in this scene has been set to generate a copycat shadow highlighted by the white pixels stored within the projection buffer.

If the object is in contact with the ground, projecting the object in its entirety can cause anomalies, as seen in figure 4.9. Where the object appears to be disconnected from the shadow resulting in the impression that the object is floating above the ground. To solve this our algorithm applies the projection gradually, along the height of the object, where no projection is applied at the base of the object. Using this gradual change of projection it is possible to generate copycat cast shadows whilst maintaining the connection of the object to the ground plane.

To achieve this we implement a transfer function that will vary the strength of projection based on the distance of a point in the image from the base of the object. Equation 4.4 can be used to calculate the strength of projection at each pixel (x, y), given the line at the base of the object $L(x, y) = Ax + By + C = 0$ and a distance threshold T. As the distance of the pixel from the specified line increases so does the strength of projection. The distance from this line is normalized between 0 and 1, where 0 is no projection (original depth information) and 1 is full projection (depth on the surface of the projection plane).

$$Strength = 1 - \frac{max(T - |L(x,y)|, 0)}{T} \tag{4.4}$$

4.3 Copycat Shadow Algorithm

Algorithm 1 Copycat Shadow Algorithm

```
RenderScene $\rightarrow$ (COLOR, NORMAL, DEPTH, PROJECTION see Fig. 4.8)
$C_{dir}$ = Camera view direction
for each pixel $P$ in PROJECTION do
    if $P$ is TRUE then
        Sample $D_{original}$ from DEPTH at $P_{xy}$
        Calculate projected depth $D_{projected}$ using 4.5
        Calculate difference in depths $\Delta D = D_{projected} - D_{original}$
        Calculate projection strength $S$ using 4.4
        Calculate depth offset $D_{offset} = (\Delta D \cdot C_{dir}) \cdot S$
        Calculate new depth value $D_{new} = D_{original} + D_{offset}$
        Store $D_{new}$ in DEPTH
    end if
end for
```

Our Copycat Shadow algorithm works by manipulating the original depths stored within the depth buffer prior to calculating occlusion. Our algorithm is supplied with the original depth information along with the new projection buffer from which new depth information is calculated for the specified objects within the scene. By only manipulating the information stored within the depth buffer itself our algorithm enables the creation of copycat cast shadows without the need to change the existing rendering pipeline. Algorithm 1 shows the processes involved in our copycat shadow algorithm. By supplying our algorithm with the projection buffer it is possible to reduce the number of calculations needed to only the pixels where the projection buffer returns true saving on unnecessary calculations and reducing the impact our algorithm has on the rendering speed.

Projected depth information Z_{proj} is calculated using the equation of a plane 4.5. Where a, b, c, and d, are the coefficients that represent the chosen projection plane, and x, y are the screen coordinates of the current sample location. This projection process adjusts the depths at the sampled pixel such that they lie on the surface of the specified projection plane, essentially flattening the three-dimensional object. By doing this we ignore the true geometric structure of the object. The resulting shadow more accurately represents the visible profile of the object. This copycat shadow appears more similar in shape to the casting object compared to the object's traditional cast shadow.

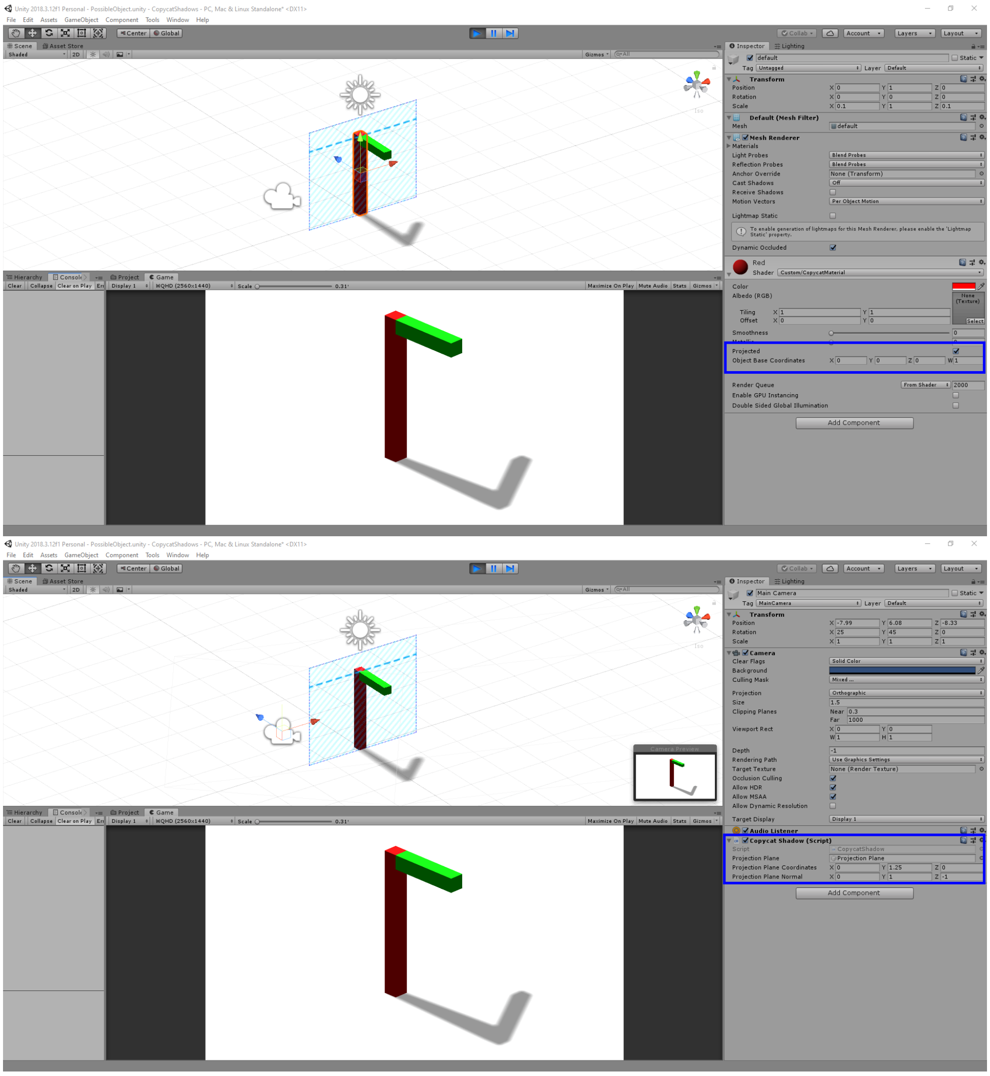
$$Z_{proj} = \frac{ax + by + d}{-c} \tag{4.5}$$

Original depth information is then projected along the view direction based on the projection strength, calculated using 4.4, and the newly calculated depth position. Taking into account view direction means it is possible to apply our copycat algorithm to any view direction along with both orthogonal and perspective viewpoints. Once the new projected depth information has been calculated we are able to use the existing rendering pipeline without the need for additional changes, the process of calculating points that are occluded in screen space can be done using existing methods [12]. By only manipulating the stored depth information it is possible of our algorithm work alongside existing features such as ambient occlusion and soft shadows.

4.3.1 Implementation

A key goal for our research was to ensure that our algorithms could work with already established tools, without the need to create a dedicated rendering environment. As with our impossible object modeling algorithm we chose to implement our copycat shadow algorithm using the Unity Engine. By negating the need for a dedicated rendering environment we hope to make our algorithm easier to implement for a wider range of users. Working within the Unity engine means we can take advantage of the built-in tools, using the existing modeling and lighting systems. Figure 4.10 shows a scene in unity using our copycat shadow algorithm.

To construct the scene a user can make use of the tools already available in the Unity engine, our algorithm works alongside the geometric models, lights, and cameras that are built into the Unity engine. Our algorithm is applied


Unity 2018.3.12f1 Personal - PossibleObject.unity - CopycatShadows - PC, Mac & Linux Standalone* <DX11>
Collab
Account
Layers
Layout
Inspector
Lighting
Add Component

default
Static □
Tag Untagged Layer Default

Transform
Position X 0 Y 1 Z 0
Rotation X 0 Y 0 Z 0
Scale X 0.1 Y 1 Z 0.1

Default (Mesh Filter)
Mesh default

☑ Mesh Renderer
Materials
Light Probes Blend Probes
Reflection Probes Blend Probes
Anchor Override None (Transform)
Cast Shadows Off
Receive Shadows □
Motion Vectors Per Object Motion

Lightmap Static □
To enable generation of lightmaps for this Mesh Renderer, please enable the 'Lightmap Static' property.
Dynamic Occluded ☑

Red
Shader Custom/CopycatMaterial

Color
Albedo (RGB) None (Texture) Select

Tiling X 1 Y 1
Offset X 0 Y 0

Smoothness 0

Projected ☑
Object Base Coordinates X 0 Y 0 Z 0 W 1

Render Queue From Shader 2000
Enable GPU Instancing □
Double Sided Global Illumination □

☑ Main Camera
Static □
Tag MainCamera Layer Default

Transform
Position X -7.99 Y 6.08 Z -8.33
Rotation X 25 Y 45 Z 0
Scale X 1 Y 1 Z 1

☑ Camera
Clear Flags Solid Color
Background
Culling Mask Mixed ...

Projection Orthographic
Size 1.5
Clipping Planes Near 0.3
Far 1000

Viewport Rect X 0 Y 0
W 1 H 1

Depth -1
Rendering Path Use Graphics Settings
Target Texture None (Render Texture)
Occlusion Culling ☑
Allow HDR ☑
Allow MSAA ☑
Allow Dynamic Resolution □

Target Display Display 1

☑ Audio Listener

☑ Copycat Shadow (Script)
Script CopycatShadow
Projection Plane Projection Plane
Projection Plane Coordinates X 0 Y 1.25 Z 0
Projection Plane Normal X 0 Y 1 Z -1

Figure 4.10: Screenshots of our copycat shadow algorithm being rendered using the Unity engine. (*top*) Demonstrates the material assigned to the object, highlighting the user input on the right. This material acts like a regular material but passes added information to our screen space shader. (*bottom*) Shows our custom shader applied to the scene camera. Highlighting the user input on the right. In this example, we have created a projection plane that is only visible in the editor, not when the final object is rendered. The projection plane is sampled directly from the plane object without the need for the user to manually enter each of the values.

through a combination of custom materials (applied to the objects) and a screen space shader (applied to the camera). The custom material will be used to construct the projection buffer, it will determine whether the object will be projected or not. When creating the custom material the user will supply a point at the base of the object, from which the strength of projection can be calculated using equation 4.4. This can also be found automatically by setting the current up axis, the vertex with the lowest value along the axis will then be chosen as the base of the object. Figure 4.10 shows the input fields the user can specify. This material behaves the same as the basic Unity materials, meaning the user can utilize the built-in coloring and texturing tools.

Our custom shader is then applied to the camera. The purpose of this shader is to use our copycat shadow algorithm to manipulate the depth buffer information before passing it on to Unity's built-in rendering pipeline. The user must supply the desired projection plane. This can be supplied by choosing a plane object in the scene, or by specifying a location in the scene and a surface normal direction for the plane. From this, our copycat algorithm calculates the plane coefficients needed for equation 4.5. Our custom shader will perform an initial pass to construct the projection buffer. For each pixel in the buffer, it will sample the intersected objects, if an object using our custom material is detected it will calculate the projection strength at that pixel using the information from our custom material and equation 4.4. Once the projection buffer is constructed the data stored within the depth buffer, constructed using Unity's built in-depth buffer tools, is manipulated using our copycat shadow algorithm. Once the depth buffer has been updated it is passed to Unity's existing rendering pipeline to calculate occlusion within the scene. Due to the low number of calculations performed during the buffer sampling process of our algorithm we are able to achieve these results in real-time.

Figure 4.11: Seemingly Impossible Model (SIM) created using line of sight deformations. (*left*) SIM as seen from the original view direction, giving the illusion of impossibility. (*right*) The same SIM viewed from above using the original deformation direction, here the true structure of the object can be seen.

4.4 Cast Shadows for Impossible Objects

To create a three dimensional model of a SIM, deformations are applied to the object geometry along the view direction [15]. These deformations are hidden from the current view position and can only be seen when the object is viewed from another direction. Using these deformations it is possible to create ambiguities within the structure of the object that gives the illusion of impossibility. Figure 4.11 shows an example of a SIM alongside a view of the SIM from above to reveal the object's true structure.

4.4.1 Traditional Cast Shadows

The images presented in Figure 4.12 show a range of seemingly impossible models (SIMs) rendered with their traditional cast shadows. Within each of the examples, the deformed section of the object used to generate the illusion of impossibility is highlighted in green. This deformation results in severe anomalies in the shape of the traditional cast shadow of each object that does not correspond to the perceived shape of the object. Due to this deformation in the cast shadows, they often appear strange to the viewer, detracting from the impossible illusion of the object. Currently, to avoid this issue SIMs are rendered without any cast shadows, this lack of an appropriate cast shadow limits their use in more complex scenes. We propose that copycat shadows can be a solution to creating appropriate looking cast shadows for these SIMs.

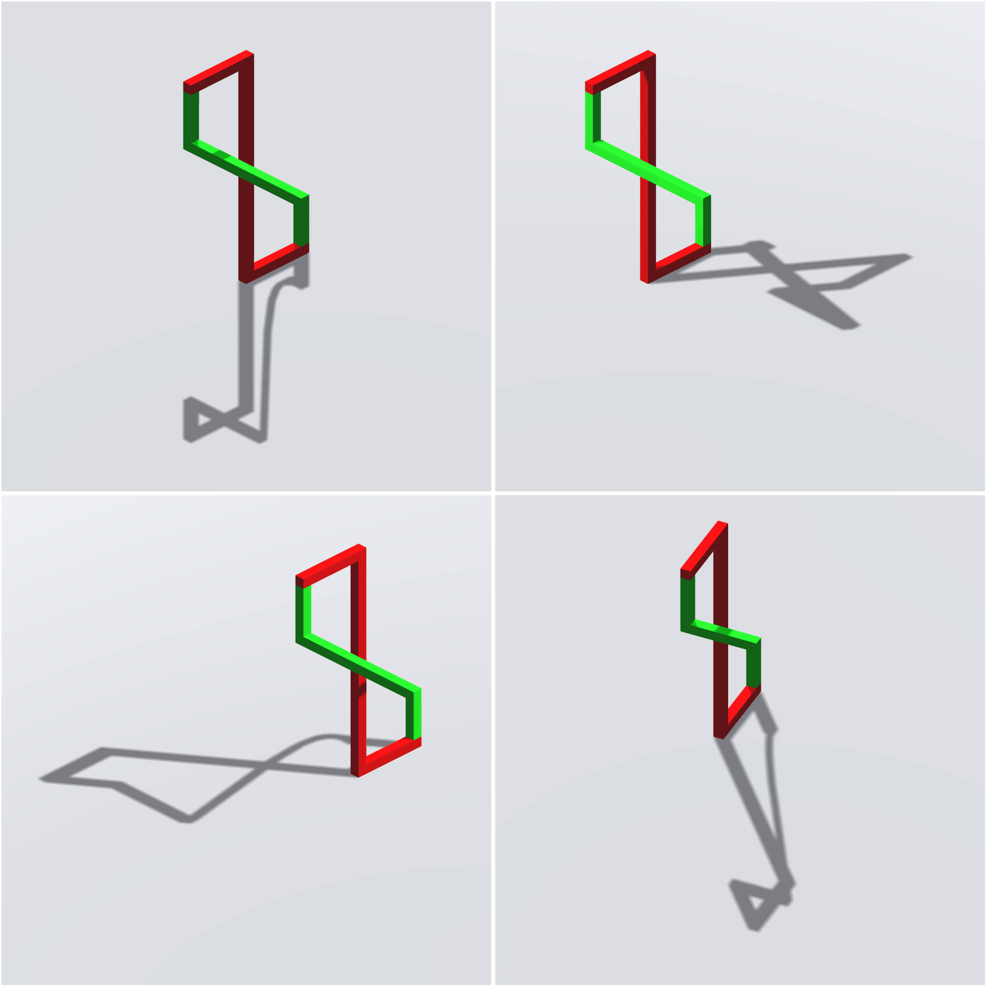


Figure 4.12: Seemingly Impossible Model (SIM) created using line of sight deformations, rendered using its traditional cast shadow. The hidden deformations used to create the illusion of impossibility result in deformations within the cast shadow. These traditional cast shadows often appear strange as the deformation in the shadow does not match the apparent orthogonality of the object.

4.4.2 Copycat Cast Shadows

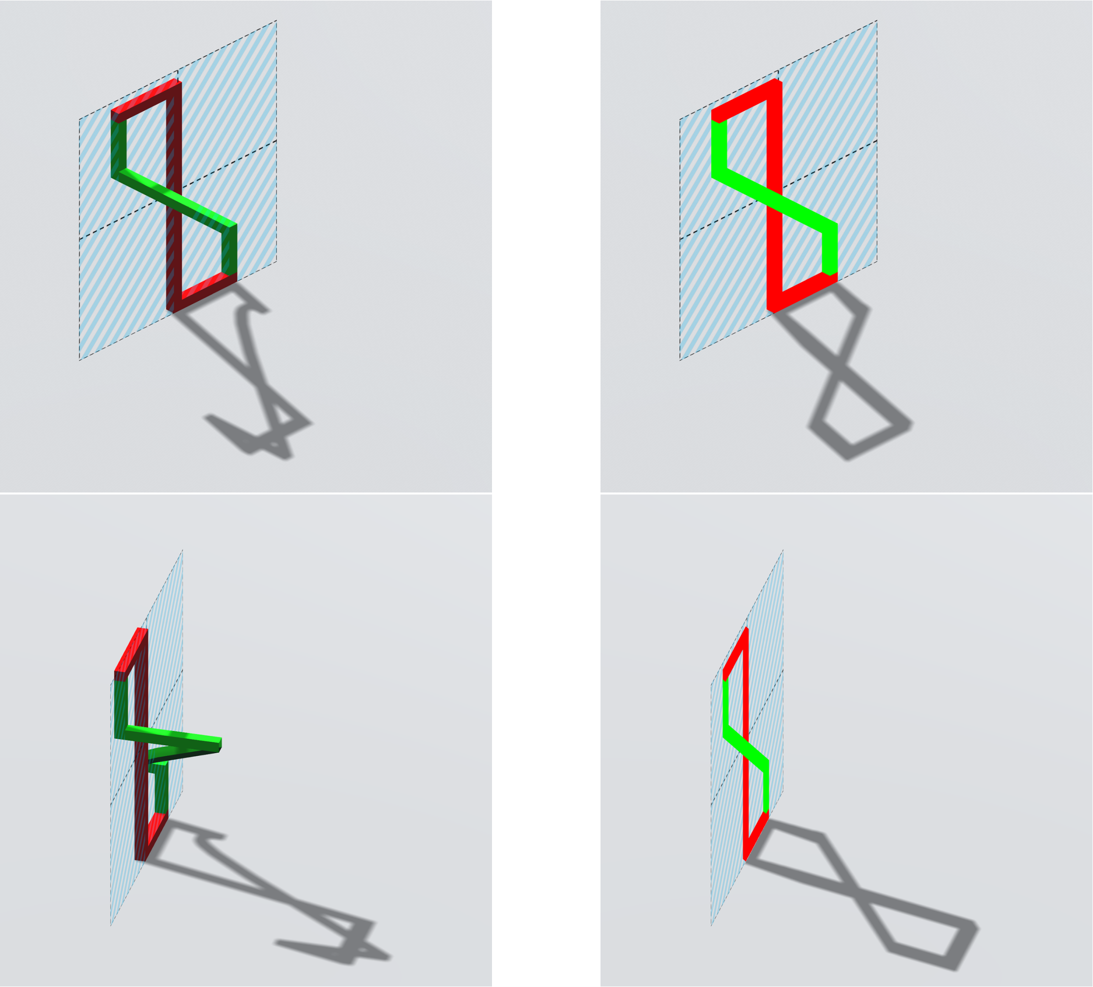


Figure 4.13: Casting appropriate shadows for seemingly impossible models using our copycat shadow algorithm. (*left*) Original three dimensional object, parts of the object are behind while others are in front of the plane in three dimensional space. (*right*) The object has been projected or flattened such that it lies on the surface of the three dimensional plane. As the projection is applied along the same direction as the deformation used to create the SIM, the deformation has no effect on the resulting copycat shadow. Using this flattened object silhouette we are able to create copycat cast shadows.

Our copycat shadow algorithm is not limited to possible objects, we are also able to produce copycat shadows for a range of seemingly impossible models. Due to the hidden deformation used to create the illusion of impossibility being applied along the line of sight our copycat shadow algorithm can use the same method of projection for these SIMs. By projecting the object depths onto a planar surface along the view direction the hidden deformations within the object do not affect the resulting copycat shadows. Figure 4.13 shows an example of a SIM and how our copycat algorithm can project the SIM onto the surface of a plane to create appropriate looking copycat shadows.

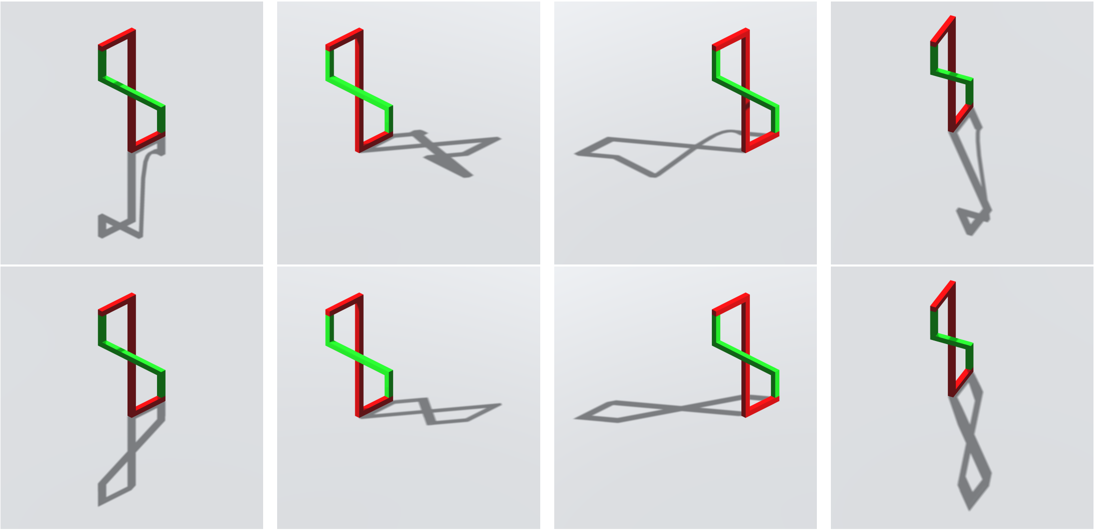


Figure 4.14: Seemingly Impossible Model (SIM) created using line of sight deformations, rendered using its traditional cast shadow (*top*) and our copycat shadow algorithm (*bottom*). Due to the projection used in our copycat shadow algorithm the hidden deformations used to create the illusion of impossibility do not have an effect on the resulting shadow. Comparing the two shadow versions these copycat shadows appear much more convincing as their shape better matches the visible shape of the object.

Comparing the SIMs traditional cast shadow to the copycat shadows generated using our algorithm, Figure 4.14, demonstrates that it is possible to generate visually appropriate cast shadows that do not contain anomalies caused by the hidden deformations. To account for the discontinuities in shading that are created on the sections of the object undergoing hidden deformations we employ the existing method of sampling surface normals from the object prior to applying deformation [79]. Our method of generating copycat cast shadows is not affected by the amount of deformation applied to the object as during the process of projecting the depth information we remove all of the hidden deformations. As our copycat algorithm works in screen space, manipulating the object depths such that they lie on the surface of a plane, the hidden deformations within the structure of the SIM do not have an effect on our algorithm meaning the same algorithm can be used across both possible and seemingly impossible objects.

4.5 Measuring Acceptable Light Directions

To examine how our copycat shadow algorithm performs under a wider range of light directions we employ the use of image moments [4, 77]. Based

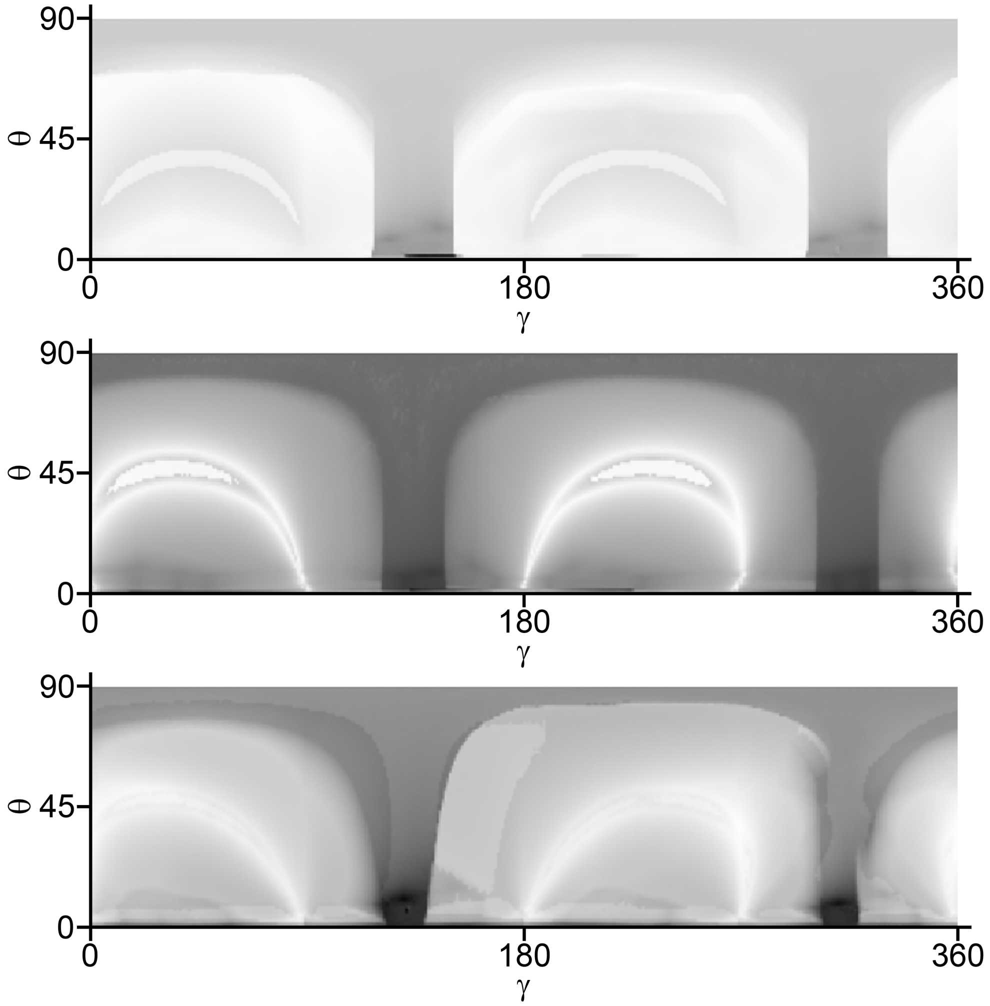


Figure 4.15: Difference between visible object shape and copycat shadow shape under different light directions, calculated through image moment comparison. γ and θ indicate the azimuth and altitude (in degrees) of the light direction, this range covers the hemisphere directly above the SIM. The color at each pixel ranges from black to white, where higher object/shadow similarity is mapped to white and lower object/shadow similarity to black. The difference value is normalized to [0,255] by considering the maximum and minimum values in each figure. The values are calculated using a distance function to compare the image moment extracted from the object to the image moment extracted from the cast shadow at each light direction. Top to Bottom, the results of the first three objects from Figure 4.20.

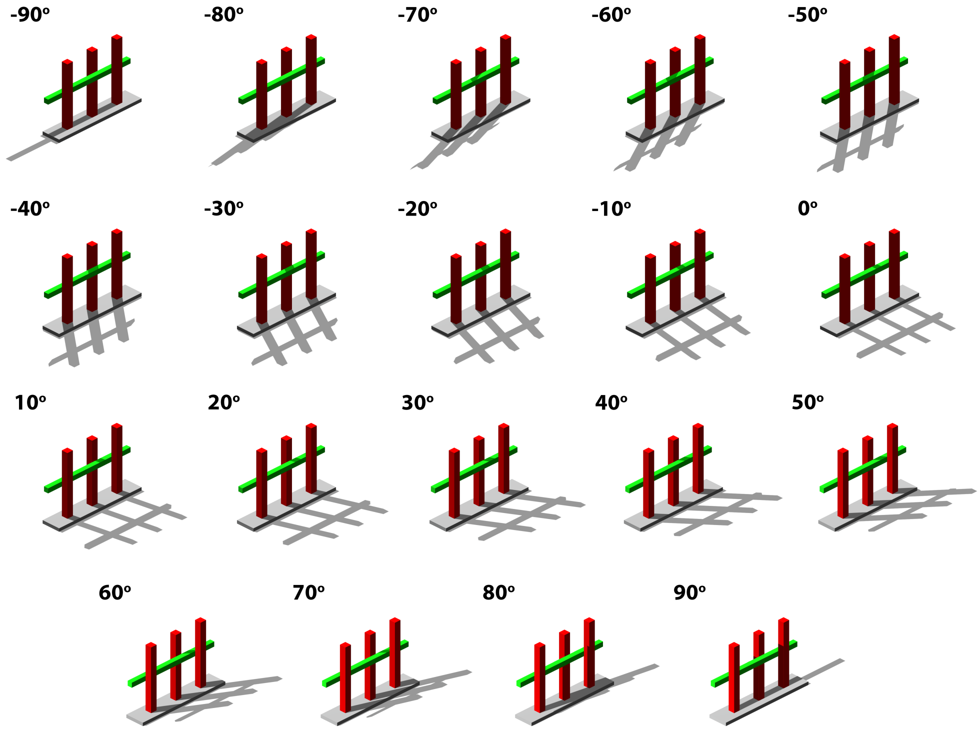


Figure 4.16: Example of a SIM rendered with our copycat shadow algorithm under a range of lighting directions. This example spans a range of 180 degrees around the azimuth, each separated by a 10-degree difference. We can see how our copycat shadow algorithm behaves under these different light directions, as the light approaches the extremes the cast shadow can appear flat which may not be appropriate for the more three-dimensional looking object. The best results can be seen towards the middle of the 180-degree range where the copycat shadow shape appears visually similar to the shape of the casting object.

on the similarity between the object shape and the shape of our generated copycat shadow we can asses how well our algorithm performs for each different light direction. We began by choosing a single view direction and setting up the projection plane for our copycat shadow algorithm, this would remain the same under all the sampled light directions. We then moved the position of the light source around the hemisphere above the object, 360 degrees along the azimuth and 90 degrees elevation in total sampling 32,400 light directions. At each light direction, we rendered the object alongside the cast shadow generated using our copycat algorithm. We then performed our image moment analysis for each light direction, calculating the image moment values for the copycat shadow shape and comparing those to the image values of the object shape using a distance function. The resulting distance functions were then normalized between 0 and 255 by considering the maximum and minimum values for each object.

Figure 4.15 plots the similarity results for the first three objects in Figure 4.20. There is a visible pattern within these results where a higher shape/shadow similarity is seen when the light direction is directly in front or behind the object, Figure 4.16 shows an example of this. As the light direction moves towards the edges of the 180-degree range the similarity between the object and shadow shape decreases, the best results can be seen when the light direction is close to 0 degrees. As the light direction approaches the edge of the 180-degree range (-90 degrees and 90 degrees) the resulting copycat shadow appears flat. Due to the method of projection used in our copycat shadow algorithm the depth information is projected onto the planar surface, meaning the three-dimensional structure of the object is lost. This results in a float looking shadow when the light direction comes from the side of the object. At these light directions, the resulting cast shadow no longer functions as a copycat shadow as it does not appear visually similar to the shape of its casting object.

For these light directions where our copycat algorithms performance suffers the cast shadow results can be improved by choosing a new projection plane with a normal direction that is closer to being parallel to the light direction. Figure 4.17 shows a set of objects both possible and impossible rendered under problematic light directions using both the object's traditional cast shadow alongside our copycat shadow method. Two versions of the copycat shadow are shown, those generated using the original projection plane appear very similar for each object since much of the depth information is removed during the process it is left with a simple rectangular shadow. By choosing an alternate projection plane where the normal direction is closer to the light direction it is possible to cast copycat shadows that more accurately replicate the visible shape of the object. To expand the range of viable light directions for the current implementation the results of the image moment measure could be used to choose a new projection plane, or restrict the light direction using simple thresholding.

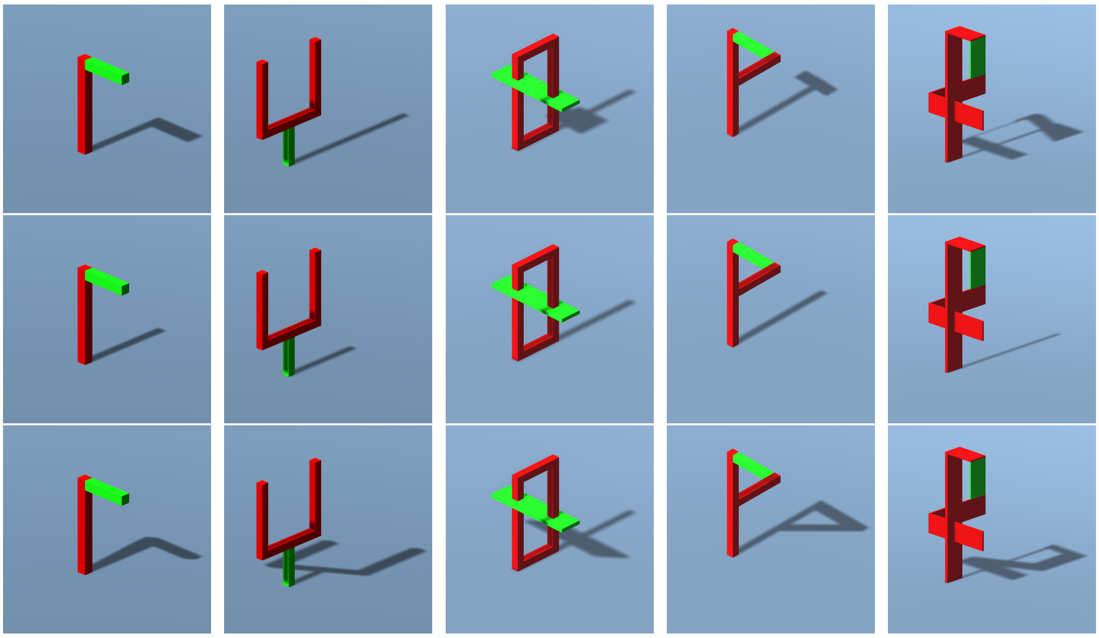


Figure 4.17: At light directions where the original projection plane fails it is possible to achieve more appropriate cast shadows by rotating the projection plane. *(top)* Objects rendered with their traditional cast shadows. *(middle)* Objects rendered with copycat shadows using original projection plane orientation. *(bottom)* Objects rendered with copycat shadows using rotated projection plane.

4.6 Results

4.6.1 Possible Objects

Copycat cast shadows can be used to make solving the shadow correspondence problem simpler, by imitating the shape of the casting object in

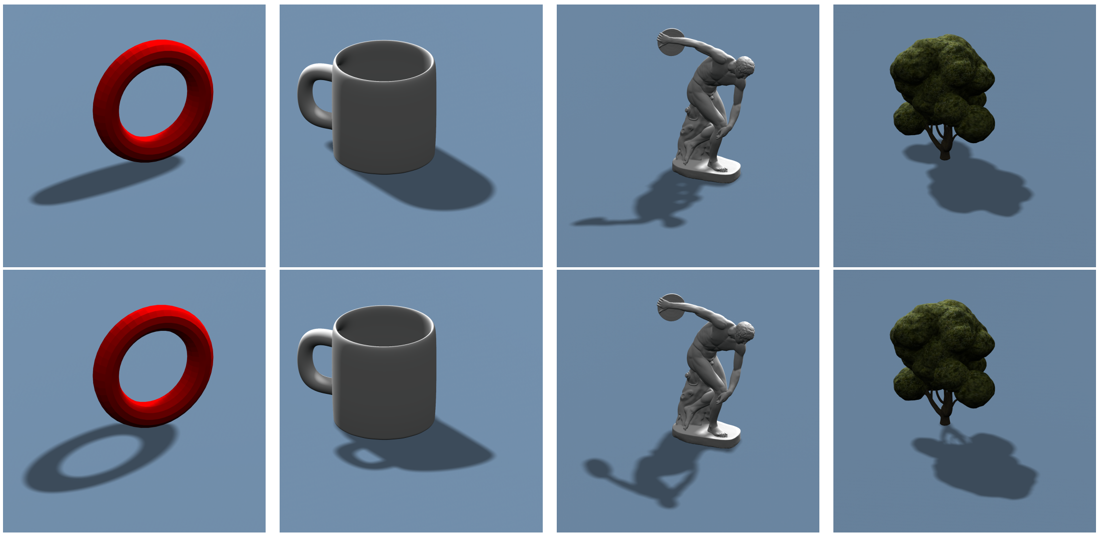


Figure 4.18: Possible objects rendered using their traditional cast shadow (*top*) and our copycat shadow algorithm (*bottom*).

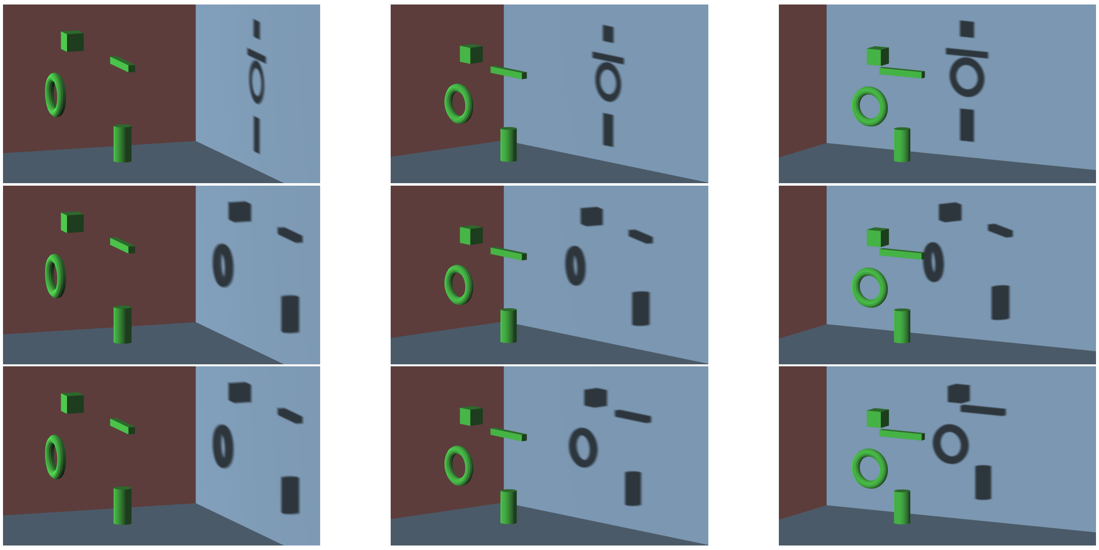


Figure 4.19: Four objects rendered using their traditional cast shadows (*top*), static copycat shadows, created by extracting the object shapes manually and adjusting their position and appearance to create the effect of cast shadows (*middle*) and our copycat shadow algorithm (*bottom*). Previous methods of creating copycat shadows would need to be manually updated for each view direction, without this the shadows would no longer be true copycat shadows. Using our algorithm new copycat shadows are generated automatically for each new viewpoint.

the shape of its shadow. Our algorithm aims to replicate the visible shape of the objects. The resulting copycat shadows often appear more compelling when compared to the traditional cast shadows. Figure 4.18 shows a series of possible objects rendered using their traditional shadow and our copycat shadow algorithm. In many cases, copycat cast shadows can appear more appropriate for the casting object due to their similarities in shape.

Our algorithm provides an automatic method for generating copycat cast shadows for a range of viewpoints and light directions. Existing approaches for generating copycat cast shadows require the use of manual post-processing effects, extracting the visible object and transforming it such that it resembles a shadow, limiting the use of copycat shadows in more complex and interactive scenes. Figure 4.19 shows a similar scene to one used by Casati [8] to demonstrate copycat shadows. Using previous copycat shadow methods when the scene is viewed from an alternate direction the copycat shadows must either be updated manually or remain static. Using our copycat algorithm it is possible to automatically generate copycat shadows for each new view of the scene, making it much more efficient for use in interactive scenes. A single projection plane is used for all three viewpoints of this scene with the copycat shadows being automatically updated each time to reflect the change in object shape and position.

4.6.2 Seemingly Impossible Models

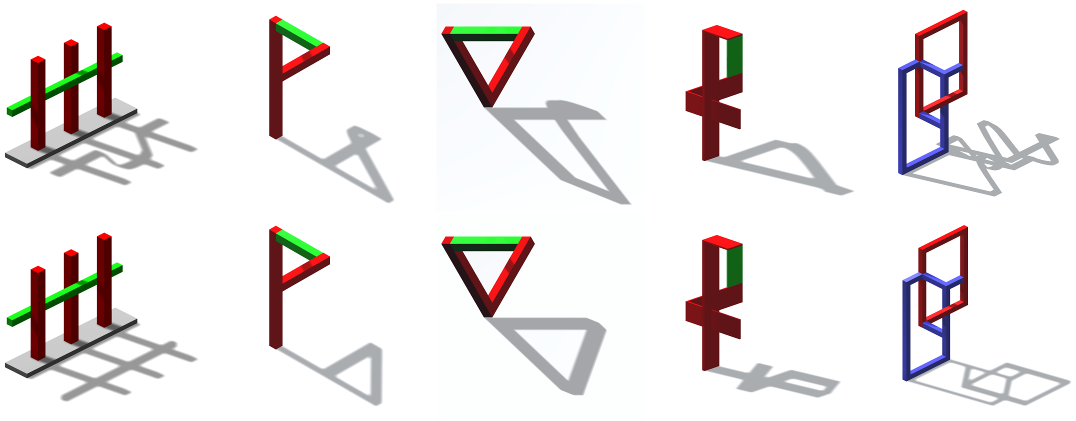


Figure 4.20: Examples of SIMs rendered using their traditional cast shadow (*top*) and our copycat algorithm (*bottom*). Each SIM shows a range of deformations needed to create the illusion of impossibility and cases where multiple deformations are used.

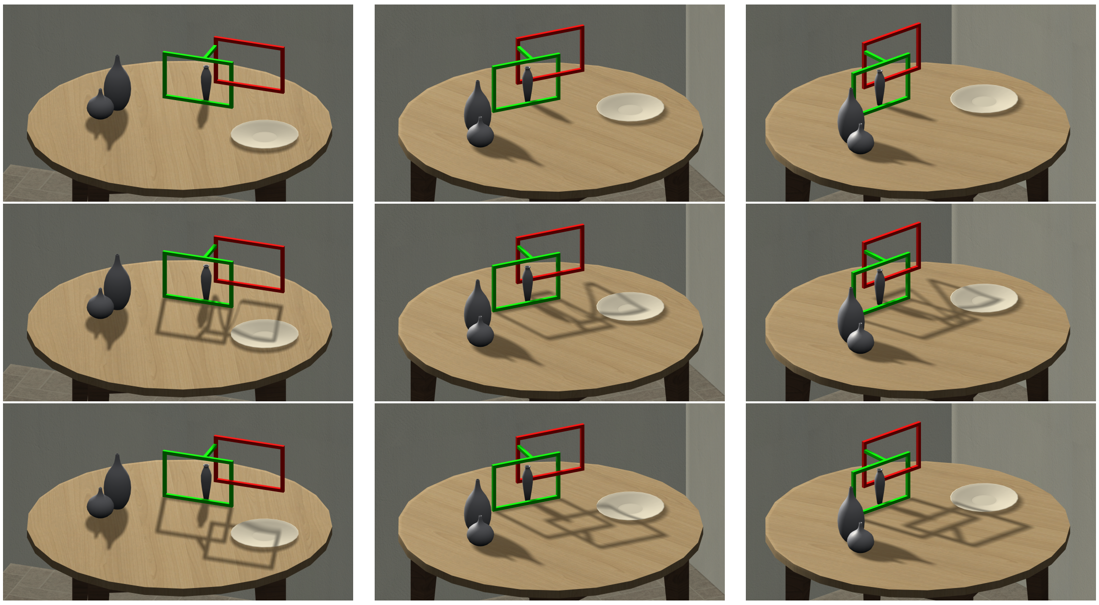


Figure 4.21: A SIM rendered within a scene alongside other objects. (*top*) Scene rendered by excluding cast shadows for the SIM. (*middle*) Scene rendered using traditional cast shadows for all objects. (*bottom*) Scene rendered using our copycat shadow algorithm to cast shadows for the SIM. Comparing these results we see that by using our copycat shadow algorithm we are able to convincingly render the SIM as part of a scene.

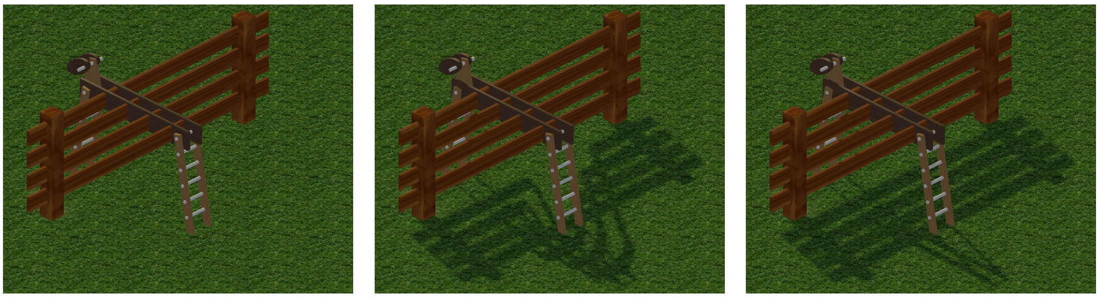


Figure 4.22: A representation of Del-Prete's "The Garden Fence", modeled using line of sight deformations and rendered using different forms of cast shadow. (*left*) Cast shadows completely excluded, similar to figure 14 by Elber [15]. (*middle*) Traditional cast shadows. (*right*) Copycat Shadow Algorithm.

Using our copycat algorithm we are able to create visually appropriate cast shadows for a range of Seemingly Impossible Models (SIMs). Figure 4.20 shows a set of SIMs created using existing deformation based techniques [15]. The hidden deformation results in deformations in the objects cast shadow, these deformations do not correspond to the visible shape of the object and appear distracting. Where previously these objects would have been rendered by excluding any form of cast shadow, our copycat algorithm enables us to render these SIMs alongside a visually compelling cast shadow.

The inclusion of a cast shadow for these SIMs enables their use in more complex scenes, Figure 4.21. By excluding shadows completely the SIM does not appear to be a natural part of the scene where all other objects cast a shadow, at the same time using the SIMs traditional cast shadow can be distracting due to its deformed nature. By using our copycat algorithm it is possible to render the SIM alongside all other objects giving the scene a more compelling appearance. Figure 4.21 also demonstrates how our algorithm handles such things as object occlusion and casting our copycat shadows onto the surface of other objects. Figure 4.22 contains a representation of Del-Prete's "The Garden Fence", similar to figure 14 by Elber [15]. By including the object's traditional cast shadows within this scene the apparently rigid fence would create a curved and deformed cast shadow, using our copycat shadow algorithm, however, it allows the inclusion of a cast shadow that does not distract from the overall image.

4.6.3 Existing Rendering Features

As our algorithm does not change the existing rendering pipeline. We have demonstrated the implementation of our algorithm into the Unity engine, allowing us to make use of existing rendering functions alongside our copycat shadows. Figure 4.23 demonstrates how our copycat algorithm performs under two separate light sources. Whilst Figure 4.24 shows copycat shadows for a SIM under both orthographic and perspective viewing. Under perspective viewing, it is also possible to implement existing ambient occlusion methods adding an additional sense of realism to the resulting image.

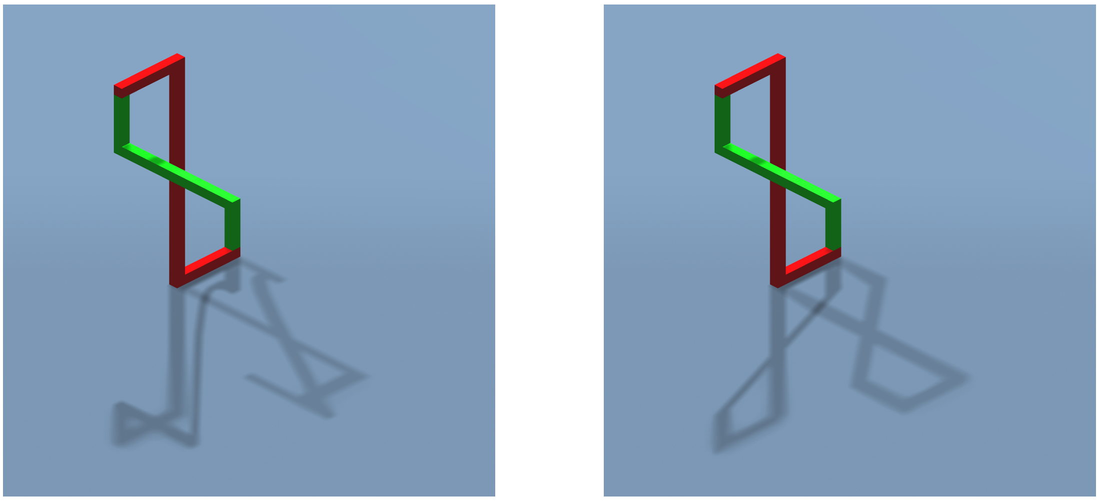


Figure 4.23: A SIM rendered under two directional light sources using its traditional cast shadow (*left*) and our copycat shadow algorithm (*right*).

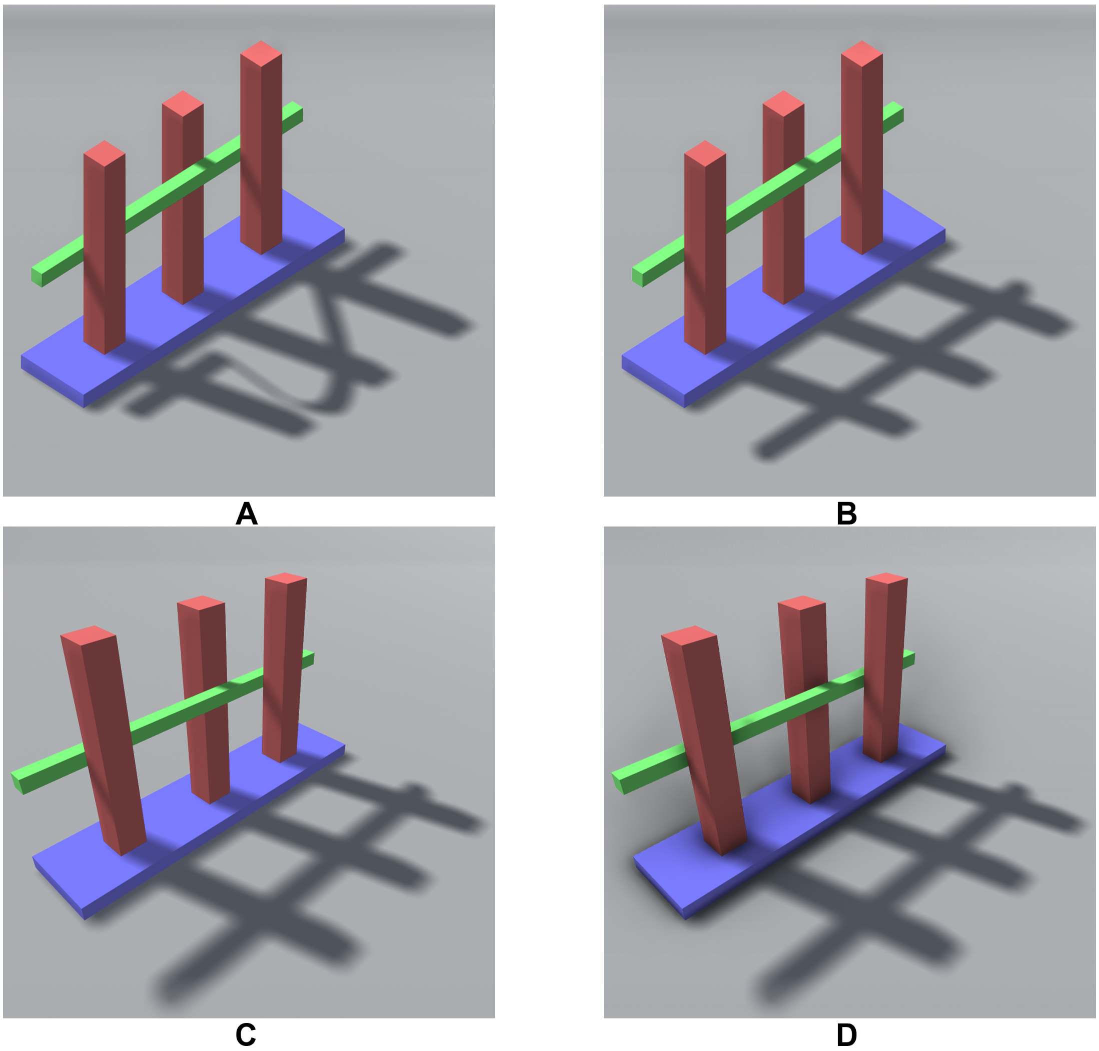


Figure 4.24: (*A*) A SIM rendered under orthographic projection with its traditional cast shadow. (*B*) The same SIM rendered using our copycat shadow algorithm. (*C*) Perspective rendering using our copycat shadow algorithm. (*D*) Rendering ambient occlusion alongside our copycat shadow algorithm.

4.6.4 Alternative Forms of Impossibility Modeling

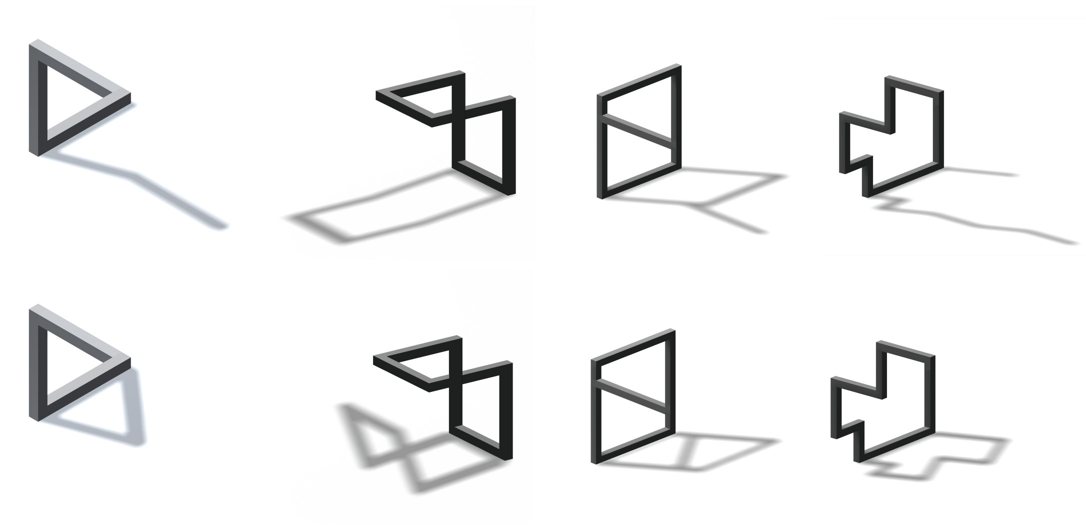


Figure 4.25: Impossible objects modeled using our transparency method and rendered with their traditional cast shadow (*top*) and copycat cast shadow (*bottom*).

As our copycat algorithm operates in screen space it is not fixed to a single form of impossibility modeling. A copycat shadow aims to replicate the visible profile of an object, meaning the hidden geometric shape that is not visible in the two-dimensional image has no effect on the shadow. This allows our copycat method to be applied to alternative methods of impossibility modeling. The SIMs used in previous figures have each been modeled using the line of sight deformation method [15]. Figure 4.25 demonstrates a set of SIMs modeled using our transparency based method where the surface transparency at sections of the object is manipulated to give the illusion of impossibility. Although our transparency method does not affect the geometry of the object we are still able to use our copycat shadow algorithm as the shape of the copycat shadows are only affected by the visible profile of the object.

4.7 Discussion

We have presented a copycat shadow algorithm that operates in screen space without changing the existing rendering pipeline, implementing it using the Unity engine allowing the use of already established features such as ambient occlusion. Our algorithm can work under a range of different light and view directions automatically, for both possible and seemingly impossible models. We have demonstrated how our copycat algorithm can create compelling cast shadows for these SIMs. Where previously cast shadows are completely excluded, opening up the ability to render these SIMs convincingly in a full three-dimensional scene. A current limitation to our algorithm is the available range of light directions. As we have demonstrated, under certain light directions our algorithm may fail to produce the optimal copycat shadows, this can be improved by changing the chosen projection plane. Further work should be done to enable the transition between these alternate projection planes automatically as the light direction changes. Currently using the image moment results, the allowed range of light directions could be restricted to only those that produce sufficiently accurate copycat shadows. By using a threshold value it may also be possible to choose which projection plane would give the best result for each light direction. However, improvements could be made by allowing for a gradual transition between the different projection planes, allowing for a continuous range of light directions to be used. This also proves to be the case for viewing directions, as the view direction approaches close to perpendicular to the projection plane the resulting shadows can appear problematic. Due to our algorithm projecting along the view direction if the angle between the view direction and projection plane normal is close to 90-degrees then the two will not intersect. For impossible objects, this may not occur often due to the natural limitation of view directions created by the object itself. This can be improved by selecting an alternative projection plane, however once again future work should be done to allow the smooth transition between projection planes. As convincing cast shadows have not been readily available for SIMs using previous methods we believe the use of copycat shadows can be an interesting avenue for further research both in rendering these objects and the perceptual relationship between the objects and their cast shadows.

Chapter 5

Visual Perception Of Impossible Objects

In this chapter, we perform four visual perception experiments to investigate the use of copycat shadows as a solution to casting visually appropriate shadows for impossible objects. Beginning by demonstrating the flaws of traditional cast shadows. Then comparing traditional cast shadows to the copycat shadows generated using our algorithm. Finally, testing the ability of our copycat algorithm to generate convincing copycat shadows under a range of light directions. Results from our experiments demonstrate that copycat shadows are not only visually preferable to the object's traditional cast shadow but also are able to convey the important spatial information for the object.

5.1 Introduction

Cast shadows have been shown to contain useful spatial and positional information about an object [35, 50, 82]. However, it has also been demonstrated that there is some leniency in our visual system as to when shadows are recognized as acceptable [39, 49]. Meaning they do not have to entirely correct for our visual system to recognize them as cast shadows. Evidence of this tolerance for inaccuracy has been seen in pictorial art, where artists have chosen to manipulate the lighting within their work for artistic intent rather than adhering to more realistic lighting constraints [10]. Although this alternate physics, seen in pictorial art, differs from what we may expect to see in the real world it does not appear jarring to the viewer, and often goes unnoticed unless carefully examined.

Artists use their own alternative physics when depicting a scene, arranging surfaces and lights to better fit their intent and not adhering to what would be considered real-world physics. Common manipulations come in the form of impossible cast shadows. One form of these impossible cast shadows are shadows that do not aim to represent the true three-dimensional structure of an object, but rather they aim to copy or imitate the visible shape of the object. These shadows, known as copycat shadows [8], although impossible are often accepted as correct and go unnoticed by the observer. Although not geometrically or physically accurate they are still able to convey important structural and positional information about the casting object. Due to the similarity between the shape of the object and its copycat shadow it may provide a simpler solution to the shadow correspondence problem, that is matching the shadows to its respective casting object.

Impossible figures are often seen as simple two-dimensional geometric figures, either line drawing or simple greyscale images. It has been shown that although we are able to understand the structure of these impossible figures our visual system is slower to process them than possible objects [23, 28]. Methods of modeling three-dimensional versions of these impossible objects have been created [15, 84]. These three-dimensional models introduce a new challenge of rendering the model under accurate lighting, this includes the objects cast shadow. Currently, due to the nature of the three-dimensional impossible objects structure, their traditional cast shadows appear deformed or strange. The current solution to this is to exclude the cast shadow completely. this results in the loss of any important information the cast shadow may contain along with making the object problematic to render as part of a scene as it lacks a cast shadow. We propose that copycat shadow may provide a solution to rendering appropriate cast shadows for these impossible objects. We performed a series of visual perception experiments to asses whether copycat shadows are perceived as appropriate shadows for impossible objects along with examining whether they still convey useful spatial information about the objects.

5.2 Experiment Stimulus

Each of the objects used in our four experiments were randomly chosen from a set of 20 impossible objects, see Figure 5.1, and their possible representations, see Figure 5.2. These 20 objects were selected as they have been used in previous visual perception research [23, 28, 26]. Impossible objects were created using the line of sight deformation method proposed by Elber [15]. To create possible versions of these impossible models we removed the deformation that had been applied, restoring the object to its original (undeformed) geometry. Objects were rendered using their traditional cast shadows generated by conventional occlusion methods. Copycat shadows were created using our copycat shadow algorithm.

5.3 Experiment 1: Identifying Impossible Objects

We see from our research that traditional cast shadows are excluded from the rendering of impossible objects as they appear strange or deformed due to the hidden structural manipulations used to create the object's impossibility. Including these deformed shadows alongside the object may make the object appear even more unnatural. Our aim for this experiment was to investigate what impact these deformed shadows had on a viewer's perception of impossible objects. We predict that the inclusion of these deformed cast shadows detracts from the impossible object illusion making the object appear unnatural within the scene. We tested this prediction by tasking observers to identify an impossible object among a set or possible objects.

5.3.1 Experiment Design

Our experiment design is based on that used by Ostrovsky et al. [60] to investigate illumination inconsistencies within a scene. In their work, they

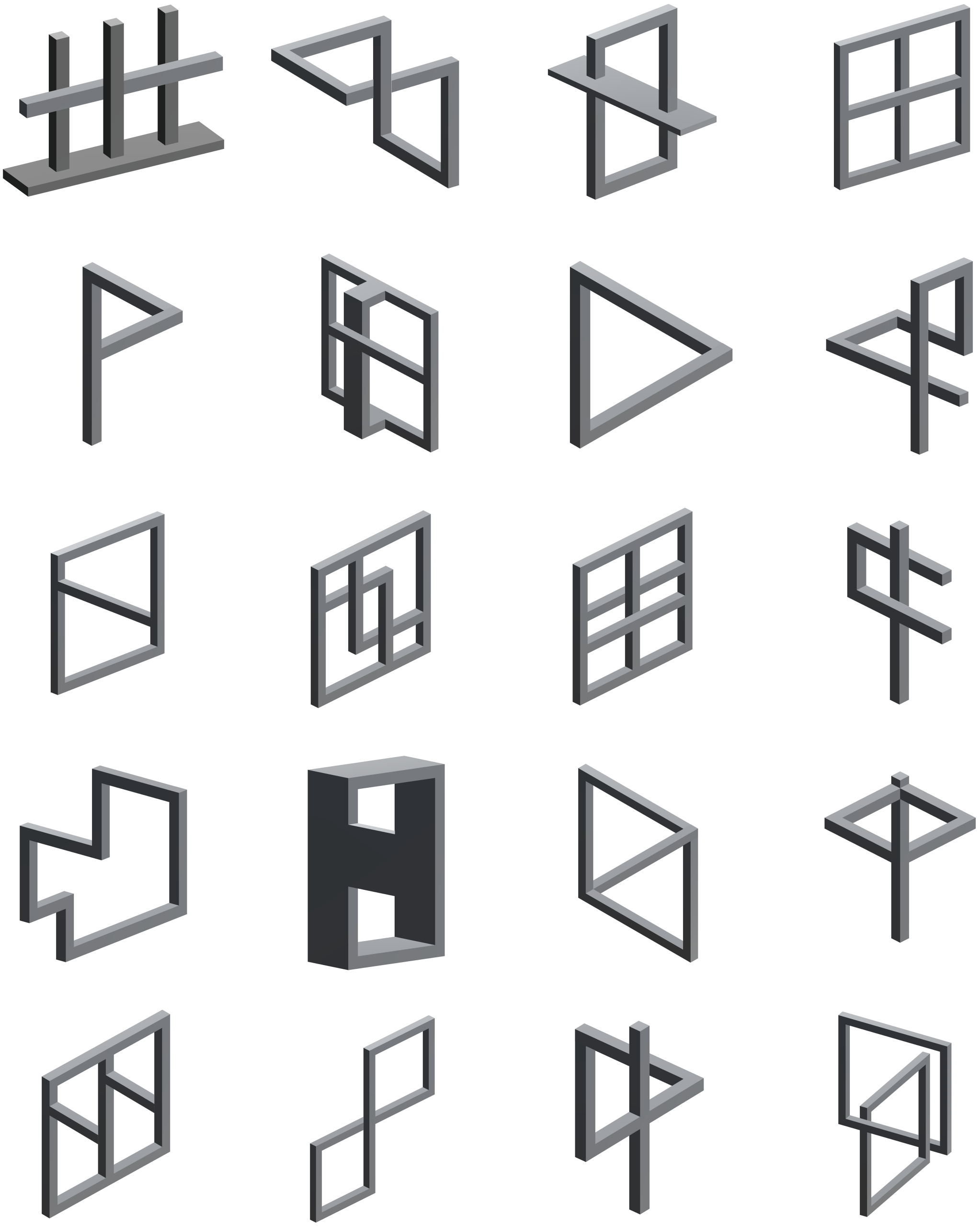


Figure 5.1: Impossible objects used in our visual perception experiments. Modeled using line of sight deformation methods.

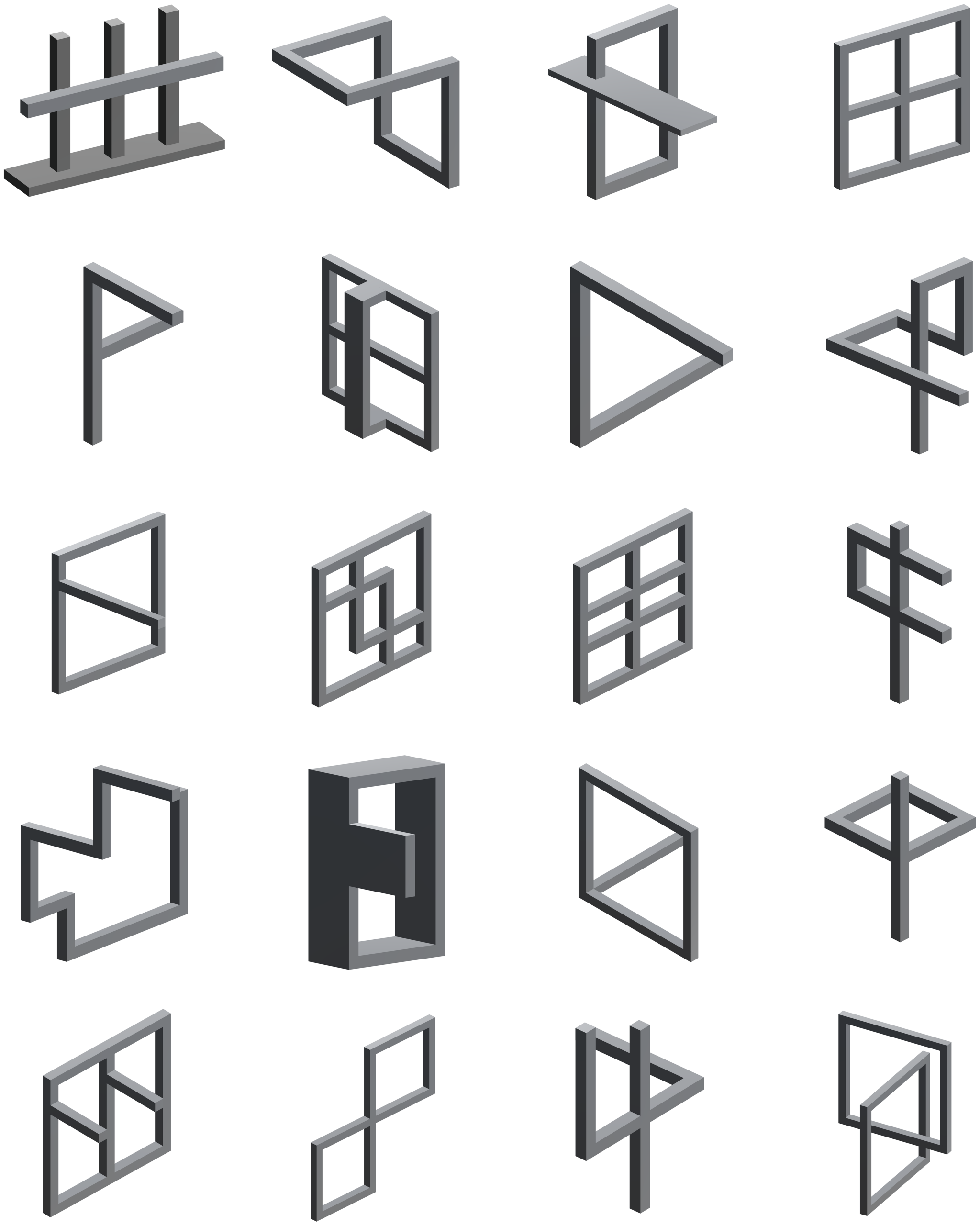


Figure 5.2: Possible objects used in our visual perception experiments.

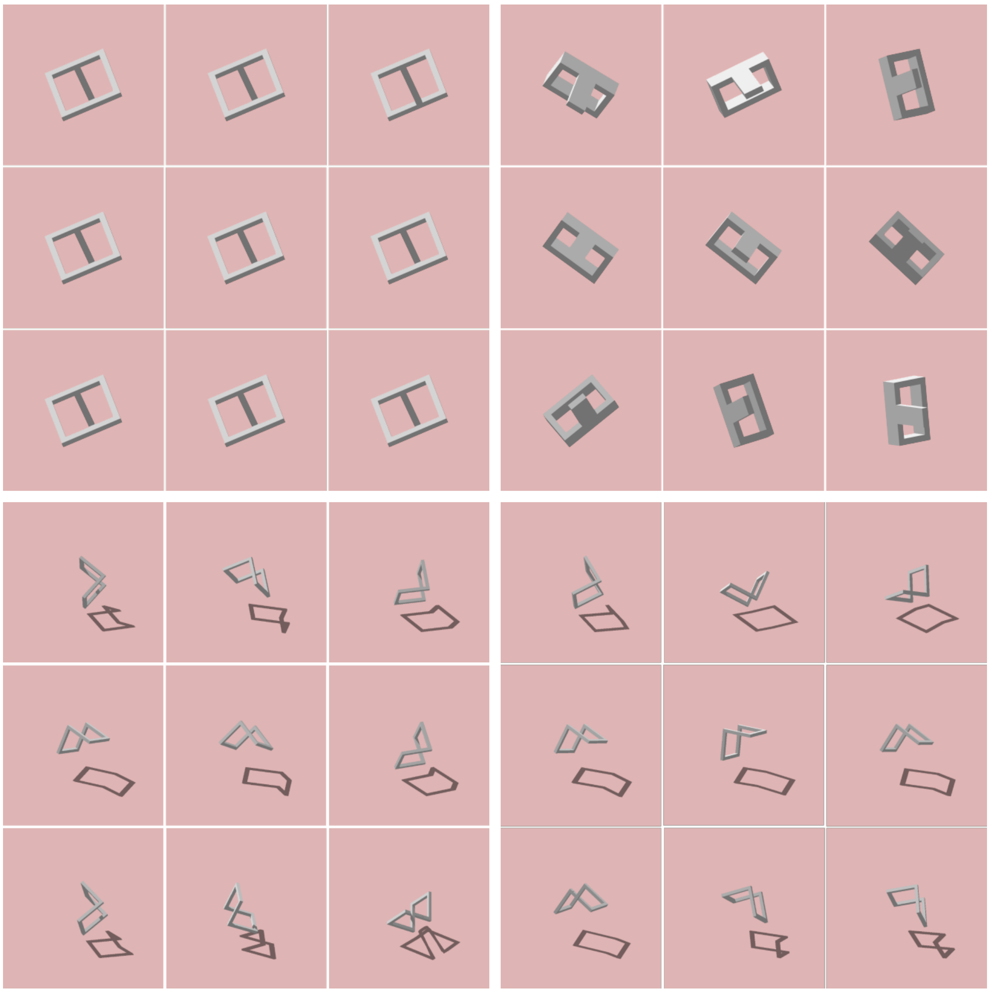


Figure 5.3: Experiment 1 Stimulus Examples. (*Top Left*) Homogeneous orientation, (*Top Right*) heterogeneous orientation, (*Bottom Left*) traditional cast shadow (impossible target object will cast a deformed shadow) and (*Bottom Right*) false shadow (target object will cast the shadow created by its corresponding possible object representation).

presented a series of images containing a set of cubes each illuminated from a single direction, whilst one target cube was illuminated from a different direction. Participants were tasked with reporting whether the display contained an anomalously illuminated target. We followed a similar setup for our experiment design. Each stimulus case would consist of 9 objects arranged in a 3x3 grid. One object would be an impossible object (the target object) the other 8 would be the corresponding possible version of that object (the distractor objects). For each stimulus, the target object's position in the grid would be randomized. With participants being tasked to identify whether the display contained an impossible object. Ten impossible objects and their corresponding possible counterparts were randomly chosen from our object poll for use in this experiment.

Our experiment was broken down into four variations each displaying a different type of stimulus, shown in Figure5.3. For version 1 we wanted to establish a baseline to determine how well participants identified the impossible objects when no other factors were involved. For this stimulus, each of the 9 objects were oriented in the same direction, homogeneous orientation, and presented with no cast shadow. This would allow us to see how well observers could identify impossible objects within a very structured scene. Similar Ostrovsky et al. [60] using homogeneous orientation alone may mean our results are not solely based on an observer's ability to identify the impossibility within an object. When all objects are identical in their orientation pop-out may occur based on simple pattern matching strategies of the observer's visual system. To reduce the impact of pattern matching strategies we also created stimulus under heterogeneous orientation. Each of the 9 objects would be orientated in a different random direction. This manipulation would not only reduce the impact of pattern matching strategies but force observers to rely on the impossibility illusion to identify the target object. This stimulus would also be a better representation of real-world conditions as it would be less rigid and structured. Both these versions would help us establish an understanding of how well observers could identify impossible objects within a set of possible object distractors based solely on the impossibility within the structure of the object.

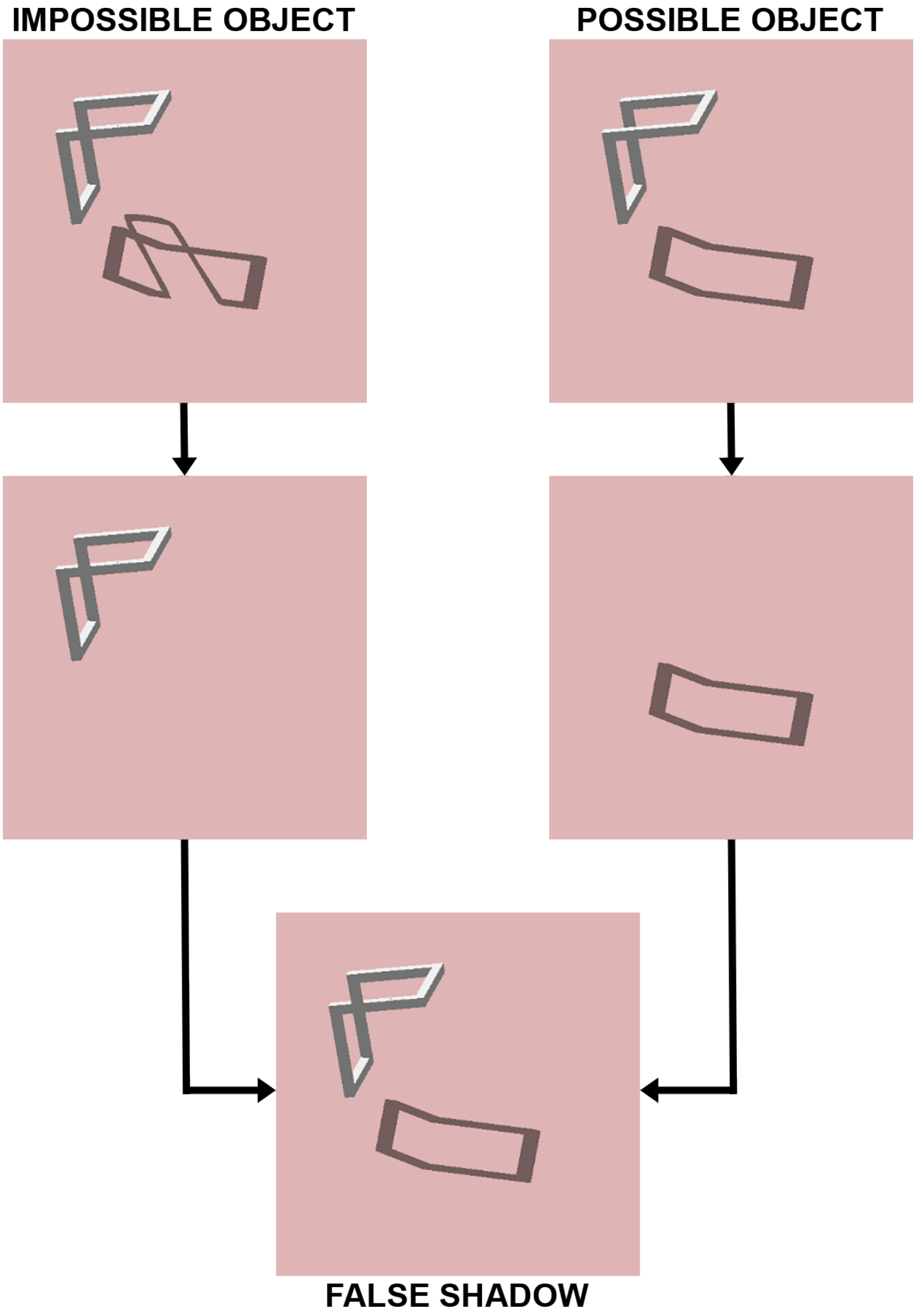
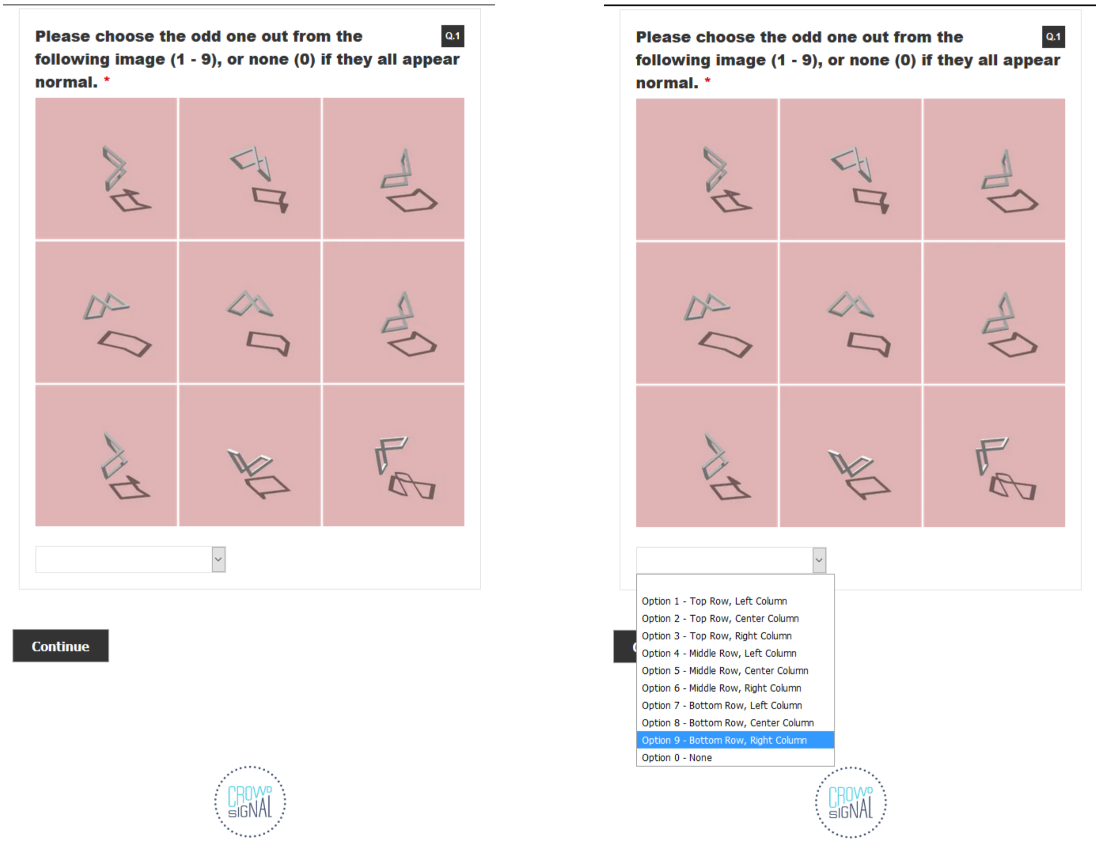


Figure 5.4: Figure demonstrating how false shadow stimulus was created for our experiment. Beginning with a matching pair of possible and impossible objects at the same orientation, we use the impossible object along side the possible objects cast shadow. Combining them together to create our false shadow stimulus.

To build on this we wanted to investigate what impact the inclusion of traditional cast shadows would have. For stimulus version 3 we used the same heterogeneous setup as before with the addition of rendering the objects cast shadows. For these cases, the target could not only be identified by the impossibility within the structure of the object but also the unnatural and deformed shape of its cast shadow. The expectation is that the deformation present in the shadow would make the target impossible object much more prominent to observers. To understand whether this effect was due to the deformation within the target objects cast shadow or whether it was simply the inclusion of any cast shadow we constructed experiment stimulus version 4. For this stimulus we used the same heterogeneous setup with cast shadows, however for the target impossible objects we removed the deformed cast shadow, replacing it with an undeformed cast shadow rendered using the corresponding possible object (a false shadow). Figure 5.4 demonstrates how these false shadows were created. This change would help us identify the effect the deformed cast shadow would have.

Each subject would be presented with one version of the stimulus images, homogeneous, heterogeneous, traditional shadow, or false shadow. Subjects would not be presented with more than one version of stimulus to avoid any learning bias. Subjects were tasked with choosing which objects from the 9 presented was the odd one out. To indicate which object was the odd one out our subjects chose the objects from 1 to 9 based on its position in the grid. An example screen from our experiment is shown in Figure 5.5. The experiment was held online using a survey hosting website. With participants being recruited through a survey subreddit. We chose this route to give us the best chance of reaching a large number of participants. There are however restrictions with this method to how accurately we can control experiment conditions. Each subject was shown a block of ten images one at a time to keep the experiment time down and encourage more participation. The accuracy of choosing the impossible target from the possible distractor objects was recorded. There was no time restriction in place allowing subjects to view the stimulus as much as they desired, they could however not return to the previous stimulus once they had submitted an answer. For each stimulus



Please choose the odd one out from the following image (1 - 9), or none (0) if they all appear normal. *

Q.1

Continue

Please choose the odd one out from the following image (1 - 9), or none (0) if they all appear normal. *

Q.1

Option 1 - Top Row, Left Column
Option 2 - Top Row, Center Column
Option 3 - Top Row, Right Column
Option 4 - Middle Row, Left Column
Option 5 - Middle Row, Center Column
Option 6 - Middle Row, Right Column
Option 7 - Bottom Row, Left Column
Option 8 - Bottom Row, Center Column
Option 9 - Bottom Row, Right Column
Option 0 - None

Figure 5.5: Screenshot of experiment one held online. Each image is presented one at a time, with participants submitting their answer using a drop down box containing the list of object positions. Participants must submit an answer before they are allowed to proceed to the next stimulus image.

image, the subject was required to choose the odd object out or indicate they all appeared the same using a drop-down list containing the grid locations of the 9 objects. They were only given instructions to choose the odd one out with no indication of what would make the object ’odd’, this was to keep subjects naive to the purpose of the experiment. We wanted to avoid priming subjects to look for object impossibility or deformations in the cast shadows. There was no mention of impossible objects or cast shadows in the experiment description.

5.3.2 Participants

Participants were recruited online from a subreddit for hosting and completing surveys/experiments. As such each subject participated voluntarily in the experiment with no reward promised. We believe this is a benefit for our data as subjects were completing the experiment under their own goodwill, in our opinion we believed this would reduce any participant from trying to influence the result or participate multiple times as there was no reward offered. We still, however, wanted to control the integrity of our data as much a possible. We did this through login / IP address data that would not allow the same subject to participate more than once, there are ways to circumvent this however again we believe as there is no reward offered this would be less likely. This data was not handled by us it was done through the survey hosting site used.

To help strengthen the integrity of our data we discarded those results where a participant started but did not complete the experiment, along with those participants with a completion time far outside the average range of 4 minutes 37 seconds. By doing this we hoped to eliminate any participants who were completing the experiment too quickly, perhaps randomly choosing answers, along with those who had perhaps started the experiment and returned after a long period of time. We also eliminated any participants who had a clear pattern in their responses, say choosing position 1 for every question. By doing this we hoped to eliminate as much unreliable data as possible. In total 312 unique subjects participated across the four different stimulus versions. Participants consisted of 189 males and 123 females aged between 19 and 46.

5.3.3 Results

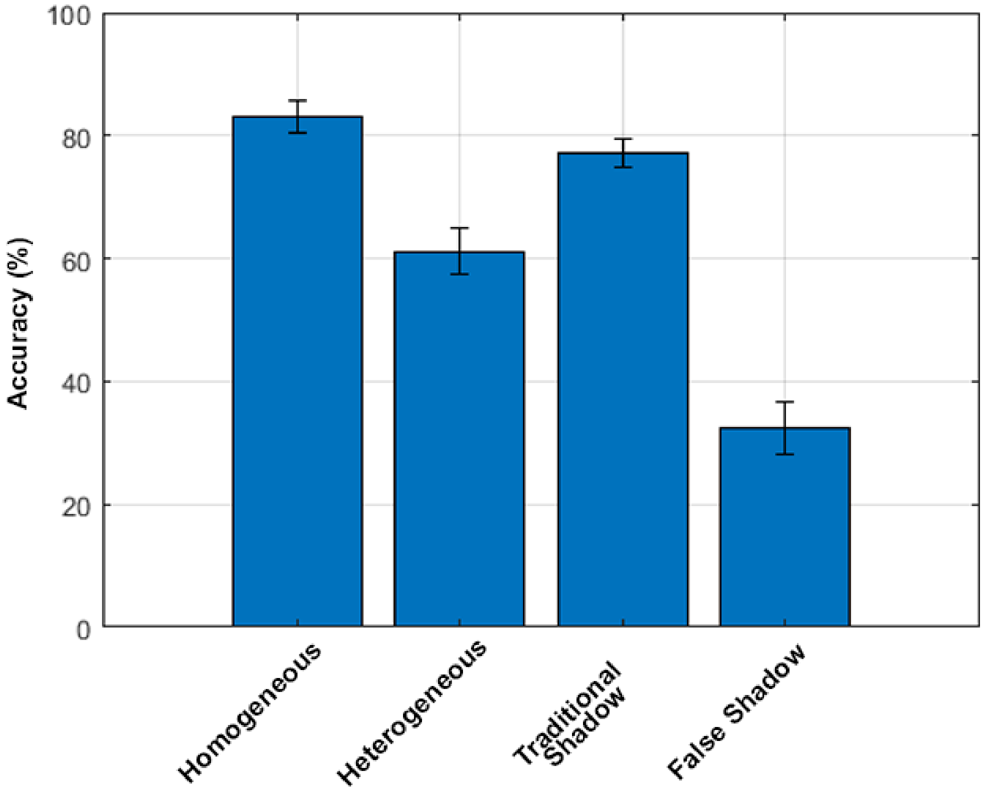


Figure 5.6: Experiment 1 Results. The graph shows accuracy (%) of participants correctly identifying the impossible target object for each stimulus type. Accuracy decreases with the heterogeneity of object orientation as hypothesized and increases with the introduction of the traditional shadow, suggesting that the deformations present in the traditional shadow aid in finding the impossible target object. Interestingly when the false shadow is used the performance decreases dramatically, this may be down to the similarity between the target objects shadow and those of the distractors giving a false impression that all objects are identical.

Answerers were collected for each participant in terms of their accuracy in choosing the target (impossible) object correctly, shown in figure 5.6. To establish a baseline we performed both a heterogeneous and homogeneous orientation experiment with no cast shadows. Results from these tests are consistent with previous work [60], with subjects achieving high accuracy (around 80%) where all objects have an identical orientation. For heterogeneous cases, where the same light direction is used across all cases but their individual orientation is randomized, performance dropped to around 60%. Indicating an inability among the subjects to correctly identify the impossible object from the group under heterogeneous orientation. To test the significance of this result we performed single factor ANOVA on the two groups. Resulting in a p value of $< 10^{-3}$ indicating the significant effect of heterogeneity.

This heterogeneous result gives us a baseline from which to compare results from our two cast shadow types. To begin with, we tested the object's traditional cast shadows. Under this case, the impossible target object would cast strangely shaped shadows due to the hidden deformations in their object structure. The results of this test are consistent with our hypothesis that these deformations would stand out and allow subjects to easily identify the target impossible objects. When traditional shadows were introduced performance rose to around 80%, close to the performance seen using homogeneous orientation. Indicating that traditional cast shadows have a significant effect ($p < 10^{-3}$) on the subject's ability to detect the target object compared to having no shadows present. We predict this increase is due to the obvious deformations present in the impossible object cast shadow.

To better identify whether the deformation in the target objects shadow was the cause of this increased performance we tested the objects using false shadows. Where target impossible objects would use a shadow created by their corresponding possible representation, eliminating any deformations present in the cast shadows. Interestingly the results of this test not only show a remarkable decrease in subject performance ($p < 10^{-8}$) compared to traditional cast shadows, as we hypothesized, but we also see a decrease in performance ($p < 10^{-4}$) compared to no shadows. This result supports our reasoning that the deformation in the traditional cast shadows aids subjects in locating the target object as when shadows are included without this deformation subject performance decreases.

5.3.4 Discussion

From our results we see that, consistent with results by Ostrovsky et al. [60], as we introduce heterogeneous orientation to our stimulus displays the subject's ability to correctly identify the target object drops significantly. We suspect this is down to their inability to rely on pattern matching strategies as each of the objects in the display are rotated randomly. The heterogeneous test results show that subjects are particularly insensitive to object possibility when there are no additional visual cues to assist them in their decision. This is in line with results by Freud et al. [28]. Within the heterogeneous test, each of the objects within a stimulus display appear globally similar, with only small local changes within the structure of the object to differentiate the impossible target objects. This may suggest that similar to results found by Freud et al. [24], our participants processed both the possible and impossible objects within the display based on their high-level global structure without focusing further on the finer details of the structure of the object.

With the introduction of the target objects traditional cast shadow results show the performance of participants rise, near to the level of the homogeneous results. This supports our hypothesis that the inclusion of the impossible objects traditional cast shadow makes it stand out, even though it is the correct cast shadow. The results from the false shadow test also support this. The decrease in performance when false shadows are introduced strengthens our conclusion that it is the deformation within the impossible objects traditional cast shadow that causes it to stand out among the distractors. This decrease also suggests that participants viewed the entire scene on a coarse scale, giving them a false impression that all objects are identical, as their cast shadows appear very similar. This is in line with results produced by Mamassian [49] where subjects did not notice that the shadows cast by their stimulus objects did not match the actual shape and orientation of the object itself.

These results seem to support our claims that the inclusion of an impossible object's traditional cast shadow can be a detriment to the visual illusion of the impossible object itself and is most likely the reason why, in previous impossible object modeling research, they are discarded from the final rendered images.

5.4 Experiment 2: Depth Perception in Impossible Objects

Impossible objects are created by exploiting depth inconsistencies that are difficult for our visual system to correctly identify within a two-dimensional image, whilst the inclusion of cast shadows within an image has been shown to convey useful spatial and positional information [50]. Experiment 2 is designed to investigate what effect cast shadows have on an observer's ability to correctly determine depth order within the structure of an object. In theory, the inclusion of cast shadows should aid observers to identify the correct depth order. Our aim is to test whether this also holds true for impossible objects.

5.4.1 Experiment Design

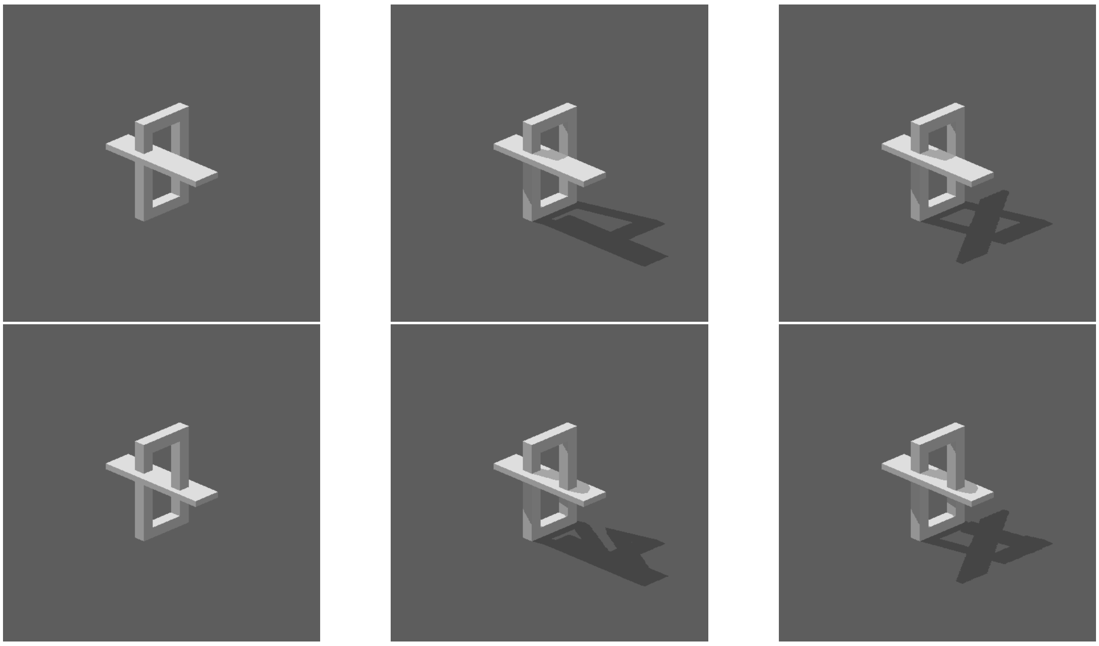


Figure 5.7: Experiment 2 Stimulus Examples. (*top*) Possible object example with no cast shadow, traditional cast shadow and copycat cast shadow. (*bottom*) Impossible object example with no cast shadow, traditional cast shadow and copycat cast shadow.

The design of this experiment is based on one used by Freud et al. [26], where fixation markers are placed on the surface on an object, and participants were tasked with choosing which marker was closer in depth. Whilst their experiment was done using two-dimensional line drawings of impossible figures we constructed our stimulus using three dimensional rendered objects.

This allowed us to test the objects under three different lighting conditions, no shadow, traditional cast shadow (where impossible objects would cast deformed shadows), and using copycat cast shadows created using our algorithm.

The stimulus consisted of 15 possible and 15 impossible objects selected randomly from our object pool, each rendered under the three different lighting conditions. On the surface of the object, two fixation markers were rendered at positions around the area of impossibility, similar to the set-up used by Freud et al. [26]. Rather than using different color fixation markers, we chose to use two different shapes, one circle one triangle. We chose this as we did not want the cases where the object cast a shadow upon the fixation marker to affect the color of the marker. For each image, during the experiment, the placement of the fixation markers would be randomized between the two positions in an effort to reduce any bias the shape of the marker may cause. Figure 5.7 demonstrates one of the stimulus objects used and all its possible image combinations. In total, our stimulus image pool consisted of 180 unique images.

During the experiment procedure subjects would be presented with a series of 90 unique images randomly selected from our stimulus pool, 45 containing possible objects and 45 containing impossible objects. Each subject would see all variations of lighting conditions, no shadow, traditional shadow, and copycat shadow. The order of the images was also randomized to avoid any bias. Subjects were tasked with choosing which of the two fixation markers appeared closer in depth. Indicating their choice by pressing one of two keys on the keyboard. Following the process described by Freud et al. [26], each stimulus image was preceded by a fixation marker displayed in the center of the screen for 1000ms. The stimulus image was then displayed on the screen until the subject had indicated a response at which point the stimulus image would disappear and a Gaussian mask would be displayed for 500ms before proceeding to the next stimulus. The experiment set-up was created using the OpenSesame [51] program and performed in the lab under the same conditions for each participant. A visual representation of the experiment

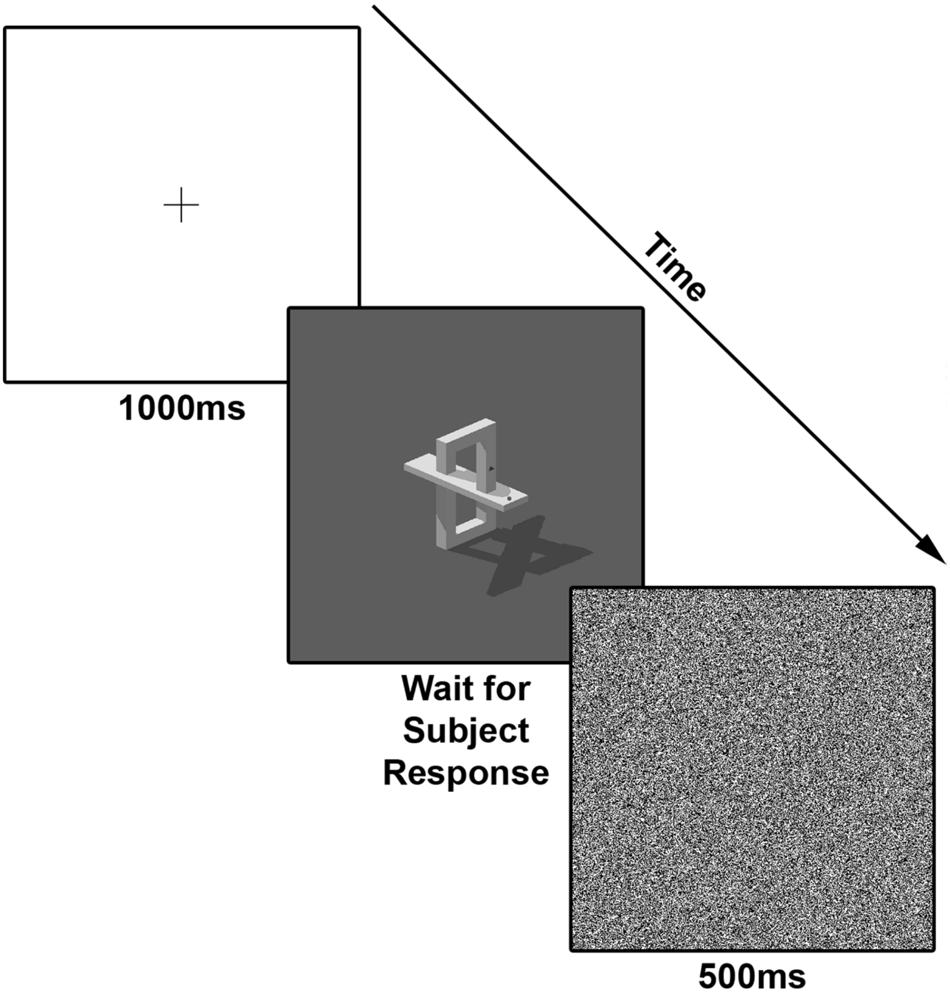
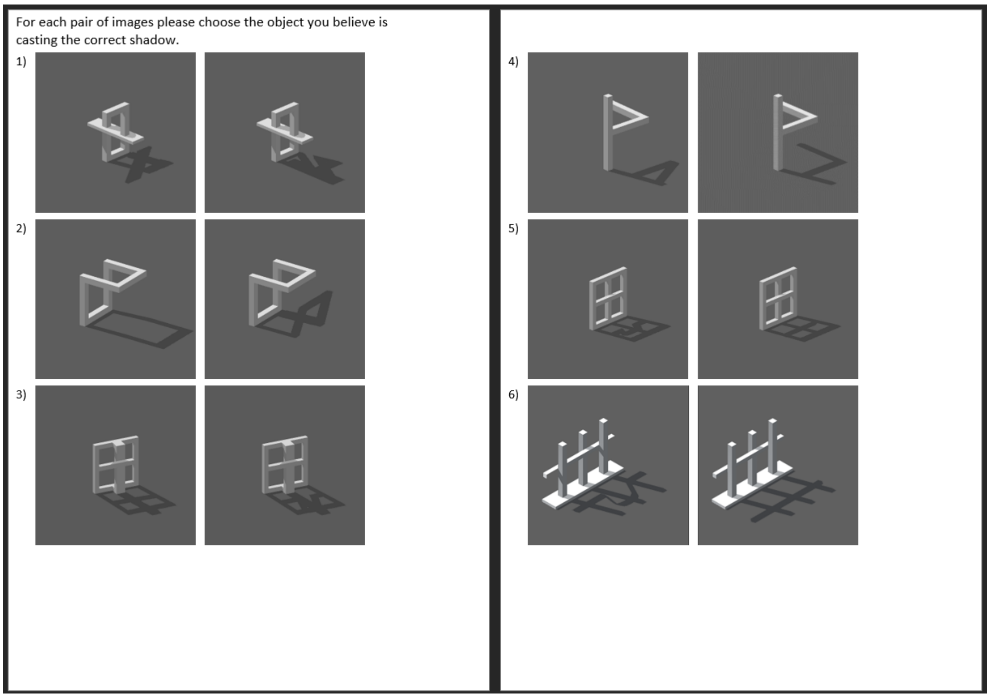


Figure 5.8: Experiment 2 design. Each stimulus image is preceded by a fixation marker in the center of the screen which is displayed for 1000ms. The stimulus image will remain on screen until the participant submits their answer, at which point the stimulus is removed and a Gaussian mask is displayed for 500ms. The process then repeats, this design is the same one used by Freud et al. [26].

process is shown in Figure 5.8. Subjects were not specifically told to base their judgement on the cast shadows, they were also not told about the difference between possible or impossible objects to avoid any bias this may cause.



For each pair of images please choose the object you believe is casting the correct shadow.

Figure 5.9: An example of the handout given after a subject has completed the experiment. Subjects were asked to choose which image in each of the pairs was casting the correct shadow. The purpose of this test was to gather a more subjective interpretation of how participants viewed the different cast shadows.

At the end of each experiment, participants were given a handout showing five possible and five impossible objects, each rendered with their traditional cast shadow and their copycat cast shadow. An example of this is shown in Figure 5.9. They were asked to choose which of the two versions of the objects they perceived to be casting the correct shadow. Our aim in this short test was to get a subjective measure of which shadow version observers felt was correct. Subjects were not informed about the difference between possible or impossible objects, or about how each of the cast shadows were created to avoid any bias this may cause.

5.4.2 Participants

Thirteen undergraduate students from the university took part in the experiment. Ten males and three females, all aged between 18 and 25 with normal or corrected to normal vision. All participants were naive to the purpose of the experiment with no mention of impossible objects or shadows given in the experiment description. At the beginning of the experiment, participants were given instructions that they would see a series of objects, each one with two fixation markers rendered on the surface of the object. They were tasked with choosing which marker (circle or triangle) appeared closer in depth by pressing the corresponding key on the keyboard. Each subject was shown 5 random stimulus images at the beginning of their trial that were not collected in the results, to ensure they understood the experiment process.

5.4.3 Results

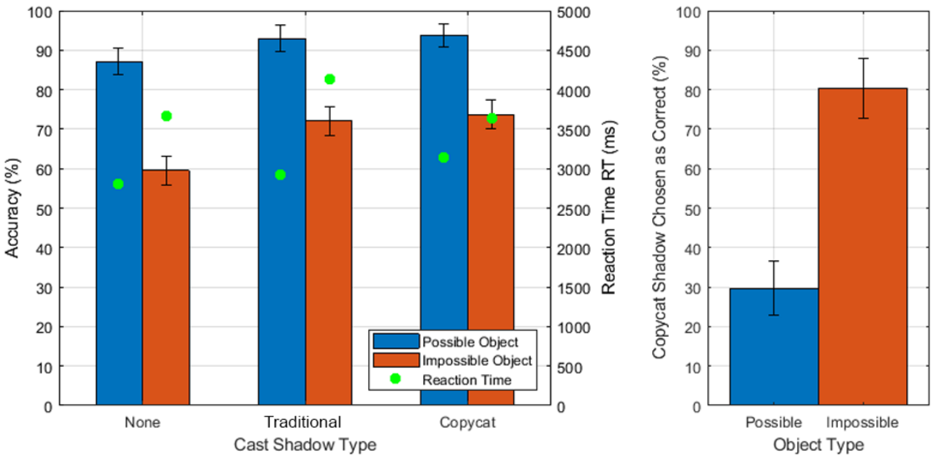


Figure 5.10: Experiment 2 Results. (*left*) Results of the timed experiment. The graph shows accuracy (%) and reaction time (ms) of participants in choosing the correct depth ordering of fixation points. Accuracy is greater for possible objects across all shadow types. When shadows are introduced there is an increase in accuracy for impossible objects, indicating that shadows can aid in the understanding of depth ordering within impossible object structures. However, there is no significant difference between traditional and copycat shadows. (*right*) Results of subjective test. Participants chose the copycat shadow as the correct shadow for impossible objects a significant amount of the time, suggesting they appear more visually appropriate than the traditional deformed shadow.

For each participant the accuracy of their ability to correctly choose the fixation marker that was closer in-depth war recorder, along with their reaction times (RT). The results of this experiment are shown in Figure 5.10. From the results, we see that subjects performed better for possible objects compared to impossible objects. Across each of the categories (no shadow, traditional shadow, and copycat shadow) higher accuracy is seen for possible objects each time (averaging around 90%) along with faster reaction times. Subject accuracy decreased for impossible objects (averaging around 67%) along with slower reaction times.

When either form of cast shadow is introduced we see an increase in accuracy for both possible and impossible objects. A larger increase is seen for impossible objects (an increase of around 14%) compared to possible objects (an increase of around 6%). We suspect this is due to the already high performance when viewing possible objects. In order to test the significance of cast shadows on the accuracy, we performed two-factor ANOVA. Resulting in a p value of $< 10^{-7}$, indicating a significant effect of the cast shadows. Object type also demonstrated a significant effect ($p < 10^{-4}$), there was no interaction effect between the two factors suggesting that cast shadows showed an improvement for both possible and impossible objects.

Although we see improved reaction times for impossible objects with copycat shadows we cannot draw a strong enough conclusion from this as results from our ANOVA test were not significant enough. The results from our subjective test, shown in Figure 5.10, suggest that subjects found the images containing the copycat cast shadows to appear more natural than those containing the object's traditional cast shadow for impossible objects. With subjects choosing the copycat shadow images as the correct image around 80% of the time, not realizing the shadows themselves are impossible.

5.4.4 Discussion

Our results show that participants perform better in both accuracy and reaction time for possible objects compared to impossible objects, this corresponds to results found by Freud Et al. [26] in their version of the experiment. We suspect the decrease in subject accuracy and reaction times for impossible objects to be due to their complex/ambiguous nature. Due to the illusions involved within these objects, the results support the conclusion that our visual system has more difficulty in interpreting and understanding the three-dimensional structure of these objects.

We also see an increase in reaction time and accuracy when cast shadows are included for possible objects, this is in line with previous research [50]. Our results also demonstrate that this is the same effect for impossible objects. Even though the illusion within these objects is based on depth ambiguities the inclusion of a cast shadow still aids observers in correctly determining and understanding depth within the structure of the object. However, from our results, we are unable to draw a conclusion as to whether one form of shadow performs better than the other. Although our subjective test indicates that copycat shadows perform better for impossible objects than their normal cast shadows this same trend is not as evident in our timed results. We suspect that the type of cast shadow has less impact within our timed test as the focus within this test was around the point of impossibility in the structure of the object. Within this focus area, the difference between the two types of shadow is minimal, much of the difference between traditional shadow and copycat cast shadow is seen in the shadow cast on the ground plane which may have been ignored for this test.

5.5 Experiment 3: Impossible Object Shadow Correspondence

The results from experiment 2 identified that the inclusion of a cast shadow improves our ability to correctly determine depth ordering within objects, including impossible objects. However, there was no significant difference found between the impossible object's traditional cast shadow and its copycat cast shadow. We suspect this is down to the shape of the cast shadow being mostly ignored. To identify which version of the cast shadow (traditional or copycat) is found to be the most visually appropriate experiment 3 was designed to specifically target the shape of the object and the corresponding shape of its cast shadow.

5.5.1 Experiment Design

For experiment 3 we changed our design from the previous experiment to be less focused on the depth ordering within the object and rather focus

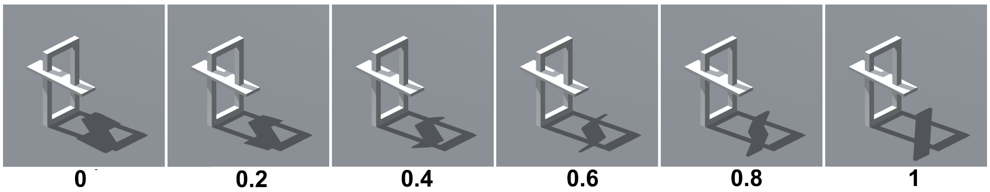


Figure 5.11: Experiment 3 stimulus example demonstrating the copycat strength effect. 0 being the objects traditional cast shadow and 1 being the full copycat shadow. Intervals between these two shadows were created by varying the amount of projection applied using our copycat shadow algorithm.

on how the shape of the objects corresponds to the shape of its cast shadow. Two fixation dots were rendered in the image, one on the surface of the object and one at a corresponding location in the object's shadow. Observers were tasked with solving the shadow correspondence problem, in other words choosing which fixation dot in the cast shadow corresponded correctly to the dot on the object's surface.

The stimulus consisted of ten possible and ten impossible objects randomly selected from our object pool. Each object would be rendered using our copycat shadow algorithm. To create the full set of stimulus we varied the strength of our copycat shadow algorithm to create a range of different shadows. To achieve the variation in copycat strength the amount of projection applied to the object by our copycat shadow algorithm was altered. At a projection strength of 0, our copycat algorithm would use the full three-dimensional geometry of the object, the resulting cast shadow would be the object's traditional cast shadow (as if rendered with conventional methods). At a projection strength of 1, the object would be fully projected onto the planar surface using our copycat algorithm, the resulting shadow would be a full copycat shadow. We varied the strength of our algorithm between 0 and 1 at intervals of 0.2, an example of this range is shown in Figure 5.11. At the intervals in between, the geometry of the object would be somewhere in between its full three-dimensional structure and projected flat onto the projection plane, resulting in a cast shadow that is somewhere between being a traditional cast shadow and a copycat cast shadow.

On each object, a fixation marker is rendered on the surface of the object with a corresponding marker rendered on the ground within the objects cast

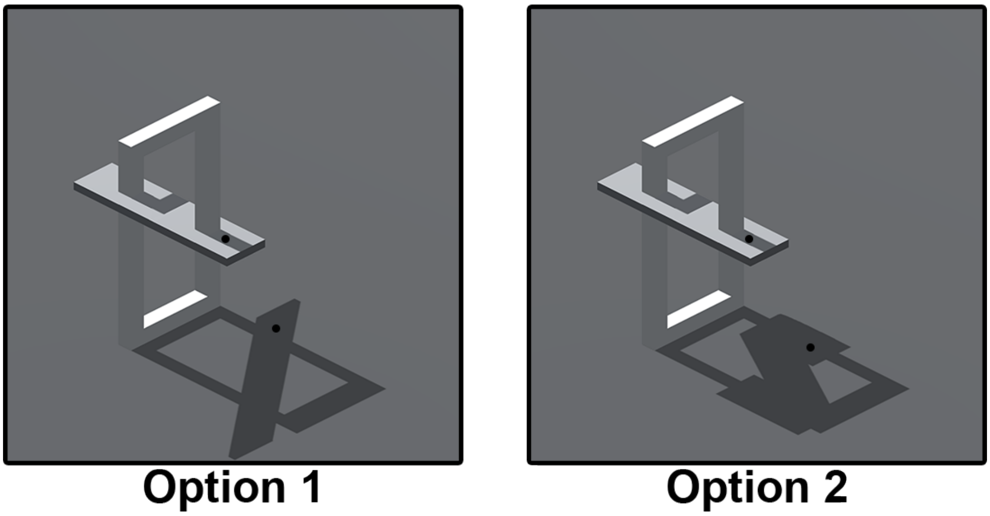
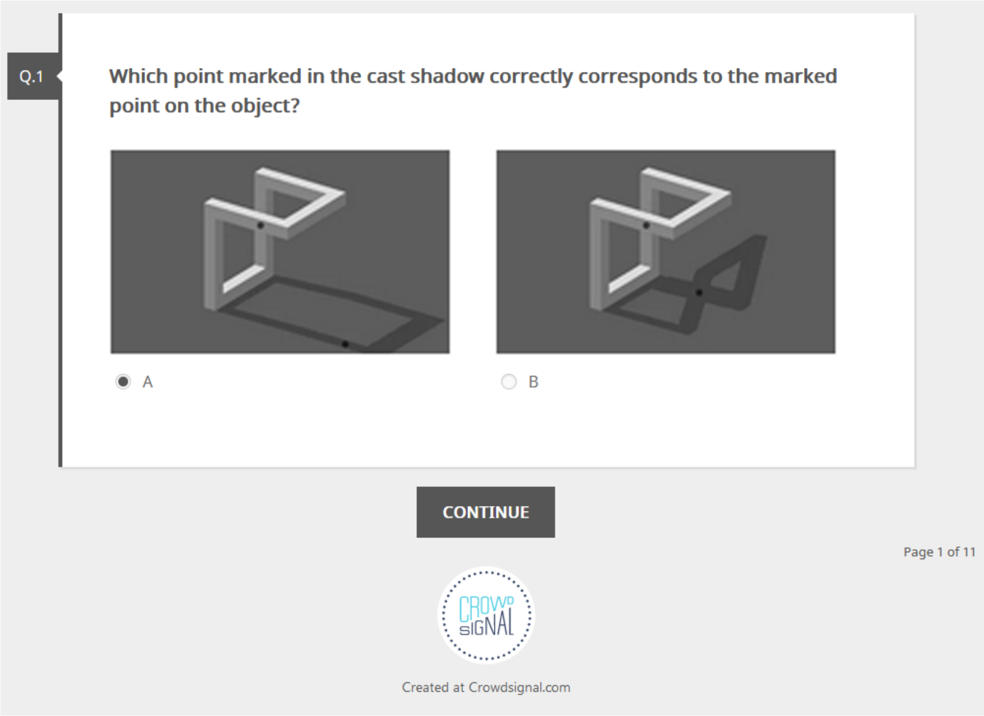


Figure 5.12: Example stimulus question. A pair of identical objects would be displayed with one object containing a copycat cast shadow and the other a traditional cast shadow.

shadow. Creating a total of 120 unique stimulus images. Subjects would be presented with a pair of stimulus images side by side, each containing the same object. One image would contain the full copycat cast shadow (projection strength 1) the other casting a shadow that is not a full copycat shadow (projection strength <1), see figure 5.12. Subjects were tasked with choosing which image contained the correct correspondence between the marker on the surface of the object and the marker in the object's shadow. The position and the order of the images were randomized to avoid any learning bias.

The experiment was held online using a survey hosting website, an example screen from our experiment is shown in Figure 5.13. Each subject was shown a block of ten random image pairs from our stimulus pool. We limited each experiment to a block of ten images to keep the experiment time down and encourage more participants. There was no time restriction, allowing participants to view the images for as long as they desired, they could however not return to previous images once an answer had been submitted. There was no explanation of impossible objects or copycat cast shadows in the experiment description to avid priming the subject's responses.



Q.1

Which point marked in the cast shadow correctly corresponds to the marked point on the object?

● A

○ B

Figure 5.13: Screenshot of experiment 3 held online. Participants were shown a pair of objects and asked which of the images contained the correct correspondence between the fixation marker on the object and in its cast shadow. Participants would indicate their choice by clicking one of the radio buttons. They were not allowed to proceed to the next set of stimulus until an answer had been submitted.

5.5.2 Participants

Participants were recruited online from a subreddit for hosting and completing surveys, the same way as experiment 1. We again restricted participants allowing them to only complete the experiment once. We also discarded results where the participant did not complete the survey and those whose completion time fell far outside the average of 3 minutes and 13 seconds, in an effort to eliminate as much unreliable data as possible. In total 75 unique subjects participated. Subjects consisted of 40 male and 35 female participants, aged between 21 and 56.

5.5.3 Results

For each participant the times they chose the image containing the full copycat shadow were recorded. The results of this experiment are shown in Figure 5.14. From these results, we see that for impossible objects subjects chose the image containing the full copycat shadow more often than any other form of cast shadow (an overall average of around 75%) while for possible

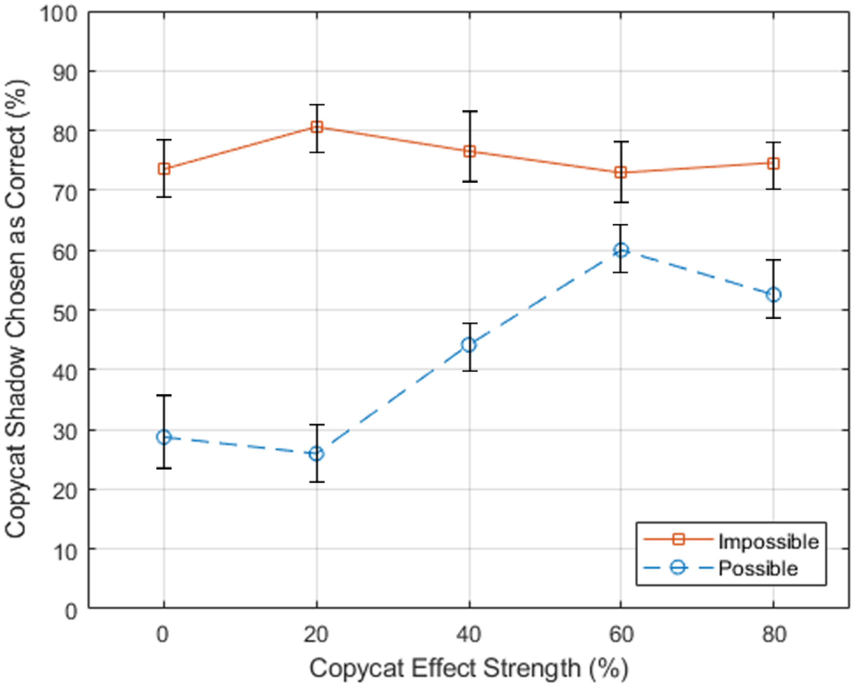


Figure 5.14: Experiment 3 Results. The graph shows the percentage of times participants chose the copycat shadow as the correct shadow correspondence. For impossible objects, subjects chose the full copycat shadow as the correct correspondence when compared to all other forms of shadow. This suggests that the copycat shadow appears the most visually appropriate for these impossible objects, any visible deformation within the cast shadows may cause them to appear unnatural.

objects this decreased (around 45%). This result is similar to that seen during our subjective test at the end of experiment 2. Performing two factor ANOVA on these results showed a significant effect ($p < 10^{-3}$) between possible and impossible objects. Suggesting that for impossible objects participants found the full copycat shadow to the most visually appropriate shadow in all cases, this did not change across the varying copycat shadow strengths as there was no significant effect found.

5.5.4 Discussion

Our results show that for possible objects subjects prefer the cast shadow that is a closer representation to the object's traditional shadow (lower copycat strength), this may be down to our ability as viewers to correctly understand the structure of the possible object. As viewers are able to understand the structure of the object they are also able to understand the shape of the shadow it would be expected to cast which is why we see a preference for those shadows that are closer to the object's traditional shadow. Our results show the opposite is true however for impossible objects, we see subjects choosing the copycat shadow across all cases. This suggests that due to the ambiguous nature of the impossible objects, subjects may not be able to fully understand its structure making the shadow correspondence problem more difficult. In these cases, the visual system may be relying on other cues to solve this problem relying more on the similarity between the visible shape of the objects and the shape of the copycat shadow. This theory supports the findings by Casati [8] that artists make use of copycat cast shadows to simplify the correspondence problem, relying more on shape matching cues to economically solve the correspondence problem.

It is also interesting to note that copycat shadows are inherently impossible, to achieve copycat shadows in the real world light would have to bend in strange ways. Even so, our results demonstrate that observers prefer these copycat shadows for impossible cases. Suggesting that although shadows contain useful spatial information our visual system can be insensitive to these impossible shadows, as demonstrated by Mamassian [49]. This suggests that for impossible objects our copycat shadow algorithm may be an improvement over the object's traditional cast shadow, conveying useful spatial cues whilst maintaining a visually appropriate shape.

5.6 Experiment 4: Copycat Shadow Algorithm Light Range

To evaluate the effectiveness of our copycat shadow algorithm we performed an experiment to help determine whether the copycat shadows generated using our algorithm appeared more natural than the traditional cast shadows generated by these impossible objects. We aimed to build upon the results gathered in experiment 3 and explore the effectiveness of our copycat algorithm under a range of light directions.

5.6.1 Experiment Design

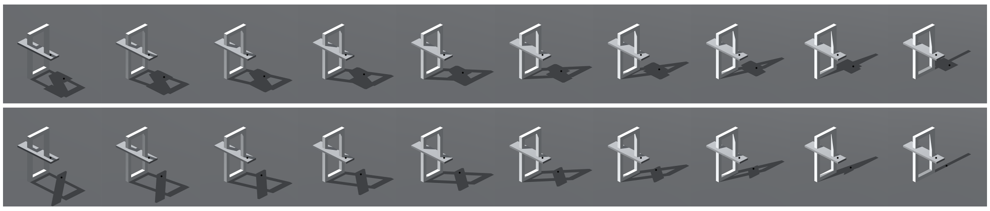


Figure 5.15: Experiment 4 Stimulus Examples. Objects were rendered under a 90-degree light range (*left 0 - right 90*) using both their traditional shadow (*bottom*) and our copycat shadow algorithm (*top*).

We tested the relationship between object structure and shadow shape by tasking participants to solve the shadow correspondence problem, similar to experiment 3, matching an indicated point on the surface of an object to its corresponding position in the cast shadow. Two fixation dots were rendered in each image, one on the surface of the object and one at a corresponding point in the cast shadow. With participants being tasked to find the correct correspondence between the two points, as in experiment 3.

For this experiment, we wanted to explore the performance of our copycat algorithm under a range of light directions. The stimulus consisted of ten impossible objects randomly selected from our object pool. Each object was rendered with their traditional cast shadow and using our copycat shadow algorithm. Each object was rendered under a range of light directions spanning 90 degrees, separated at 10-degree intervals. Figure 5.15 shows one of the objects rendered under all light directions. Creating a total of 200 unique stimulus images. Within each image, a fixation dot would be rendered on the surface of the object with a corresponding fixation dot in the objects cast shadow.

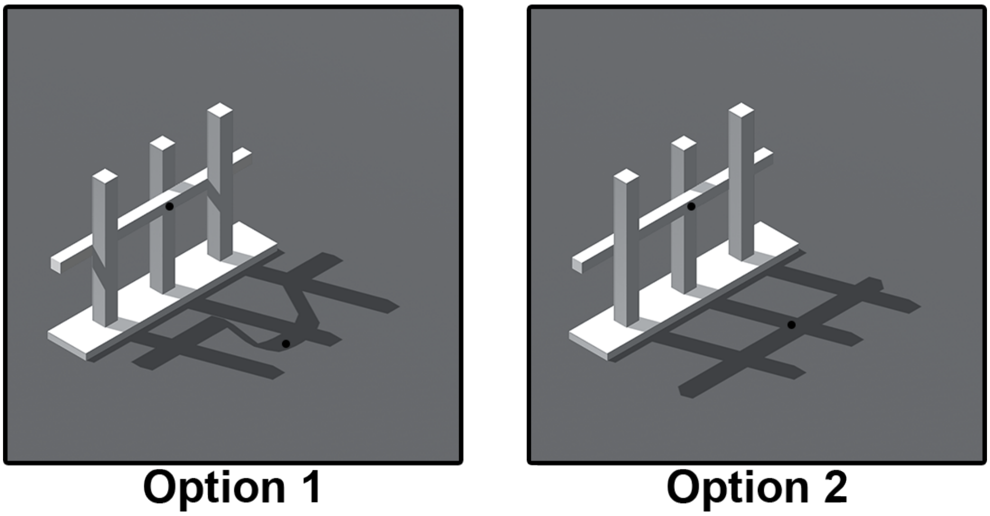
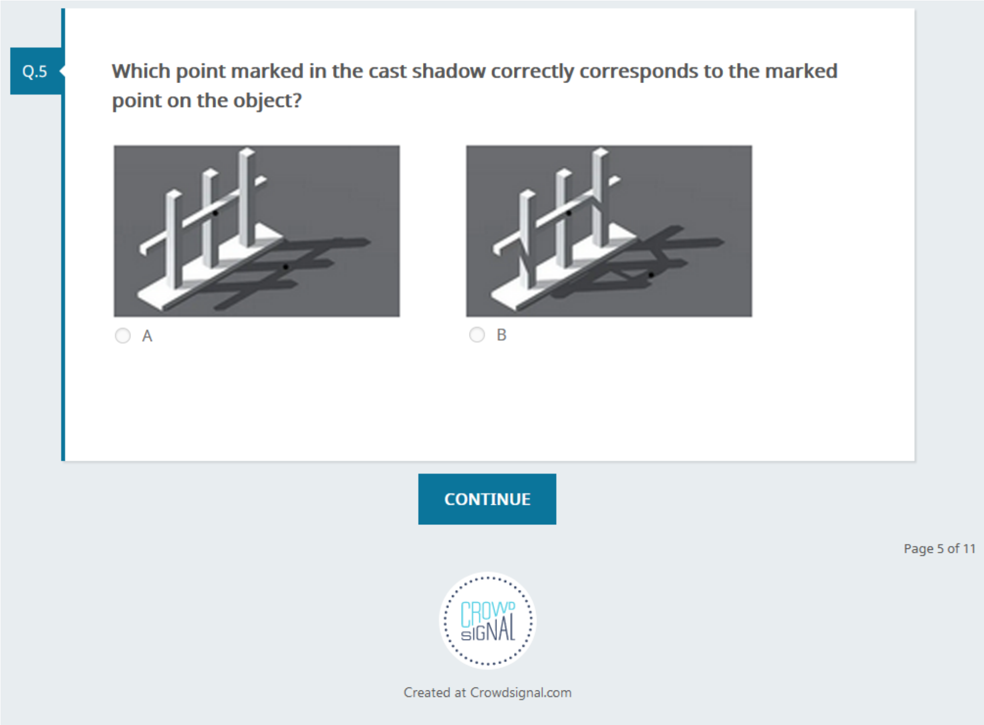


Figure 5.16: Experiment 4 sample stimulus. A pair of images containing the same object would be displayed, one casting its traditional shadow the other casting a copycat shadow generated using our algorithm. Fixation markers were rendered on the surface of the object and at the corresponding point in the cast shadow.

During the experiment subjects would be presented with a pair of images side by side, each containing the same object. One image would contain the object alongside its traditional cast shadow the other containing the object alongside its copycat shadow. Both shadows would be cast using the same light direction, Figure 5.16 shows a set of example stimulus. Subjects were tasked with choosing which image contained the correct correspondence between the marker on the surface of the object and the marker in the object cast shadow. The position and order of the images were randomized to reduce any learning bias.

The experiment was held online using a survey hosting website, an example screen from our is shown in Figure 5.17. Each subject was shown a block of ten images in order to keep experiment time down and encourage more participants. Subjects would see the full range of light directions (0 to 90 degrees) along with seeing each of the ten impossible objects once. It was



Q.5 Which point marked in the cast shadow correctly corresponds to the marked point on the object?

○ A

○ B

Figure 5.17: Screenshot of experiment 4, held online. Similar to experiment 3 participants were shown a pair of images and tasked with choosing the image they felt contained the correct correspondence between the object and its cast shadow, indicating their choice using the radio buttons. Participants were not allowed to continue to the next set of stimulus until an answer had been submitted.

randomized as to what order they would see the light directions in along with what object would be seen under each light direction in an effort to reduce any bias the subjects may gain from seeing the light direction change. We also ensured that subjects would not see the same object twice in an effort to reduce any learning bias. There was no time restriction in place, allowing participants to view the images for as long as they desired. They could not return to previous images once an answer had been submitted. There was no explanation of impossible objects or copycat shadows in the experiment description so as to avoid priming the subject's responses.

5.6.2 Participants

Participants were recruited online from a subreddit for hosting and completing surveys and experiments, the same method used in experiments 1 and 3. We once again restricted participants allowing them to only complete the experiment once. We also discarded results from participants who did not complete the experiment and those whose completion time fell far outside the average of 3 minutes and 23 seconds, in an effort to eliminate as much unreliable data as possible. In total 106 unique subjects participated. Subjects consisted of 58 male and 48 female, aged between 20 and 47.

5.6.3 Results

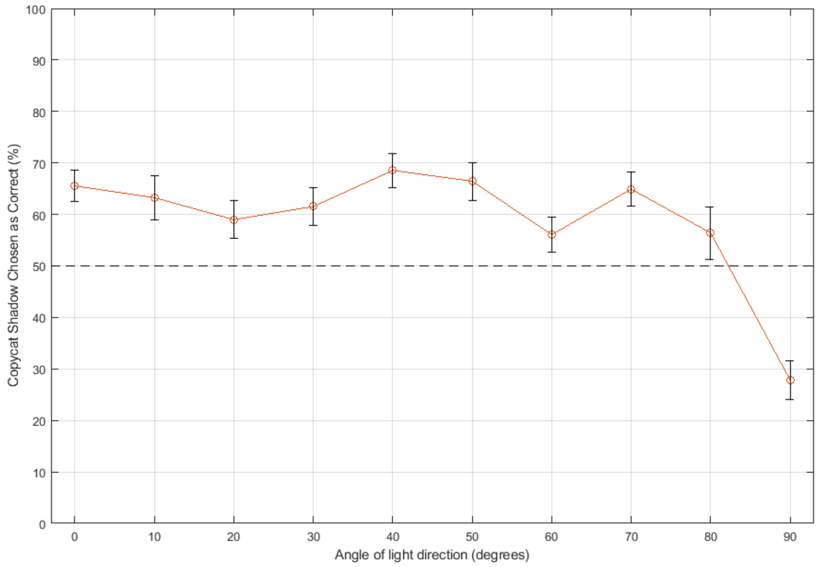


Figure 5.18: Experiment 4 Results. The graph shows the percentage of times the copycat cast shadow was chosen as the correct correspondence compared to the object's traditional cast shadow. The copycat shadow is preferred across the majority of light directions indicating our copycat algorithms success in being able to work effectively under a range of light directions. There is a decline as the light direction approaches 90 degrees, we predict this is due to the cast shadow generated by our algorithm appearing relatively flat at these light directions which may not correspond well to the more complex three-dimensional casting object.

For each participant, the number of times they chose the image containing the copycat cast shadow was recorded. Results for this experiment are shown in Figure 5.18. Our results show that subjects chose the copycat cast shadow as the correct correspondence across all light directions except 90 degrees. For each of the light directions (0 - 80 degrees) where the copycat cast shadow is preferred we see an average of around 62%, this drops however at 90 degrees to 27%. Performing a two factor ANOVA test on the results show that cast shadow type has a significant effect ($p < 10^{-7}$), light direction is also shown to have an effect ($p < 10^{-2}$).

5.6.4 Discussion

On the basis of our results, we can infer that subjects found the shadows generated using our copycat algorithm to be more visually appropriate than the traditional cast shadow for impossible objects. This further supports our findings from experiments 2 and 3. We suspect this preference for the copycat shadow is due to the difficulty our visual system has in correctly understanding the structure of these impossible objects, making it much more difficult to correctly solve the shadow correspondence problem due to the ambiguous nature of the object. A copycat shadow may have an advantage here compared to traditional cast shadows due to their similarity in shape to the visible shape of the casting object. As both, the object and shadow appear similar in shape it may be simpler for our visual system to identify the correspondence between them. When viewing the traditional cast shadows the deformation present in the shadows does not correspond to the apparent orthogonality of the impossible object, making them appear unnatural to the observer.

As the light direction approaches 90 degrees we see a decrease in the number of times copycat shadows generated using our algorithm is chosen as the correct correspondence, this supports some of the findings we discussed previously in our work. We suspect this is down to the copycat shadow losing some of its detail at these angles, the shadows generated using our copycat algorithm can appear flat when compared to the three-dimensional appearance of the casting object.

The results gathered in this experiment suggest that shadows generated using our copycat algorithm can provide convincing, visually appropriate cast shadows for impossible objects. Where previously these objects were rendered with no cast shadows, we believe these results show are copycat algorithm can produce cast shadows that are an improvement over the objects traditional cast shadow. Providing a solution for rendering impossible objects in more complex scenes where the lack of cast shadow would appear unnatural.

5.7 Summary and Discussion

We have conducted a series of visual perception experiments demonstrating the potential of copycat cast shadows. In particular their use as a solution for producing a convincing, visually appropriate cast shadow for three-dimensional impossible objects, where previously the most common option was to exclude shadows for these objects completely. Results from experiment 1 demonstrate that using the traditional shadow for these impossible objects can appear strange due to the unnatural structure of the object causing anomalies in its cast shadow. We have also briefly explored how our visual perception of these objects changes with the inclusion of a cast shadow. Previous visual experiments performed using impossible figures used simple two-dimensional line drawings. Our results show that the inclusion of a cast shadow can aid in understanding the depth ambiguities used in creating these impossible objects, enabling us to correctly determine depth within these three-dimensional impossible objects. Results for experiment 4 also demonstrate that shadows created using our copycat shadow algorithm appear more visually appropriate for impossible objects than their traditional cast shadows. This is true across a range of light directions, showing the strength of our copycat shadow algorithm but also identifying areas of weakness. Using the results from this experiment we can go back and look at how to improve the shadows generated by our algorithm under those problematic light directions.

Through our series of experiments, we have explored how our copycat shadow algorithm performs as a substitute solution for casting shadows of impossible objects. Due to our ability to model three dimensional versions of impossible objects we have been able to conduct experiments that have not been possible before. Where previous methods used line-drawn two-dimensional objects [14, 23, 25] we have used fully three-dimensional objects, allowing us to examine the objects in novel ways such as using different view directions or alongside a rendered cast shadow. This method of using three-dimensional impossible objects opens up interesting avenues for future work. Carrasco and Seamon [5] explored how observers interpreted the complexity and perceived impossibility of these objects. From our experiments and previous research into cast shadows [35, 41, 82] we have seen shadows have a positive effect on an observers ability to understand the composition and shape of an object. It would be interesting to investigate whether the inclusion of cast shadows alongside these impossible objects alters an observer's perception of the impossible object, changing how complex or impossible they appear.

We conducted many of our experiments online to enable us to reach a wider range of participants to conduct a large number of experiments. Using this online method however we lose control over certain experiment conditions, as an example, there is no way to ensure all participants are viewing the same size images as they may all be using different screens and devices. This may have had an unseen impact on our results, as such, improvements could be made by conducting similar experiments under a more controlled in-person setting. Using this data to support the results gathered from our online experiments.

Results have shown however that for possible objects the copycat shadows used here do not appear preferable over the object's traditional cast shadow. We believe this is down to the objects used in these tests being isolated and in contact with the ground. Under these conditions, the correspondence problem is made easier due to the object and shadow being connected. A direction for future work may be to investigate the effect of copycat shadows for possible objects under different situations. Suspending the object in the air such that there is a disconnection between the object and shadow or using a scene containing multiple objects. Both cases would increase the difficulty of the correspondence problem and may make copycat shadows more effective. These are also similar to the situations where copycat shadows are used in pictorial art [8].

Chapter 6

Conclusions and Future Work

While impossible figures are commonly seen as two-dimensional images they are rarely seen as three-dimensional models. The main reason for this is that the section of the object that creates the illusion of impossibility can be difficult to replicate in three dimensions. Some progress has been made in presenting modeling methods to create three-dimensional versions of these impossible figures. However even when images of these three-dimensional models are rendered they are often presented using simple shading and lighting, most notably they often lack any form of cast shadows. The modeling methods used to simulate the impossibility illusion in these objects create ambiguities in the object cast shadows and as such, cast shadows are often excluded from the rendered image completely. We believe this lack of a visually appropriate cast shadow for these impossible objects is restricting their use in more complex three-dimensional scenes.

In this work, we have presented methods of modeling and rendering three-dimensional impossible objects with convincing cast shadows. By creating a method for generating visually appropriate cast shadows and enabling them to be rendered convincingly in three-dimensional scenes we hope to expand the use of impossible objects in computer graphics. In this chapter, we summarise the contributions of our work and highlight possible future directions.

6.1 Summary

In chapter 3 we introduced a modeling method for three-dimensional impossible objects, based on transparency. Beginning with a regular three-dimensional model our modeling method can simulate the illusion of impossibility through manipulations to the surface transparency of the object. Demonstrating its ability to simulate a range of impossibility types and render novel viewpoints of the objects.

Chapter 4 presented our screen space algorithm for generating copycat cast shadows automatically, where previously manual methods had been used to create this copycatting effect. Using our algorithm it is possible to automatically generate copycat shadows for objects under a wide range of light and view directions. As our algorithm operates in screen space it does not disrupt the existing rendering pipeline, making it useable alongside existing effects such as perspective projection and ambient occlusion. Using these copycat shadows we are able to render three-dimensional impossible objects with visually appropriate cast shadows, rendering these objects convincingly in a three-dimensional scene.

In Chapter 5 explore the visual perception of these three-dimensional impossible object and copycat cast shadows by conducting a series of visual perception experiments. The results gathered demonstrate the benefits a cast shadow can have on our ability to accurately perceive depth within a two-dimensional image, something commonly known for possible objects, and from our results is also the case for impossible objects. We have explored the use of copycat shadows as a solution to casting visually appropriate shadows for impossible objects. Results from our experiments show that subjects found copycat shadows to be more compelling than the impossible objects traditional cast shadow, strengthening the case for our copycat algorithm and measuring its effectiveness under a range of light directions.

6.2 Revisiting the Thesis Objectives

At the beginning of this thesis we outlined the set of aims that follows:

- Explore the existing uses of impossible objects within computer graphics and pictorial art. Investigate the existing methods that have been implemented to model and render these impossible objects.

- Investigate the role of cast shadows in object perception.

- Design a new impossible object modeling algorithm based on transparency.

- Design an algorithm for automatically generating copycat cast shadows. Where previously these shadows are created using manual post-processing.

- Implement our algorithms using existing tools so that they are easy to use without the need for any dedicated rendering environment.

- Analyse the effect cast shadows have on a viewer's perception of impossible objects through a series of visual perception experiments.

We believe the first and second objectives have been successfully completed in Chapter 2. In this chapter, we began by investigating the history of impossible figures in pictorial art. Along with reviewing the existing methods of impossible object modeling. Creating a taxonomy to highlight their implementation and effectiveness at modeling various types of impossible figures. From this taxonomy, we identified the weaknesses of previous methods and attempted to address those using our own algorithm. Chapter 2 also investigates the role cast shadows play in our perception of objects, along with presenting evidence to the use of impossible shadows in pictorial art. Where we learned that although cast shadows contain a variety of useful information, the human visual system can be particularly insensitive to inconsistent or impossible illuminations. Evidenced in both pictorial art and perceptual studies.

The third objective has been achieved in Chapter 3, where we present our transparency based modeling algorithm. We create a new method of modeling three-dimensional impossible figures through the manipulation of the surface transparency of the object. We demonstrate its ability to handle multiple types of impossible objects along with rendering objects under different view directions. Comparing these results with existing modeling methods.

The fourth objective was explored in Chapter 4, where we present our copycat cast shadow algorithm. We took inspiration from the impossible shadows present in pictorial art and existing visual perception research to design an algorithm that can automatically generate copycat cast shadows. With minimal user input, we have demonstrated how our algorithm works under a range of view and light directions.

Objective five has been addressed in both Chapters 3 and 4. When designing our algorithms we implemented them within the Unity engine. Demonstrating their use with existing features such as textures, ambient occlusion, perspective, and orthogonal cameras.

The sixth objective we set was to analyse the effect of cast shadows in combination with impossible objects. In Chapter 5 we present results from a series of visual perception experiments. Beginning by looking at what effect traditional cast shadows on participant's ability to correctly identify impossible objects among a set of distractors. Results indicate the inclusion of traditional cast shadows greatly aids participants in locating the target object, we suspect this is due to the presence of hidden deformations in the structure of the object being revealed in its cast shadow. This supports our case that an alternate solution is needed to cast convincing shadows for these objects. We then examined how traditional cast shadows compared to our copycat cast shadows for tasks including depth ordering and shadow correspondence. With results showing that for impossible objects our copycat cast shadows perform better. Our final experiment aims to examine the performance of our copycat shadows under a range of light sources, attempting to identify where our copycat shadows succeed and where they fail. The results of this experiment can be used to further improve our algorithm. Although we may have begun to meet this objective the subject of cast shadow and impossible object perception is vast. What we do believe however is that by presenting methods of modeling and rendering these impossible objects, alongside such features as cast shadows, opens up new avenues for future research, where previously these rendering options were not available.

6.3 Limitations

We identify the three main limitations of this work. The first main limitation is within the acceptable light ranges of our copycat shadow algorithm. Although we have shown that our algorithm renders shadows under a wide range of light directions, we have also identified light directions were our algorithm does not perform well. We have demonstrated this through a measure applied to estimate the performance of our algorithm along with results from our visual perception experiments. As such there are areas that require improvement, light directions under which our algorithm does not produce convincing copycat shadows.

The second limitation is within the creation of our projection plane for the copycat shadow algorithm. We identify how to choose a projection plane based on the object's surface location and normal. However, if the object is cylindrical this becomes more difficult and requires more manual input. There is also the issue that arises when changing the projection plane. Currently, its implementation is static, this restricts the range of light and view directions where our copycat algorithm performs successfully. By implementing a method to allow the manipulation of the projection plane may be a solution to increasing the performance of our algorithm.

The third main limitation is with our method of conducting our visual perception experiments. By holding the experiments online we are able to reach a wider audience, easily recruiting many participants for each test. The limitation, however, is in the control that is lost over the experiment procedure. Under lab conditions, aspects such as screen size, image size, and brightness can all be controlled or measured. We sacrifice this ability when conducting our experiment online, although we try to implement measures to strengthen the integrity of our data there will be some unseen impact on our results. Even with these limitations, we believe the results from our experiments provide some initial insights into the effect cast shadows have on impossible objects.

6.4 Future Work

We believe that this thesis has attained the objectives set out at the beginning. However, there are also questions left unanswered and new questions that have been raised. While our transparency based modeling algorithm has been shown to work under a range of view directions, we have identified areas where it fails to produce an impossible object. Whilst this may be due to the inherent view limitations of the structure of the object we believe that further improvements could be made to extend the range of acceptable view directions. We also identify a possible alternative method of transparency based modeling that we predict could reduce the amount of user input needed. By basing the transparency of sections of the object that overlap. A key question we believe would be interesting to explore would be how impossible models could be rendered with convincing reflections? Similar to how the hidden deformations of the model were revealed through its cast shadow, they would also be revealed in its reflection. This problem is yet to be addressed in previous research. Perhaps a similar method to our copycat algorithm could suffice, where a flattened version of the object is used for reflection, removing the hidden deformations. Although our perception experiments prove a beginning look at how cast shadows and impossible objects interact with one another there is still much to explore. We believe an interesting area would be to look at how cast shadows change our perception of impossibility. With impossible objects often being created through depth ambiguities and the proven ability of cast shadows to help viewers interpret depth information, does the inclusion of these cast shadows make the object appear less impossible to the viewer?

6.5 Conclusion

We began our research with the vision of creating the ability to render three-dimensional objects in complex scenes, under complete lighting effects such as cast shadows and shading. Much effort has been applied to achieve this goal, beginning with exploring how the human visual system interprets both cast shadows and impossible objects through an investigation of their use in pictorial art and perception research. Reflecting on our research questions set out at the beginning of our thesis, see Section (1.1.4). We believe that we have presented a valid solution for rendering visually convincing cast shadows for impossible objects, through the use of a copycat shadow method. Our algorithm has been implemented such that it has minimal impact on the user as it works using exiting tools and rendering engines. Finally, we have explored the use of these copycat shadows alongside impossible objects through a series of visual perception experiments, proving that observers find these copycat cast shadows visually convincing.

References

[1] L. Bavoil and M. Sainz, 'Screen space ambient occlusion', *NVIDIA developer information: http://developers. nvidia. com*, vol. 6, 2008 (p. 27).

[2] ——, 'Multi-layer dual-resolution screen-space ambient occlusion', in *SIGGRAPH 2009: Talks*, ACM, 2009, p. 45 (p. 27).

[3] W. L. Braje, G. E. Legge and D. Kersten, 'Invariant recognition of natural objects in the presence of shadows', *Perception*, vol. 29, no. 4, pp. 383–398, 2000 (p. 32).

[4] N. Canterakis, '3d zernike moments and zernike affine invariants for 3d image analysis and recognition', in *In 11th Scandinavian Conf. on Image Analysis*, 1999, pp. 85–93 (pp. 67, 80).

[5] M. Carrasco and J. G. Seamon, 'Priming impossible figures in the object decision test: The critical importance of perceived stimulus complexity', *Psychonomic Bulletin & Review*, vol. 3, no. 3, pp. 344–351, Sep. 1996, ISSN: 1531-5320. DOI: `10.3758/BF03210758`. [Online]. Available: `https://doi.org/10.3758/BF03210758` (p. 123).

[6] R. Casati, 'The shadow knows: A primer on the informational structure of cast shadows', *Perception*, vol. 33, no. 11, pp. 1385–1396, 2004 (p. 31).

[7] ——, 'The cognitive science of holes and cast shadows', *Trends in Cognitive Sciences*, vol. 10, no. 2, pp. 54–55, 2006, ISSN: 13646613. DOI: `10.1016/j.tics.2005.12.008` (p. 62).

[8] ——, 'The copycat solution to the shadow correspondence problem', *Perception*, vol. 37, no. 4, pp. 495–503, 2008, ISSN: 03010066. DOI: `10.1068/p5588`. [Online]. Available: `https://doi.org/10.1068/p5588` (pp. 5, 36, 60–63, 86, 93, 117, 124).

[9] U. Castiello, 'Implicit processing of shadows', *Vision Research*, vol. 41, no. 18, pp. 2305–2309, 2001 (p. 32).

[10] P. Cavanagh, ‘The artist as neuroscientist’, *Nature*, vol. 434, no. 7031, p. 301, 2005 (pp. 34, 63, 92).

[11] T. Chiba, T. Moriya and T. Takahashi, ‘An extended modeling method of optical illusion objects in general rendering environments’, in *2018 International Workshop on Advanced Image Technology (IWAIT)*, 2018, pp. 1–4 (pp. 21, 23, 56).

[12] B. Cowan, S. Khattak, B. Kapralos and A. Hogue, ‘Screen space point sampled shadows’, *2015 IEEE Games Entertainment Media Conference, GEM 2015*, pp. 1–7, 2016. DOI: 10.1109/GEM.2015.7377248 (p. 74).

[13] T. M. Cowan, ‘Organizing the properties of impossible figures’, *Perception*, vol. 6, no. 1, pp. 41–56, 1977 (p. 18).

[14] I. E. Dror, C. Ivey and C. Rogus, ‘Visual mental rotation of possible and impossible objects’, *Psychonomic Bulletin & Review*, vol. 4, no. 2, pp. 242–247, 1997 (pp. 37, 123).

[15] G. Elber, ‘Modeling (seemingly) impossible models’, *Computers and Graphics (Pergamon)*, vol. 35, no. 3, pp. 632–638, 2011, ISSN: 00978493. DOI: 10.1016/j.cag.2011.03.015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0097849311000549 (pp. 1, 4, 22, 40, 42, 43, 46, 52, 60, 77, 87, 88, 90, 93, 94).

[16] J. T. Enns and R. A. Rensink, ‘Influence of scene-based properties on visual search’, *Science*, vol. 247, no. 4943, pp. 721–723, 1990 (p. 33).

[17] B. Ernst, *Adventures with impossible figures*. Parkwest Pubns, 1986 (pp. 2, 19).

[18] M. ESCHER. (). ’waterfall’ by m. c. escher, [Online]. Available: https://www.mcescher.com/gallery/recognition-success/waterfall/ (visited on 1st May 2019) (pp. 3, 4, 55).

[19] ——, (). Official website: M. c. escher, [Online]. Available: https://www.mcescher.com/ (visited on 1st May 2019) (pp. 1, 14).

[20] H. Farid and M. J. Bravo, ‘Image forensic analyses that elude the human visual system’, in *Media forensics and security II*, International Society for Optics and Photonics, vol. 7541, 2010, p. 754 106 (p. 34).

[21] R. Fernando, ‘Percentage-closer soft shadows’, in *ACM SIGGRAPH 2005 Sketches*, ser. SIGGRAPH ’05, Los Angeles, California: ACM, 2005. DOI:

10.1145/1187112.1187153. [Online]. Available: http://doi.acm.org/10.1145/1187112.1187153 (p. 26).

[22] J. A. Ferwerda, ‘Three varieties of realism in computer graphics’, in *Human Vision and Electronic Imaging VIII*, International Society for Optics and Photonics, vol. 5007, 2003, pp. 290–298 (p. 24).

[23] E. Freud, G. Avidan and T. Ganel, ‘Holistic processing of impossible objects: Evidence from garner’s speeded-classification task’, *Vision Research*, vol. 93, pp. 10–18, 2013 (pp. 1, 93, 94, 123).

[24] E. Freud, T. Ganel and G. Avidan, ‘Representation of possible and impossible objects in the human visual cortex: Evidence from fmri adaptation’, *NeuroImage*, vol. 64, pp. 685–692, 2013, ISSN: 1053-8119. DOI: https://doi.org/10.1016/j.neuroimage.2012.08.070. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1053811912008798 (pp. 1, 37, 105).

[25] ——, ‘Impossible expectations: Fmri adaptation in the lateral occipital complex (loc) is modulated by the statistical regularities of 3d structural information’, *NeuroImage*, vol. 122, pp. 188–194, 2015, ISSN: 1053-8119. DOI: https://doi.org/10.1016/j.neuroimage.2015.07.085. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1053811915007053 (pp. 37, 123).

[26] E. Freud, T. Ganel, I. Shelef, M. D. Hammer, G. Avidan and M. Behrmann, ‘Three-Dimensional Representations of Objects in Dorsal Cortex are Dissociable from Those in Ventral Cortex’, *Cerebral Cortex*, vol. 27, no. 1, pp. 422–434, Oct. 2015, ISSN: 1047-3211. DOI: 10.1093/cercor/bhv229. eprint: http://oup.prod.sis.lan/cercor/article-pdf/27/1/422/25160213/bhv229.pdf. [Online]. Available: https://doi.org/10.1093/cercor/bhv229 (pp. 38, 94, 106–108, 111).

[27] E. Freud, B.-S. Hadad, G. Avidan and T. Ganel, ‘Evidence for similar early but not late representation of possible and impossible objects’, *Frontiers in Psychology*, vol. 6, p. 94, 2015 (p. 38).

[28] E. Freud, G. Rosenthal, T. Ganel and G. Avidan, ‘Sensitivity to object impossibility in the human visual cortex: Evidence from functional connectivity’, *Journal of Cognitive Neuroscience*, vol. 27, no. 5, pp. 1029–1043, 2015 (pp. 93, 94, 105).

[29] D. Friedman and Y. M. Cycowicz, ‘Repetition priming of possible and impossible objects from erp and behavioral perspectives’, *Psychophysiology*, vol. 43, no. 6, pp. 569–578, 2006 (p. 37).

[30] J. Gumbau, M. Chover and M. Sbert, ‘Screen space soft shadows’, 2010 (p. 27).

[31] J.-M. Hasenfratz, M. Lapierre, N. Holzschuch and F. X. Sillion, ‘A survey of real-time soft shadows algorithms’, *Computer Graphics Forum*, vol. 22, Dec. 2003. DOI: `10.1111/j.1467-8659.2003.00722.x` (p. 24).

[32] G. Hori, ‘Prototyping impossible objects with vr’, in *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*, ACM, 2018, p. 114 (p. 24).

[33] M.-K. Hu, ‘Visual pattern recognition by moment invariants’, *IRE transactions on information theory*, vol. 8, no. 2, pp. 179–187, 1962 (p. 68).

[34] Z. Huang and J. Leng, ‘Analysis of hu’s moment invariants on image scaling and rotation’, in *2010 2nd International Conference on Computer Engineering and Technology*, IEEE, vol. 7, 2010, pp. V7–476 (p. 68).

[35] G. S. Hubona, P. N. Wheeler, G. W. Shirah and M. Brandt, ‘The role of object shadows in promoting 3d visualization’, *ACM Transactions on Computer-Human Interaction*, vol. 6, no. 3, pp. 214–242, 1999 (pp. 92, 124).

[36] D. Huffman, ‘Impossible objects as nonsense sentences, machine intelligence 6’, *Machine Intelligence*, vol. 6, pp. 295–323, 1971 (p. 17).

[37] T. Imai, Y. Kanamori and J. Mitani, ‘Real-time screen-space liquid rendering with complex refractions’, *Computer Animation and Virtual Worlds*, vol. 27, no. 3-4, pp. 425–434, 2016. DOI: `10.1002/cav.1707`. eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1002/cav.1707`. [Online]. Available: `https://onlinelibrary.wiley.com/doi/abs/10.1002/cav.1707` (p. 29).

[38] T. Isenberg, B. Freudenberg, N. Halper, S. Schlechtweg and T. Strothotte, ‘A developer’s guide to silhouette algorithms for polygonal models’, *IEEE Computer Graphics and Applications*, vol. 23, no. 4, pp. 28–37, Jul. 2003, ISSN: 0272-1716. DOI: `10.1109/MCG.2003.1210862` (p. 29).

[39] J. Jacobson and S. Werner, ‘Why cast shadows are expendable: Insensitivity of human observers and the inherent ambiguity of cast shadows in

pictorial art', *Perception*, vol. 33, no. 11, pp. 1369–1383, 2004 (pp. 34, 62, 92).

[40] J. Jimenez, V. Sundstedt and D. Gutierrez, 'Screen-space perceptual rendering of human skin', *ACM Trans. Appl. Percept.*, vol. 6, no. 4, 23:1–23:15, Oct. 2009, ISSN: 1544-3558. DOI: 10.1145/1609967.1609970. [Online]. Available: http://doi.acm.org/10.1145/1609967.1609970 (p. 29).

[41] D. Kersten, P. Mamassian and D. C. Knill, 'Moving cast shadows and the perception of relative depth', 1994 (pp. 31, 124).

[42] C. W. Khoh and P. Kovesi. (1999). Animating impossible objects, [Online]. Available: https://www.peterkovesi.com/projects/impossible/impossible.html (pp. 20, 23).

[43] A. Khotanzad and Y. H. Hong, 'Invariant image recognition by zernike moments', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 5, pp. 489–497, 1990, ISSN: 0162-8828. DOI: 10.1109/34.55109 (p. 67).

[44] D. C. Knill, P. Mamassian and D. Kersten, 'Geometry of shadows', *JOSA A*, vol. 14, no. 12, pp. 3216–3232, 1997 (p. 30).

[45] Z. Kulpa, 'Putting order in the impossible', *Perception*, vol. 16, no. 2, pp. 201–214, 1987 (pp. 2, 18, 21).

[46] J. Lansdown and S. Schofield, 'Expressive rendering: A review of non-photorealistic techniques', *IEEE Computer Graphics and Applications*, vol. 15, no. 3, pp. 29–37, May 1995, ISSN: 0272-1716. DOI: 10.1109/38.376610 (p. 29).

[47] B. J. Loos and P.-P. Sloan, 'Volumetric obscurance', in *Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games*, ACM, 2010, pp. 151–156 (p. 27).

[48] M. Magdics, C. Sauvaget, R. J. Garcia and M. Sbert, 'Post-processing npr effects for video games', in *Proceedings of the 12th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*, ser. VRCAI '13, Hong Kong, Hong Kong: ACM, 2013, pp. 147–156, ISBN: 978-1-4503-2590-5. DOI: 10.1145/2534329.2534348. [Online]. Available: http://doi.acm.org/10.1145/2534329.2534348 (p. 29).

[49] P. Mamassian, ‘Impossible shadows and the shadow correspondence problem’, *Perception*, vol. 33, no. 11, pp. 1279–1290, 2004 (pp. 36, 62, 92, 105, 117).

[50] P. Mamassian, D. C. Knill and D. Kersten, ‘The perception of cast shadows’, *Trends in cognitive sciences*, vol. 2, no. 8, pp. 288–295, 1998 (pp. 31, 92, 106, 112).

[51] S. Mathôt, D. Schreij and J. Theeuwes, ‘OpenSesame: An open-source, graphical experiment builder for the social sciences’, *Behavior Research Methods*, vol. 44, no. 2, pp. 314–324, Jun. 2012, ISSN: 1554-3528. DOI: 10.3758/s13428-011-0168-7. [Online]. Available: https://doi.org/10.3758/s13428-011-0168-7 (p. 107).

[52] M. McGuire and M. Mara, ‘Efficient gpu screen-space ray tracing’, *Journal of Computer Graphics Techniques (JCGT)*, vol. 3, no. 4, pp. 73–85, 2014 (p. 29).

[53] M. McGuire, B. Osman, M. Bukowski and P. Hennessy, ‘The alchemy screen-space ambient obscurance algorithm’, in *Proceedings of the ACM SIGGRAPH Symposium on High Performance Graphics*, ACM, 2011, pp. 25–32 (p. 27).

[54] J. L. Mitchell, C. Brennan and D. Card, ‘Real-time image-space outlining for non-photorealistic rendering’, in *ACM SIGGRAPH 2002 Conference Abstracts and Applications*, ser. SIGGRAPH ’02, San Antonio, Texas: ACM, 2002, pp. 239–239, ISBN: 1-58113-525-4. DOI: 10.1145/1242073.1242252. [Online]. Available: http://doi.acm.org/10.1145/1242073.1242252 (p. 29).

[55] M. Mittring, ‘Finding next gen: Cryengine 2’, in *ACM SIGGRAPH 2007 courses*, ACM, 2007, pp. 97–121 (p. 28).

[56] K. Nakatsu, T. Takahashi and T. Moriya, ‘A stereoscopic representation of impossible rectangle twisted torus figure’, in *ACM SIGGRAPH 2012 Posters*, ser. SIGGRAPH ’12, Los Angeles, California: Association for Computing Machinery, 2012, ISBN: 9781450316828. DOI: 10.1145/2342896.2342899. [Online]. Available: https://doi.org/10.1145/2342896.2342899 (p. 21).

[57] G. Nichols, R. Penmatsa and C. Wyman, ‘Interactive, multiresolution image-space rendering for dynamic area lighting’, in *Computer Graphics Forum*, Wiley Online Library, vol. 29, 2010, pp. 1279–1288 (p. 29).

[58] G. Nichols, J. Shopf and C. Wyman, ‘Hierarchical image-space radiosity for interactive global illumination’, in *Computer Graphics Forum*, Wiley Online Library, vol. 28, 2009, pp. 1141–1149 (p. 29).

[59] G. Nichols and C. Wyman, ‘Multiresolution splatting for indirect illumination’, in *Proceedings of the 2009 symposium on Interactive 3D graphics and games*, ACM, 2009, pp. 83–90 (p. 29).

[60] Y. Ostrovsky, P. Cavanagh and P. Sinha, ‘Perceiving illumination inconsistencies in scenes’, *Perception*, vol. 34, no. 11, pp. 1301–1314, 2005 (pp. 33, 34, 94, 98, 103, 104).

[61] S. Owada and J. Fujiki, ‘DynaFusion: A Modeling System for Interactive Impossible Objects’, in *Proceedings of the Sixth International Symposium on Non-Photorealistic Animation and Rendering (NPAR 2008, June 9–11, 2008, Annecy, France)*, 2008, pp. 65–68, ISBN: 9781605581507. DOI: `http://doi.acm.org/10.1145/1377980.1377994` (pp. 20, 60).

[62] S. Owada, J. Fujiki and A. Andre, ‘Panofusion: Stereoscopic panoramic viewing system for view-dependent impossible objects’, *ACM NPAR*, 2008 (p. 20).

[63] G. Papaioannou, M. L. Menexi and C. Papadopoulos, ‘Real-time volume-based ambient occlusion’, *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 5, pp. 752–762, 2010 (p. 27).

[64] L. S. PENROSE and R. PENROSE, ‘Impossible objects: A special type of visual illusion’, *British Journal of Psychology*, vol. 49, no. 1, pp. 31–33, 1958. DOI: `10.1111/j.2044-8295.1958.tb00634.x`. eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.2044-8295.1958.tb00634.x`. [Online]. Available: `https://onlinelibrary.wiley.com/doi/abs/10.1111/j.2044-8295.1958.tb00634.x` (pp. 1, 14).

[65] V. S. Ramachandran *et al.*, ‘Perceiving shape from shading’, *Scientific American*, vol. 259, no. 2, pp. 76–83, 1988 (p. 30).

[66] W. T. Reeves, D. H. Salesin and R. L. Cook, ‘Rendering antialiased shadows with depth maps’, in *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH

’87, New York, NY, USA: Association for Computing Machinery, 1987, pp. 283–291, ISBN: 0897912276. DOI: 10.1145/37401.37435. [Online]. Available: https://doi.org/10.1145/37401.37435 (p. 25).

[67] T. Ritschel, C. Dachsbacher, T. Grosch and J. Kautz, ‘The state of the art in interactive global illumination’, *Computer Graphics Forum*, vol. 31, no. 1, pp. 160–188, 2012. DOI: 10.1111/j.1467-8659.2012.02093.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2012.02093.x. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2012.02093.x (p. 24).

[68] T. Ritschel, T. Grosch and H.-P. Seidel, ‘Approximating dynamic global illumination in image space’, in *Proceedings of the 2009 symposium on Interactive 3D graphics and games*, ACM, 2009, pp. 75–82 (p. 29).

[69] M. Sattler, R. Sarlette, T. Mucken and R. Klein, ‘Exploitation of human shadow perception for fast shadow rendering’, in *Proceedings of the 2nd symposium on Applied perception in graphics and visualization*, ACM, 2005, pp. 131–134 (p. 34).

[70] G. Savransky, D. Dimerman and C. Gotsman, ‘Modeling and rendering escher-like impossible scenes’, *Computer Graphics Forum*, vol. 18, no. 2, pp. 173–179, 1999. DOI: 10.1111/1467-8659.00367. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/1467-8659.00367. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/1467-8659.00367 (p. 21).

[71] R. Sayeed and T. Howard, ‘State of the Art Non-Photorealistic Rendering (NPR) Techniques’, in *Theory and Practice of Computer Graphics 2006*, L. M. Lever and M. McDerby, Eds., The Eurographics Association, 2006, ISBN: 3-905673-59-2. DOI: 10.2312/LocalChapterEvents/TPCG/TPCG06/089-098 (p. 29).

[72] P.-P. Sloan, N. K. Govindaraju, D. Nowrouzezahrai and J. Snyder, ‘Image-based proxy accumulation for real-time soft global illumination’, in *15th Pacific Conference on Computer Graphics and Applications (PG’07)*, IEEE, 2007, pp. 97–105 (p. 29).

[73] N. Smedberg and D. Wright, ‘Rendering techniques in gears of war 2’, in *Niklas Smedberg, Daniel Wright//Proceedings of the Game Developers Conference—2009*, 2009 (p. 28).

[74] K. Sugihara, 'Classification of impossible objects', *Perception*, vol. 11, no. 1, pp. 65–74, 1982 (pp. 16, 17).

[75] J. Sun and P. Perona, 'Shading and stereo in early perception of shape and reflectance', *Perception*, vol. 26, no. 4, pp. 519–529, 1997 (p. 33).

[76] L. Szirmay-Kalos, T. Umenhoffer, B. Toth, L. Szecsi and M. Sbert, 'Volumetric ambient occlusion for real-time rendering and games', *IEEE Computer Graphics and Applications*, vol. 30, no. 1, pp. 70–79, 2009 (p. 28).

[77] O. Tahri and F. Chaumette, 'Point-based and region-based image moments for visual servoing of planar objects', *IEEE Transactions on Robotics*, vol. 21, no. 6, pp. 1116–1127, 2005, ISSN: 1552-3098. DOI: `10.1109/TRO.2005.853500` (pp. 67, 80).

[78] E. B. Thro, 'Distinguishing two classes of impossible objects', *Perception*, vol. 12, no. 6, pp. 733–751, 1983 (p. 17).

[79] S. Tsuruno, 'Natural Expression of Physical Models of Impossible Figures and Motions', *International Journal of Asia Digital Art & Design*, vol. 2, no. 2, pp. 88–95, 2015. [Online]. Available: `http://www.adada.info/journals/vol.18%7B%5C_%7Dno4.4.pdf` (pp. 22, 24, 43, 61, 80).

[80] D. Uribe. (2001). A set of impossible tiles, [Online]. Available: `https://im-possible.info/english/articles/tiles/tiles.html` (p. 20).

[81] N. Wade *et al.*, *Visual allusions: Pictures of perception*. Psychology Press, 1990 (p. 12).

[82] L. Wanger, 'The effect of shadow quality on the perception of spatial relationships in computer generated imagery', *SI3D*, vol. 92, pp. 39–42, 1992 (pp. 31, 32, 92, 124).

[83] L. Williams, 'Casting curved shadows on curved surfaces', in *Proceedings of the 5th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '78, New York, NY, USA: Association for Computing Machinery, 1978, pp. 270–274, ISBN: 9781450379083. DOI: `10.1145/800248.807402`. [Online]. Available: `https://doi.org/10.1145/800248.807402` (p. 25).

[84] T.-P. Wu, C.-W. Fu, S.-K. Yeung, J. Jia and C.-K. Tang, 'Modeling and rendering of impossible figures', *ACM Trans. Graph.*, vol. 29, no. 2, 13:1–13:15, Apr. 2010, ISSN: 0730-0301. DOI: `10.1145/1731047.1731051`.

[Online]. Available: http://doi.acm.org/10.1145/1731047.1731051 (pp. 1, 4, 22, 40, 43, 52, 60, 93).