Interactive Design and Optics-Based Visualization of Arbitrary Non-Euclidean Kaleidoscopic Orbifolds

Jinta Zheng, Eugene Zhang, *Senior Member, IEEE*, and Yue Zhang, *Member, IEEE*

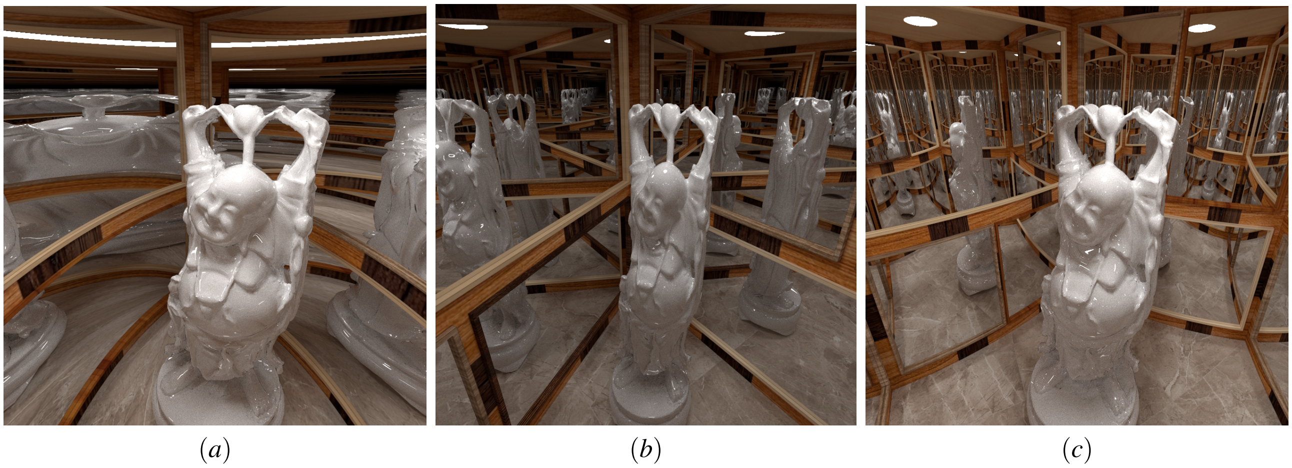


Fig. 1: Our mirror-metaphor visualization of orbifolds: (a) a spherical orbifold, (b) a Euclidean orbifold, and (c) a hyperbolic orbifold. Notice the unusual deformation in the reflections of Buddha and the beams in (a) and (c). All of the orbifolds in this figure were created using our orbifold construction algorithm (Section 5.1).

Abstract—Orbifolds are a modern mathematical concept that arises in the research of hyperbolic geometry with applications in computer graphics and visualization. In this paper, we make use of rooms with mirrors as the visual metaphor for orbifolds. Given any arbitrary two-dimensional kaleidoscopic orbifold, we provide an algorithm to construct a Euclidean, spherical, or hyperbolic polygon to match the orbifold. This polygon is then used to create a room for which the polygon serves as the floor and the ceiling. With our system that implements Möbius transformations, the user can interactively edit the scene and see the reflections of the edited objects. To correctly visualize non-Euclidean orbifolds, we adapt the rendering algorithms to account for the geodesics in these spaces, which light rays follow. Our interactive orbifold design system allows the user to create arbitrary *two-dimensional kaleidoscopic* orbifolds. In addition, our mirror-based orbifold visualization approach has the potential of helping our users gain insight on the orbifold, including its orbifold notation as well as its universal cover, which can also be the spherical space and the hyperbolic space.

Index Terms—Kaleidoscopic orbifolds, orbifold visualization, math visualization, orbifold construction, spherical geometry, hyperbolic geometry

1 Introduction

Orbifolds are a modern mathematical concept originated from the study of low-dimensional topology [32]. This concept has been used to study the geometric structures of hyperbolic spaces and to prove in 2003 the famous Poincaré conjecture for the three-dimensional case, which is the last case and thus the hardest case to be addressed [25]. In addition, orbifolds have been used to describe spatial symmetries in String Theory [13]. Orbifold theory has found applications in tensor field topology [27, 29, 31], remeshing [8, 20, 22], non-photorealistic rendering [16, 28, 36], and texture synthesis [2–4, 26].

The concept of orbifolds can be difficult to digest, as its definition involves various mathematical notions from topology and abstract algebra, such as *Hausdorff spaces*, *groups and group actions*, *charts*, and *gluing maps*. The problem is further compounded by the fact that most of the orbifolds are non-Euclidean, thus making the understanding and visualization of orbifolds more difficult. In this paper, we focus on two-dimensional *kaleidoscopic* orbifolds that are generated by reflections in the Euclidean plane, the sphere, and the hyperbolic space. We have built an interactive system with which our users can create any arbitrary two-dimensional kaleidoscopic orbifolds and interact with them to gain insight into the orbifolds.

Most existing work on visualizing two-dimensional orbifolds make use of texture patterns that tile the plane seamlessly. Such an approach is commonly used to create illustrations. We observe the recent trend of engaging learning through 3D graphics and animations and employ an approach inspired by the kaleidoscopes. By placing some simple objects inside the kaleidoscope, fascinating images appear when we look through the viewing hole. To be more engaging, our system promotes a visual experience of being inside the kaleidoscope entirely. We utilize a room with mirrors as a visual metaphor for a two-dimensional orbifold.

Interestingly, the reflectional symmetries in a kaleidoscope correspond

- *Jinta Zheng is with Oregon State University. E-mail: zhenjint@eecs.oregonstate.edu.*
- *Eugene Zhang is with Oregon State University. E-mail: zhange@eecs.oregonstate.edu.*
- *Yue Zhang is with Oregon State University. E-mail: zhangyue@oregonstate.edu.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

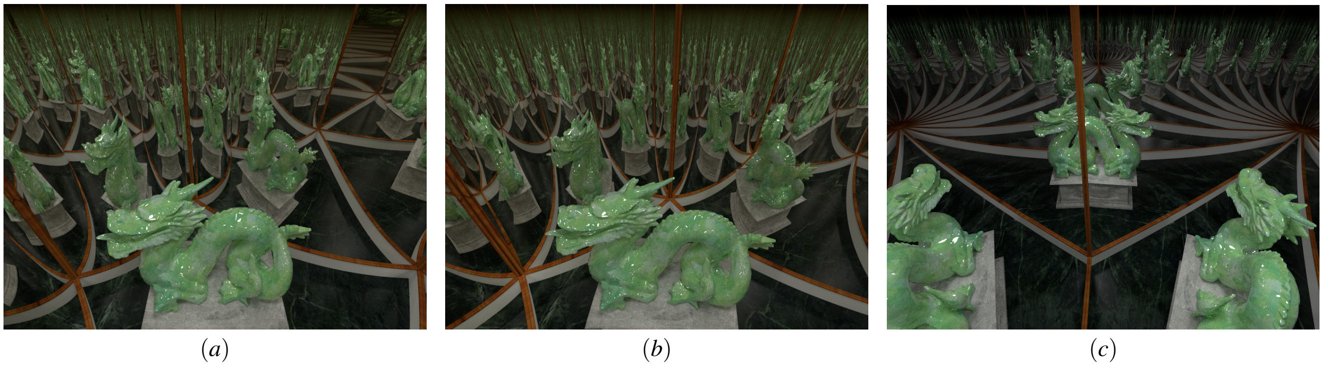
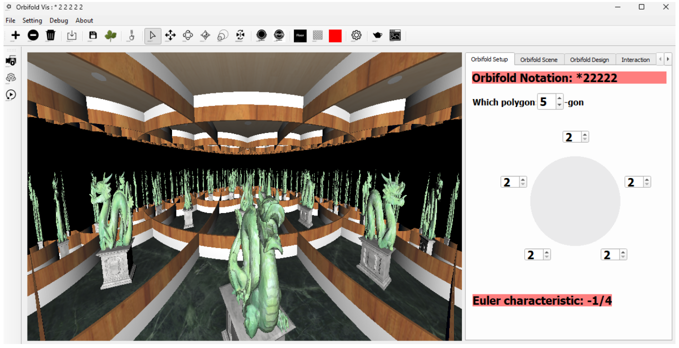


Fig. 2: In (a-b), we show the comparison of the rendering of a hyperbolic orbifold with straight rays as in the Euclidean space (a) and the geodesics in the hyperbolic space (b). In (c), we render a triangular non-orbifold with corner angles $\frac{2\pi}{3}$-$\frac{\pi}{12}$-$\frac{\pi}{12}$. Notice the dragon now has two heads.



Orbifold Notation: *22222

Which polygon 5 -gon

2

2 2

2 2

Euler characteristic: -1/4

Fig. 3: The interface of our design system.

to the behaviors of a particular Euclidean orbifold. As shown in Figure 1, our system produces orbifolds (configuration of the ceiling and the floor) and the symmetry that each orbifold induces. In addition, through the bending of the mirror frames and the unfamiliar deformations of Buddha in (a) and (c), the notions of spherical geometry and hyperbolic geometry are visually delivered, respectively.

While there are only a handful of Euclidean orbifolds, there are infinitely many spherical and hyperbolic orbifolds. In fact, any polygon whose corner angles can each be expressed as $\frac{\pi}{k}$ ($k \in \mathbb{N}^+$) is an orbifold. To the best of our knowledge, there is no algorithm published that allows the realization of arbitrary such polygons when their natural spaces are hyperbolic. Most available tools focus on regular polygons. For arbitrary polygons that represent an orbifold, the lengths of the edges are challenging to determine.

In this paper, we address this difficulty by providing an algorithm that can interactively realize any orbifold, whether spherical, hyperbolic, or Euclidean (Section 5.1). As part of our algorithm, we provide a complete enumeration of two-dimensional kaleidoscopic orbifolds based on the cardinality of the underlying polygon and the type of the *universal cover* (Section 4). With this ability, any two-dimensional orbifold can be converted to a room, whose ceiling and floor have the configuration of the polygon. Our system further allows interactive scene editing, with the room and its virtual copies being visible at the same time (Figure 3). Furthermore, the creation of the reflected copies in the underlying space (*universal cover*) that is either the sphere or the hyperbolic space can reinforce the perception of geometric deformations of these non-Euclidean spaces. To achieve this, we present a system to generate the universal cover of any polygonal orbifold and provide interactive updates to the virtual rooms in the universal cover through Möbius transformations.

Light rays travel along the geodesics in the spherical and hyperbolic spaces. When rendering a non-Euclidean scene using Euclidean straight lines, incorrect appearances result as shown in Figure 2 (a). We modify the rendering algorithms to account for the correct paths for rays (Figure 2 (b)). In addition, by adjusting the attenuation of some or all the mirrors in the scene (Figure 4), we can further emphasize the orbifold itself (a), intensify the emphasis of the universal cover (c), or highlight the *translational cover* of the orbifold (the room and an adjacent virtual room). With our design system and visualization, users can customize their orbifolds for their purposes.

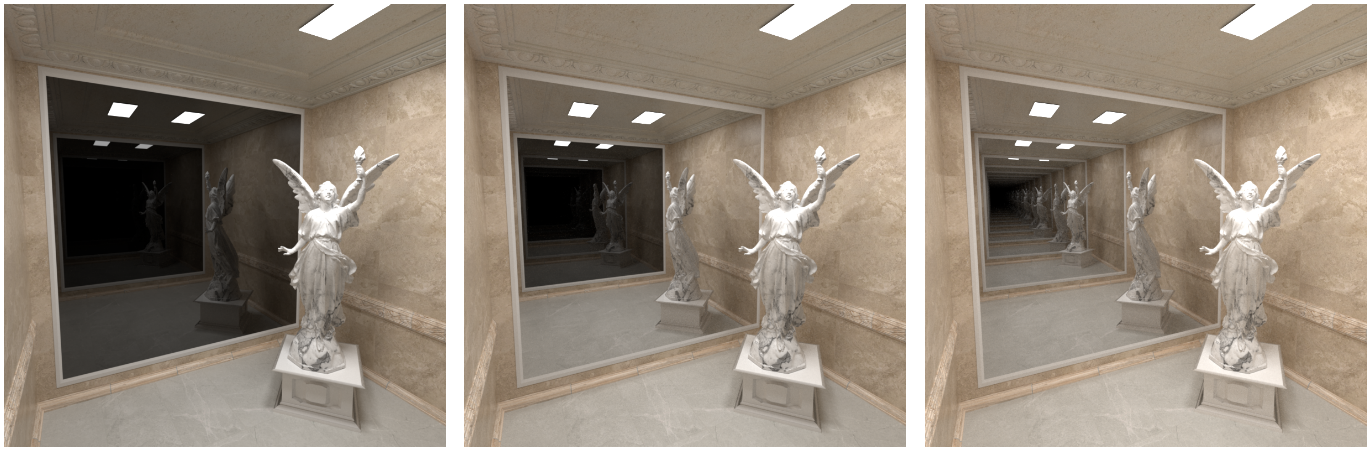
2 Related Work

Our work follows recent research in mathematics visualization, such as quaternions [15], knots and links [33], and branched covering spaces [30].

Orbifolds are a modern mathematical concept [32]. The notion of orbifolds has found applications in texture synthesis [2–4]. Nieser and Polthier provide visualization of analytic functions over the complex plane [23], while Roy et al. [30] visualize the notion and properties of branched covering spaces with respect to N-way rotational symmetry (N-RoSy) fields [28]. All of this work focuses on the visualization and processing of orbifolds on two-dimensional surfaces. Moreover, the orbifolds that they address only possess rotational symmetries, which are a subclass of reflectional symmetries. In this paper, we address orbifolds that are generated by reflections, which include not only reflectional symmetries but also rotational and translational symmetries.

Conway et al. [9] explain the concepts and results related to planar orbifolds by using popular artwork containing textures with symmetries. Their approach focuses on Euclidean orbifolds. In our work, we provide a system to generate a room that matches *any* given two-dimensional kaleidoscopic orbifold, even when its universal cover is a non-Euclidean space. Furthermore, we make use of the mirror metaphor to leverage real-life experience with mirrors, which provides a complementary approach to the texture-based visualization of orbifolds.

Our mirror metaphor turns a two-dimensional orbifold into a three-dimensional room, which can also be considered as a three-dimensional orbifold that is the product of the two-dimensional orbifold (floor and ceiling) with a line segment (the height of the room). There has been some past research on visualizing three-dimensional orbifolds [7, 24], with a focus on the three-dimensional sphere $\mathbb{S}^3$ and three-dimensional hyperbolic space $\mathbb{H}^3$. In these spaces, the geodesics are either a circular arc or a hyperbola. In contrast, the geodesics in our product spaces are spirals, which makes ray-triangle intersection different from those in $\mathbb{S}^3$ and $\mathbb{H}^3$. Moreover, past research often focuses on using some famous orbifolds in $\mathbb{S}^3$ and $\mathbb{H}^3$ such as the Poincaré sphere and the mirror dodecahedron. In our paper, we allow the visualization of any arbitrary two-dimensional orbifolds.



(a) (b) (c)

Fig. 4: By varying the attenuations of the mirrors, our visualization can emphasize the orbifold (a), its translational cover (b), or its universal cover (c).

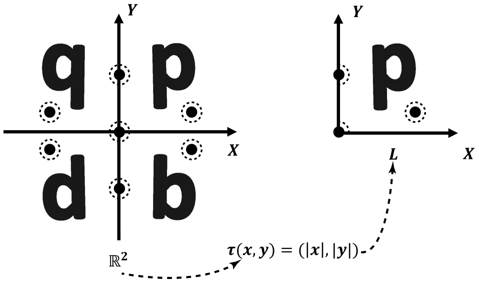
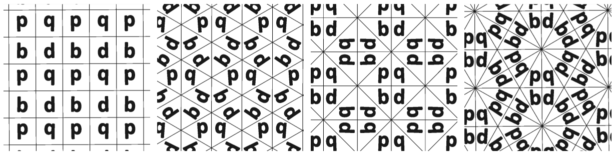


Fig. 5: An orbifold (right: the first quadrant in the real plane) and its universal cover (left: all four quadrants) are related by a covering map $\tau(x,y) = (|x|,|y|)$. The neighborhood of the origin in the orbifold (right) is a quarter disk. The map τ introduces a symmetry group consisting a horizontal reflection (the letter p to q), a vertical reflection (the letter p to b), and a rotation by π (the letter p to d). The symmetry group is the Dihedral group $\mathbb{D}_2$.



(a) $*2222$ (b) $*333$ (c) $*244$ (d) $*236$

Fig. 6: The four Euclidean orbifolds.

3 Background on Orbifolds

In this section, we review necessary mathematical background on orbifolds used in this paper which include the concepts of *groups* and *group actions* [18], *orbifolds* [11,21], and *non-Euclidean spaces* [12]. For a rigorous definition of these concepts, we refer our readers to the aforementioned references.

An orbifold O is a topological space X paired with a discrete symmetry group G such that X locally resembles a Euclidean disk under the action of G. To better illustrate this, consider the space $L = \{(x,y)|x,y \geq 0\}$ (Figure 5 (right)). For each point in the first quadrant, we can find a small enough disk-shaped neighborhood. However, for a point on the positive Y-axis, there is a neighborhood of the shape of a half-disk which corresponds to a full disk in the Euclidean plane (Figure 5 (left)) under the reflection across the Y-axis. In other words, the union of the half disk in the first quadrant and its mirror reflection form a full disk. Finally, the origin has a quarter-disk-shaped neighborhood (Figure 5 (right)) which corresponds to a full disk in the Euclidean plane (Figure 5 (left)) when being combined with its reflection across the X-axis (in the fourth quadrant), the reflection across the Y-axis (in the second quadrant), and the rotation by π around the origin (in the third quadrant). Thus, L is an orbifold.

Globally, we can see that L is the range of the following function $\tau(x,y) = (|x|,|y|)$, which introduces a map from $\mathbb{R}^2$ to L with the symmetry illustrated as follows. The letter p (Figure 5 (right)) corresponds to the letter q in the second quadrant (Figure 5 (left)) through the reflection across the Y-axis and the letter b in the fourth quadrant through the reflection across the X-axis. In addition, it corresponds to the letter d in the third quadrant through a rotation of π around the origin, which is a composition of the two aforementioned reflections. Thus, the symmetry induced by the map τ leads to a symmetry group of four elements: the identity, two reflections, and one rotation. The group is the *Dihedral group* of order 2, i.e. $\mathbb{D}_2$, which, when acted on $\mathbb{R}^2$, leads to the orbifold L. It has a *corner point* at the origin and two mirror lines (the positive X-axis and the positive Y-axis).

In general, a two-dimensional *kaleidoscopic* orbifold O is a polygon with a symmetry group induced by reflections across all of its edges. Without causing ambiguity, we also refer to the polygon as O. Each edge of the polygon is thus a mirror line, and every vertex of the polygon is a corner point corresponding to the symmetry of $\mathbb{D}_k$, the dihedral group of order k. Note that $\mathbb{D}_k$ consists of k rotations (including the identity) and k reflections. In Figure 6 we show four such orbifolds, whose polygons have a configurations of a square (a), a $60^\circ - 60^\circ - 60^\circ$ triangle (b), a $90^\circ - 45^\circ - 45^\circ$ triangle (c), and a $90^\circ - 60^\circ - 30^\circ$ triangle (d). These orbifolds are given the *orbifold notations* (a) $*2222$, (b) $*333$, (c) $*244$, and (d) $*236$, respectively. A generic kaleidoscopic orbifold corresponding to an N-gon O is given the notation $*k_1...k_N$ where the $*$ indicates the existence of the mirror and k_i implies that the angle of the polygon at the i-th corner is $\frac{\pi}{k_i}$.

An orbifold (the polygon) and all of its virtual copies through its symmetry group can seamlessly tile a space, which is its *universal cover*. The aforementioned orbifolds are kaleidoscopic orbifolds whose universal cover is the Euclidean plane, thus *Euclidean orbifolds*. Each Euclidean orbifold has a *translational cover*, which, along with its translational copies, form the universal cover. The translational cover of $*2222$ consists of the orbifold, two of its reflections, and one rotation by π (Figure 6 (a): any 2×2 subgrid with the letters q, p, d, and b). The translations needed to generate the universal cover is the Gaussian

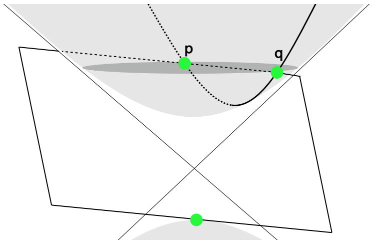


Fig. 7: The hyperbolic space can be modeled as the upper-sheet of a double-sheet hyperboloid. The geodesic passing through p and q in the space is the intersection of the hyperboloid with the plane that passes through p and q as well as the vertex of the lower sheet.

integer grid $\mathbb{Z}[i]$ [14]. The translational covers of the other Euclidean orbifolds respectively consist of six copies arranged in a hexagon (Figure 6 (b): $*333$), eight copies arranged in an octagon (Figure 6(c): $*244$), and twelve copies arranged in a dodecagon (Figure 6(d): $*236$). The set of translations for $*244$ is also $\mathbb{Z}[i]$. On the other hand, the set of translations for $*333$ and $*236$ is the Eisenstein integer grid $\mathbb{Z}[\omega]$ [14] where $\omega = \frac{-1+\sqrt{3}i}{2}$.

While it may seem that these are the only kaleidoscopic orbifolds and that all kaleidoscopic orbifolds must be triangular or rectangular, there are many more. In fact, given an arbitrary polygon with at least three sides and whose corner angles divide π individually, there is an orbifold that corresponds to the polygon. Figure 14 shows a room with three, four, and five mirrors, respectively. However, these orbifolds cannot tile the Euclidean plane as their universal covers are either the unit sphere (spherical orbifolds) or the hyperbolic plane (hyperbolic orbifolds). The hyperbolic plane can be modeled as the upper sheet of the double-sheet hyperboloid $z^2 - x^2 - y^2 = 1$ (Figure 7). Like the Euclidean orbifolds, both spherical orbifolds and hyperbolic orbifolds are polygons whose edges follow the geodesics in their universal cover. The geodesics in the unit sphere are the great circles, and the geodesic passing through two mutually distinct points p and q in the hyperbolic space is the intersection of the plane containing p, q and the vertex of the lower-sheet of the hyperboloid (Figure 7: the curve passing through p and q). In fact, given an orbifold $O = *k_1 ... k_N$ where N is the number of walls and $k_i > 1$ $(1 \leq i \leq N)$, its universal cover is decided by the *Euler characteristic* of the orbifold as follows:

$$\chi(O) = \sum_{i=1}^{N} \frac{1}{2k_i} - \frac{N}{2} + 1. \qquad (1)$$

An orbifold O is spherical, Euclidean, or hyperbolic when $\chi(O) > 0$, $\chi(O) = 0$, $\chi(O) < 0$, respectively.

In the next section, we describe our orbifold design system starting with an enumeration of all two-dimensional kaleidoscopic orbifolds.

4 Kaleidoscopic Orbifold Enumeration

While there has been a complete enumeration of spherical and Euclidean orbifolds, to our best knowledge such an enumeration is not explicitly given for hyperbolic orbifolds. In addition, the enumeration for spherical and Euclidean orbifolds is in the form of an exhaustive list. Our orbifold design system is based on the number of walls (the cardinality of the underlying polygon) in the orbifold. Thus, we strive for an *explicit* enumeration for all two-dimensional kaleidoscopic orbifolds based on the combination of the polygon cardinality and the universal cover.

Table 1: Our enumeration of all two-dimensional kaleidoscopic orbifolds based on the cardinality and the universal cover of the orbifolds. The orbifolds on each row have the same N, which is the cardinality of the underlying polygon.

N	Spherical	Euclidean	Hyperbolic
1	$*$		
2	$*22, *33, *44, \ldots$		
3	$*222, *223, *224 \ldots$		
	$*233, *234, *235$	$*236$	$*237, *238, *239, \ldots$
		$*244$	$*245, *246, *247, \ldots$
			$*2k_2k_3$ $(k_3 \geq k_2 > 4)$
		$*333$	$*334, *335, *336, \ldots$
			$*3k_2k_3$ $(k_3 \geq k_2 > 3)$
			$*k_1k_2k_3$ $(k_3 \geq k_2 \geq k_1 > 3)$
4		$*2222$	$*k_1k_2k_3k_4$ $(\max_{1 \leq i \leq 4} k_i > 2)$
5			$*k_1k_2k_3k_4k_5$
6			$*k_1k_2k_3k_4k_5k_6$
7			$*k_1k_2k_3k_4k_5k_6k_7$
8			$*k_1k_2k_3k_4k_5k_6k_7k_8$
9			$*k_1k_2k_3k_4k_5k_6k_7k_8k_9$
$\vdots$			$\vdots$

There are three types of spherical orbifolds: (1) one mirror, (2) two mirrors, and (3) three mirrors. The only one mirror spherical orbifold is $*$, which corresponds to a room that is half of the sphere with its boundary being the mirror. There are no corners. In this case, one can consider the room as a monogon. In the second case, the room has two mirrors that intersect at $\frac{\pi}{k}$ at both ends where $k > 1$. These are diangular orbifolds $*kk$, which are the section of the unit sphere that are between two longitudes that are $\frac{\pi}{k}$ apart. Note that $*11$ is the same as $*$ since the corner angles are π. In fact, every corner with an angle π can be removed from the list of corners. Thus, in the orbifold notation, we require every number to be at least 2 when there are at least two walls.

For the triangular spherical orbifolds, i.e., three mirrors, there are two sub-types. The first sub-type has the form $*22k$ where $k > 1$. Figure 14 (a) shows one such orbifold ($*222$). This type of orbifolds can be obtained by taking half of the orbifold $*kk$ in the northern hemisphere and adding a mirror on the equator. The second sub-type has the form $*23k$ where $k = 3, 4, 5$. Notice that when $k = 6$ we have $*236$, a Euclidean orbifold. From the discussion, we can see there are more spherical orbifolds than Euclidean orbifolds.

There are *bad* orbifolds, namely, $*k$ where $k > 1$ and $*k_1k_2$ where $k_2 > k_1 > 1$. Note that neither type of the bad orbifolds can be realized because it is not physically possible to have one great circle intersecting itself at an angle not equal to π, nor is it possible to have two different great circles that intersect at different angles where they meet. In our system, we do not construct bad orbifolds.

The rest of polygonal kaleidoscopic orbifolds are hyperbolic, and there are no bad hyperbolic orbifolds. There are three cases: (1) three mirrors, (2) four mirrors, and (3) five or more mirrors. An orbifold is hyperbolic if its has five or more mirrors (e.g. Figure 14 (c)). In addition, all quadrangular orbifolds except $*2222$ are hyperbolic (e.g. Figure 14 (b)). Finally, triangular hyperbolic orbifolds include six sub-types: (1) $*23k$ where $k > 6$, (2) $*24k$ where $k > 4$, (3) $*2k_2k_3$ where $k_3 \geq k_2 > 4$, (4) $*33k$ where $k > 3$, (5) $*3k_2k_3$ where $k_3 \geq k_2 > 3$, and (6) $*k_1k_2k_3$ where $k_3 \geq k_2 \geq k_1 > 3$. Notice the three cases, each of which corresponds to a Euclidean orbifold that serves as the border between the set of spherical and the set of hyperbolic orbifolds, namely, $*236$ for type (1), $*244$ for type (2), and $*333$ for type (4).

Our enumeration of all two-dimensional kaleidoscopic orbifolds based on the combination of the cardinality of the underlying polygon and the type of its universal cover is shown in Table 1. We provide the computations behind our enumeration in Appendix A.

5 Interactive Orbifold Scene Design

Our orbifold visualization system consists of two components: a design panel and the display (Figure 3). We employ the Irrlicht game engine [1], which provides an effective balance between interactivity and functionality.

In the design panel, the user can specify the type of the scene by entering its orbifold notation in the form of a number N for the number of walls in the scene and a list of N numbers, $k_1, k_2, ..., k_N$. Here, k_i indicates that the angle of the i-th corner is $\frac{\pi}{k_i}$.

The default value of N is five, and five evenly spaced nodes are displayed on the disk inside the design panel (Figure 3), each of which has a default value of two, i.e. $*22222$. The user can change the value of each node, which can be a non-integer in order to create non-orbifold scenes (Figure 2 (c)). The user can also change N, which results in a room with more or fewer walls. The default value for each node in the new setting is again two. Recall that there are two cases that are not physically realizable: (1) a circular room with a single mirror ($N = 1$) that self-intersects at an angle not equal to π ($*k$ where $k > 1$), and (2) a room with two mirrors whose two intersection angles are mutually distinct ($*k_1k_2$ where $k_2 > k_1$). Thus, we disallow these cases from occurring during the design phase. For example, when $N = 1$, the value of the only node is set to one and cannot be changed. Similarly, when $N = 2$, if the user changes the value of one node, the value of the other node is automatically updated to match it.

Given the orbifold notation, our system instantaneously generates an empty room (a *right* polygonal prism) whose floor and ceiling are congruent to the orbifold and whose walls are the sides of the prism. In our system, it is possible to have multiple mirrors on a wall as shown in Figure 1. The user can also change the height of the room, the color and attenuation of a mirror, and the textures for the ceiling and the floor. Objects can be added to the scene, whose locations, orientations, sizes, base colors, and material properties (e.g. marble, glass) can be modified from their default values as needed. Light sources can also be added to the scene, with control over their locations, shapes, and optical properties. Unwanted objects and light sources can be removed from the scene.

All of the above scene design operations are interactively rendered in order to support the *What-You-See-Is-What-You-Get* (WYSIWYG) paradigm, and all of the examples included in the paper and accompanying video were created using our design system.

At the core of our system is the ability to create a room given any arbitrary two-dimensional kaleidoscopic orbifold and to correctly deform an object in the scene when it moves. In addition, in our design system, mirrors are not explicitly generated. Instead, we emulate the mirror effects by creating copies of the original room which together approximate the universal cover of the orbifold. We provide detail on each of these topics next.

5.1 Orbifold Scene Construction

We first compute the Euler characteristic of the orbifold O. If $\chi(O) = 0$, i.e. a Euclidean orbifold, it is then $*2222$, $*333$, $*236$, or $*244$. As we know the internal angles and the ratios between the side lengths, we can place the vertices of the underlying triangle or square on the floor. Then, with a user-specified room height, we create the ceiling polygon by duplicating the floor polygon and raising it to match the height. Both the floor and the ceiling are represented by a triangular mesh, so are the rectangular walls in the room.

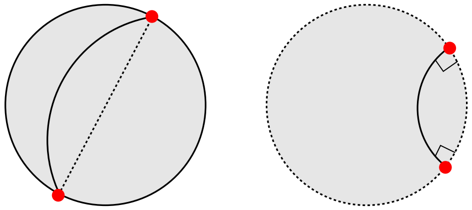


Fig. 8: Under the stereographic projection, a geodesic in the sphere is mapped to a circle that intersects the boundary of the unit disk at a pair of antipodal points (left). Using the Poincaré disk model, a geodesic in the hyperbolic space is mapped to a circular arc that interests the boundary of the disk at the right angle (right).

For a non-Euclidean orbifold, its universal cover is either the sphere or the hyperbolic space. Constructing a 3D room over the sphere and the hyperboloid would require a second sphere or hyperboloid to hold the ceiling. While it is possible to construct the room this way, we instead choose to express the orbifold using a planar model, i.e. the stereographic projection for the sphere [17] and the Poincaré disk [12] for the hyperbolic space. By using these models, we have a unified framework in which any polygon, regardless of the type of its universal cover, can be constructed in the plane. Next, we describe our algorithm to identify the side lengths of any non-Euclidean kaleidoscopic orbifold.

Spherical Orbifolds: The stereographic projection [17] maps the unit sphere to the plane $z = 0$ such that the equator (a unit circle) is mapped to itself and the north pole is mapped to the origin in the plane. In this case, the northern hemisphere is mapped to the inside of the unit disk bounded by the equator while the southern hemisphere is mapped to the outside of the unit disk. This plane can be identified as the complex plane, i.e. the set of complex numbers. The south pole is mapped to ∞. The geodesics are mapped to circles in the plane that intersect the unit circle at a pair of antipodal points (Figure 8 (left)).

As shown in Table 1, the underlying polygon of a spherical orbifold is either a monogon, a diangle, or a triangle. In all of these cases, O can be contained inside a hemisphere. That is, under the stereographic projection, it can be contained inside the unit disk in the complex plane. For the monogon, i.e. $*$, the equator is the mirror. For a diangular orbifold $*kk$ ($k > 1$), the corner points are on the real axis which are connected by a pair of circle segments that intersect at the corner points at $\frac{\pi}{k}$. Note that the stereographic projection is conformal [10], i.e. angle-preserving, thus our choice of the angles at the intersection points. The triangular orbifold $*22k$ ($k > 1$) is exactly half of the orbifold $*kk$ for the same k (Figure 9: top row).

Finally, the orbifolds $*23k$ ($k = 3, 4, 5$) can be constructed by placing the corner points (Figure 9, bottom row) in the stereographic plane as follows. Let p_1, p_2, p_3 be the corners of the orbifold corresponding to 2, 3 and k, respectively. The spherical lengths, $d_{i,i+1}$, of the edges p_ip_{i+1} in the polygon are uniquely determined by the angles $\frac{\pi}{k_i}$, $\frac{\pi}{k_{i+1}}$ and $\frac{\pi}{k_{i+2}}$ of the polygon as follows [32]:

$$d_{i,i+1} = \cos^{-1}\left(\frac{\cos\left(\frac{\pi}{k_{i+2}}\right) + \cos\left(\frac{\pi}{k_i}\right)\cos\left(\frac{\pi}{k_{i+1}}\right)}{\sin\left(\frac{\pi}{k_i}\right)\sin\left(\frac{\pi}{k_{i+1}}\right)}\right) \qquad (2)$$

where $i = 1, 2, 3$, $d_{3,4} = d_{3,1}$, $k_4 = k_1 = 2$, $k_5 = k_2 = 3$, $k_3 = k$, and $p_4 = p_1$. With this information, we first place the vertex p_1 at $(1, 0)$ in the complex plane. Next, we compute a geodesic emanating from p_1, along which we travel for a *spherical* distance of $d_{1,2}$ to find p_2. Since p_1 and p_2 are both represented as complex numbers in the stereographic plane, their spherical distance can be computed from their complex number representations as follows [17]:

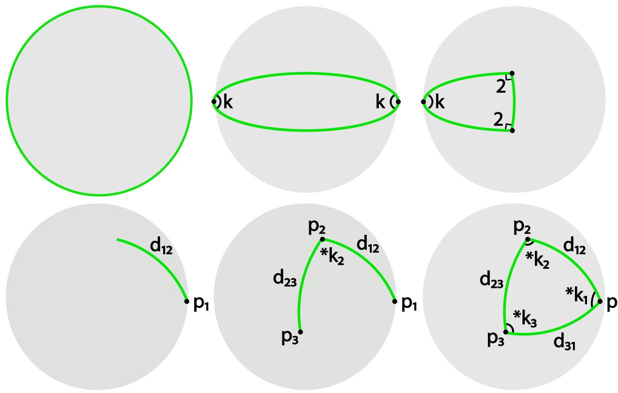


Fig. 9: In the stereographic plane, the $*$ is realized as the unit disk (top row: left). A $*kk$ orbifold can be created by placing its corner points at a pair of antipodal points on the unit disk (top row: middle). The $*22k$ type of orbifold is half of the $*kk$ orbifold (top row: right). The $*23k$ ($k = 3,4,5$) can be constructed as shown in the bottom row.

$$d(p_1, p_2) = 2\tan^{-1}\left(\left|\frac{p_2 - p_1}{1 + \overline{p_1}p_2}\right|\right) \tag{3}$$

where $\overline{p}$ is the conjugate of a complex number p. Solving for p_2 in Equation 3 can be challenging given any arbitrary geodesic γ. However, on the unit circle in the stereographic plane, one can find p_2 without the need to solve Equation 3. This is because the unit circle corresponds to the equator in the sphere under the stereographic projection, thus the spherical distance between the two points is the same as the arc distance between them on the unit circle. However, γ is not always the unit circle. To address this, we identify a translation in the sphere, which, under the stereographic projection, takes γ (Figure 10 (left): the arc) to the upper-half of the unit circle. Such a translation can be modelled by Möbius transformations [6] that have the following form:

$$f(z) = e^{i\theta}\frac{z - z_0}{\overline{z_0}z + 1}. \tag{4}$$

Here, $\theta \in [0, 2\pi)$ and z_0 and z are complex numbers. A Möbius transformation is uniquely determined by three pairs of corresponding points [17]. We first extend γ until it intersects the unit disk and map the intersection points to $(-1,0)$ and $(1,0)$, respectively (Figure 10 (left)). We select the third point to be middle point of γ, which is mapped to $(0,1)$. Call this Möbius transformation M. Since Möbius transformations in the stereographic plane correspond to rotating the sphere before the stereographic projection, spherical distance is preserved M. Thus, we can find the point $M(p_2)$ on the unit circle which has an arc distance of $d_{1,2}$ from $M(p_1)$. Finally, performing the inverse of M, which is also a Möbius transformation to $M(p_2)$, we can find $p_2 = M^{-1}(M(p_2))$.

From p_2, we compute a second geodesic which has an angle of $\frac{\pi}{k_2}$ with the first geodesic. Then, travelling along the new geodesic for a *spherical* distance of $d_{2,3}$, we can locate p_3. Thus, we have constructed the $*23k$ type of orbifolds (Figure 9: bottom row).

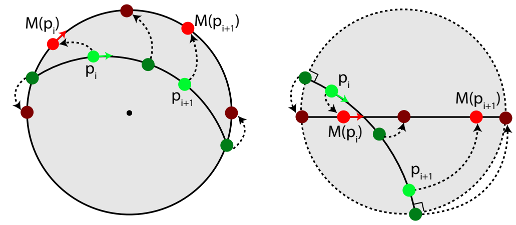


Fig. 10: Given a point p_i, we simplify the computation of p_{i+1} by performing a Möbius transformation in the stereographic plane for spherical orbifolds (left) and the Poincaré disk for hyperbolic orbifolds (right). In both cases, the unique Möbius transformation M maps the extended geodesic (including the intersections with the unit disk) to the upper semi-circle for spherical orbifolds and line segment between $(-1,0)$ and $(1,0)$ for hyperbolic orbifolds. In addition, the center of extended geodesic is mapped to $(0,1)$ in the spherical case (left) and $(0,0)$ in the hyperbolic case (right). Then we identify $M(p_{i+1})$ from which we can recover p_{i+1} using the inverse of M.

Hyperbolic Orbifolds: We now consider the hyperbolic case, where $\chi(O) < 0$. The hyperbolic space can be modelled by the Poincaré disk [12], which is the interior of the unit disk in the plane. Under this model, the geodesics of the hyperbolic space are circles that intersect the boundary of unit disk at the right angle (Figure 8 (right)). Recall that a hyperbolic orbifold is a polygon with three or more sides (Table 1). To create such a polygon, we follow the same approach for spherical orbifolds. That is, we start with the location of p_1 in the Poincaré disk and a geodesic emanating from p_1. We then travel along this geodesic for a prescribed distance to locate p_2. From there, we identify a new geodesic whose angle with the original geodesic is $\frac{\pi}{k_2}$, which we follow to identify p_3. This process terminates once when we have identified p_N. The main difference lies in the fact that the hyperbolic distance between two points in the Poincaré disk is different from the spherical distance of the same two points in the stereographic plane.

Given a triangular hyperbolic orbifold $*k_1k_2k_3$, the length of the edge between p_i and p_{i+1} ($i = 1,2,3$, $d_{3,4} = d_{3,1}$, $k_4 = k_1$, $k_5 = k_2$, and $p_4 = p_1$) is given by [32]

$$d_{i,i+1} = \cosh^{-1}\left(\frac{\cos\left(\frac{\pi}{k_{i+2}}\right) + \cos\left(\frac{\pi}{k_i}\right)\cos\left(\frac{\pi}{k_{i+1}}\right)}{\sin\left(\frac{\pi}{k_i}\right)\sin\left(\frac{\pi}{k_{i+1}}\right)}\right) \tag{5}$$

In addition, when represented as complex numbers in the plane containing the Poincaré disk, the hyperbolic distance between p_i and p_{i+1} is given by [17]:

$$d(p_i, p_{i+1}) = \ln\left(\frac{|1 - \overline{p_i}p_{i+1}| + |p_{i+1} - p_i|}{|1 - \overline{p_i}p_{i+1}| - |p_{i+1} - p_i|}\right) \tag{6}$$

Similar to the case of spherical orbifolds, finding p_{i+1} from p_i on an arbitrary geodesic γ in the Poincaré disk requires solving Equation 6 for p_{i+1} which can be challenging. To simplify the matter, we identify the hyperbolic translation in the Poincaré disk that takes γ (Figure 10 (right): the arc) to the line segment from $(-1,0)$ to $(1,0)$ in the Poicnaré disk (Figure 10 (right)). This translation takes the intersection with the unit circle (two dark green points on the unit circle) to $(1,0)$ and $(-1,0)$ (the rightmost and the leftmost burgundy points). It also takes the middle points on γ (the middle dark green point) to $(0,0)$ (the middle burgundy point). Such a translation can be modelled by a Möbius transformation of the following form:

$$f(z) = e^{i\theta}\frac{z - z_0}{1 - \overline{z_0}z} \tag{7}$$

where $\theta \in [0, 2\pi)$ and $|z_0| < 1$. Call the translation M. Since translations maintain hyperbolic distances, the hyperbolic distance between p_i and p_{i+1} is the same as the distance between $M(p_i)$ and $M(p_{i+1})$. However, since $M(p_i)$ and $M(p_{i+1})$ are on the real axis, it is easier to solve Equation 6. Once we have found $M(p_{i+1})$, we can recover p_{i+1} by applying M^{-1}.

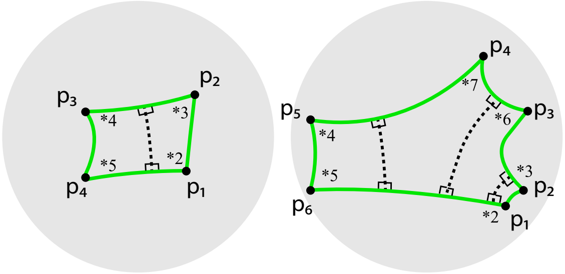


Fig. 11: Two example scenarios of our decomposition algorithm for hyperbolic orbifolds: (left) a quad is divided into two $*22k_3k_4$ type quads, and (right) a hexagon is divided into two $*22k_3k_4$ quads and two $*2222k_5$ pentagons.

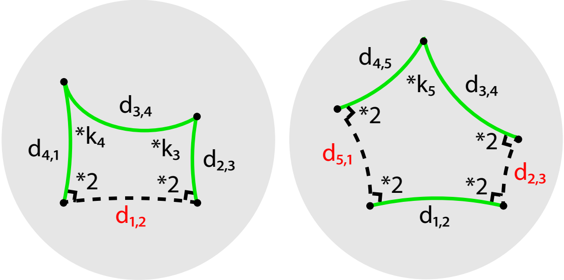


Fig. 12: The special quad $*22k_3k_4$ (left) and the special pentagon $*2222k_5$ (right) are the building blocks of our decomposition algorithm. The free variables are colored in red, which correspond to the edges (dashed) introduced during the decomposition.

However, deciding the side lengths of a hyperbolic polygon with at least four edges is more challenging as there are no published formulas to the best of our knowledge. To address this, we compute the side lengths based on two facts [21]: (1) any quadrangular hyperbolic polygon can be decomposed into the disjoint union of at most two quads with two right angles ($*22k_3k_4$), and (2) any hyperbolic polygon with at least five sides can be decomposed into the disjoint union of a finite number of quads of the type $*22k_3k_4$ and pentagons of the type $*2222k_5$, i.e. four right angles. Examples of the two facts are shown in Figure 11.

The side lengths of the $*22k_3k_4$ polygon (Figure 12 (left)) are given by [21, 32]:

$$
\begin{aligned}
d_{2,3} &= \sinh^{-1}\left(\frac{\cos\left(\frac{\pi}{k_4}\right)+\cos\left(\frac{\pi}{k_3}\right)\cosh\left(d_{1,2}\right)}{\sin\left(\frac{\pi}{k_3}\right)\sinh\left(d_{1,2}\right)}\right)\\
d_{3,4} &= \cosh^{-1}\left(\frac{\cosh\left(d_{1,2}\right)+\cos\left(\frac{\pi}{k_3}\right)\cos\left(\frac{\pi}{k_4}\right)}{\sin\left(\frac{\pi}{k_3}\right)\sin\left(\frac{\pi}{k_4}\right)}\right)\\
d_{4,1} &= \sinh^{-1}\left(\frac{\cos\left(\frac{\pi}{k_3}\right)+\cos\left(\frac{\pi}{k_4}\right)\cosh\left(d_{1,2}\right)}{\sin\left(\frac{\pi}{k_4}\right)\sinh\left(d_{1,2}\right)}\right)
\end{aligned}
\tag{8}
$$

where $d_{1,2}$, the length of the cut edge in the decomposition, is a free variable. This is similar to the case where the width and length of the $*2222$ orbifold (a rectangle) are free variables.

A generic quadrangular orbifold ($*k_1k_2k_3k_4$) can be decomposed into the disjoint union of two quads ($*22k_1k_2$) and ($*22k_3k_4$) (Figure 11, left). This allows us to compute side lengths of the two special quads, which, when combined, give rise to the side lengths of the generic quad.

The side lengths of the $*2222k_5$ polygon (Figure 12 (right)) can computed as follows: [21, 32]:

$$
\begin{aligned}
d_{1,2} &= \cosh^{-1}\left(\frac{\cos\left(\frac{\pi}{k_5}\right)+\cosh\left(d_{2,3}\right)\cosh\left(d_{5,1}\right)}{\sinh\left(d_{2,3}\right)\sinh\left(d_{5,1}\right)}\right)\\
d_{4,5} &= \sinh^{-1}\left(\frac{\cosh\left(d_{2,3}\right)+\cos\left(\frac{\pi}{k_5}\right)\cosh\left(d_{5,1}\right)}{\sinh\left(d_{5,1}\right)\sin\left(\frac{\pi}{k_5}\right)}\right)\\
d_{3,4} &= \sinh^{-1}\left(\frac{\cosh\left(d_{5,1}\right)+\cos\left(\frac{\pi}{k_5}\right)\cosh\left(d_{2,3}\right)}{\sinh\left(d_{2,3}\right)\sin\left(\frac{\pi}{k_5}\right)}\right)
\end{aligned}
\tag{9}
$$

where $d_{2,3}$ and $d_{5,1}$, the cut edges in the decomposition, are free variables.

Generic pentagons and higher-order N-sided polygons can be decomposed as the union of $N-4$ pentagons of the type $*2222k_5$ with up to two additional quads of the type $*22k_1k_2$. To do so, we first consider the simplest case where there exist $k_i = k_{i+1} = 2$ and $k_j = k_{j+1} = 2$ where $i, i+1, j, j+1$ are mutually distinct. We can find a geodesic that intersects $p_{i-1}p_i$ and $p_{i+1}p_{i+2}$ at the right angle. This geodesic removes from the original polygon a pentagon involving p_i, p_{i+1}, and p_{i+2} which has four right internal angles, thus the type $*2222k_5$. The remaining polygon has one fewer vertex and still has four internal right angles (Figure 11). Repeating this process can lead to $N-4$ pentagons of the type $*2222k_5$. On the other hand, a generic polygon with at least five edges can be reduced into the simpler setting by removing up to two quads of the type $*22k_3k_4$. We can then compute the side lengths of each of the sub-polygons, which, when combined, give the side lengths of the original polygon.

Every cut edge in the decomposition gives rise a free variable, which can be modified by the user. The default value for the free variables is set to 1.4.

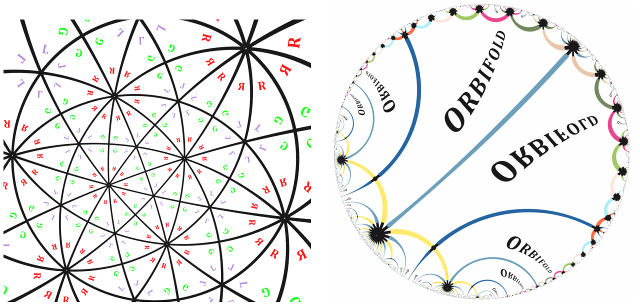
With our algorithm, any spherical and hyperbolic orbifold can be constructed given its orbifold notation. Figure 13 show two example orbifolds with their universal covers.

5.2 Object Embedding and Movement

Once the orbifold has been realized as a polygonal prism, the user can add objects to the scene.

However, when bringing an object, which is created in a presumably Euclidean space, to a non-Euclidean space, a natural question to ask is how to perform the embedding. Due to the difference in their respective distance metrics, it is not always possible to embed the model in such a way that the length of every edge in the mesh is maintained. To address this challenge, when embedding the object into the scene, we first place it so that its center of mass is at the origin of the plane for both the stereographic plane and the Poincaré disk. The coordinates of the object are now considered their corresponding coordinates in the stereographic plane and the Poincaré disk. Then, the embedded mesh is translated to the user-specified initial location with the translation native to the non-Euclidean space.

Translations in both the spherical and hyperbolic spaces are isometries. In the spherical space, translations can be modelled in the stereographic plane by Möbius transformations according to Equation 4. Interestingly, translations in the hyperbolic space using the Poincaré disk can also be modelled by Möbius transformations according to Equation 7.



(*a*) $*235$ (*b*) $*23456789(10)(11)$

Fig. 13: Two orbifolds generated using our orbifold layout creation algorithm: (a) the $*235$ orbifold (spherical) and (b) the orbifold $*23456789(10)(11)$ (hyperbolic).

We store the Möbius transformation of each object, and update it when the model is interactively moved inside the room. The Möbius transformation, which corresponds to a translation of the spherical space or the hyperbolic space, is applied to all the vertices in the mesh.

5.3 Universal Cover Construction

Being able to see the universal cover, i.e. all mirror reflections, can be important for a user while exploring our tool. However, capturing mirror reflections can be computationally expensive with high-quality renderers such as Mitsuba [19]. Williams and Zhang [34] address this for Euclidean kaleidoscopic orbifolds by creating a finite number of copies of the reflections of the original room that approximate the universal cover. This is based on two observations. First, as copies are farther away from the room, their images perceived by the viewer approach the vanishing line and thus do not contribute much to the pixels. Second, the color intensity of faraway copies diminishes as the number of bounces off from the mirrors increases. We employ a similar approach, which focuses on non-Euclidean orbifolds. Two examples of universal covers created by our method are shown in Figure 13: (a) the spherical orbifold $*235$ and (b) the hyperbolic orbifold $*23456789(10)(11)$.

In our system, the construction of the universal cover for Euclidean orbifolds closely follows that of Williams and Zhang [34], which computes the translational cover of the orbifold and generates additional copies of the translational cover using either the Gaussian integer lattice $\mathbb{Z}[i]$ for the $*2222$ and $*442$ cases and the Eisenstein integer lattice $\mathbb{Z}[\omega]$ for the $*333$ and $*632$ cases.

For spherical and hyperbolic orbifolds, the notion of translational cover is not well-defined. Consequently, we employ the following process. Starting from the original room, we iteratively add a virtual copy by reflecting the room across one of its mirrors. It is also possible to reflect a virtual room across its mirror. To avoid duplicates, i.e. a virtual room that is discovered through two different paths from the original room, we compare the center of a potentially new room to the centers of already visited room and virtual rooms [35].

To locate the corners of each newly added room, we apply reflections in the spherical or hyperbolic spaces to the already visited room involved in the reflection. Inside the stereographic plane, a reflection in the spherical space can be represented as the compositions of some Möbius transformations (Equation 4) and the conjugation function with respect to the real axis $f(z) = \bar{z}$. Similarly, inside the Poincaré disk, a reflection in the hyperbolic space can also be expressed as the composition of some Möbius transformation (Equation 7) and the conjugation function with respect to the real axis. The Möbius transformation is stored in the form of a 2×2 matrix whose entries consist of a, b, c, and d [5].

For each copy of the room we save the transformation needed to take the original room to the copy, which is a combination of a Möbius transformation and up to one conjugation function. In addition, for each object in the original room, we also save its Möbius transformation. Then, the position and orientation of an object in a virtual copy of the original scene can be calculated by combining the matrix for the virtual room and the matrix of the same object in the original scene. Also, the computation for both object manipulation and universal cover construction is done using the shader.

6 Optics-Based Orbifold Visualization

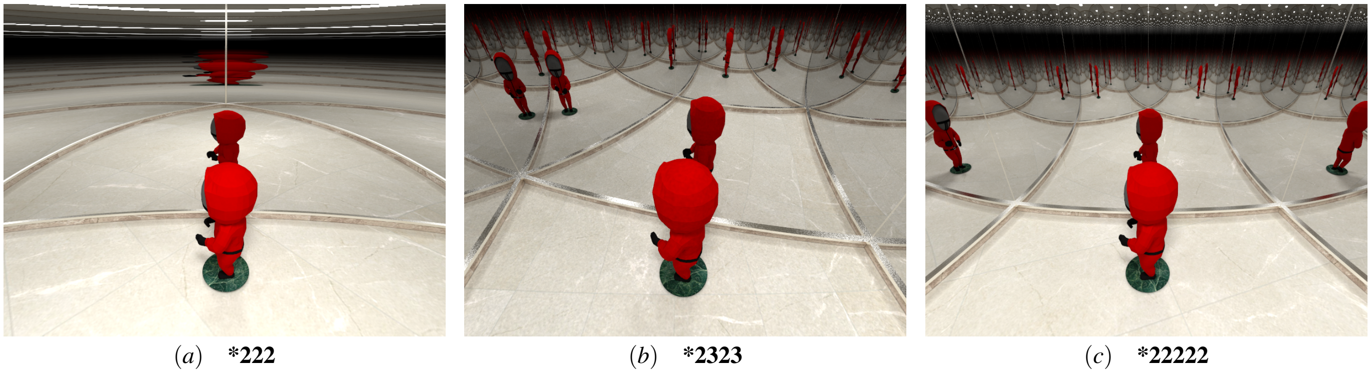
Our mirror-based visual metaphor and interactive orbifold design algorithm are motivated by a number of tasks for understanding the concept of orbifolds. Additional results illustrating concepts and properties related to orbifolds can be found in Appendix B. Below we list four tasks and show the results of using our approach to address them.

Recognizing Non-Orbifolds: One of the most fundamental tasks for orbifold visualization is to decide whether a given mirror scene is an orbifold. Non-orbifolds can be difficult to conceptualize, especially when one or more angles at the corners violates $\frac{\pi}{k}$ for $k \in \mathbb{N}^+$. The resulting rendered images and/or videos can deliver a sense of whether the room is a kaleidoscopic orbifold. When the room is a kaleidoscopic orbifold, the viewer should be able to move around the scene and see a consistent larger scene (the universal cover) that is seamlessly tiled by copies of the original room (the orbifold). On the other hand, if the viewer sees conflicting results, such as the double-headed dragon in Figure 2 (c), it is an indication of a non-orbifold. In this particular example, the room has a corner with an angle of $\frac{\pi}{k}$ where $k = \frac{2}{3}$, which is not an integer.

Sensing the Orbifold: An orbifold is a topological space that stands on its own. Using our mirrored scenes, we can produce visualizations that emphasize different aspects of an orbifold. For example, for the scene shown in Figure 4, we can adjust the attenuation factors for the mirrors to emphasize the orbifold itself (Figure 4 (a)) by setting high attenuation for both mirrors, or the translational cover (Figure 4 (b)) by setting high attenuation for one mirror and low attenuation for the other mirror, or the universal cover (Figure 4 (c)) by setting low attenuation for both mirrors. Notice that in all of these cases, our mirror-based visualization provides a clear sense of the orbifold, the room.

Determining the Orbifold Notation: Once an intuitive sense of the orbifold is established and the structure of the reflection is observed, it becomes a relatively straightforward task to determine the type of the orbifold in terms of its orbifold notation. Given the power of 3D graphics and animation, we produce either a panorama of the scene or an animation in which an avatar walks along the walls (see the accompanying video). As the avatar approaches the i-th corner, so do its nearest reflections. There appear to be k_i pairs of avatars approaching the corner. Then, k_i pairs of the same avatars leave the corner for the next one. This helps the user identify k_i for the i-th corner. By walking around the room one time, each corner is visited. This can help the user write down the orbifold notation.

Identifying the Type of the Universal Cover: With the same panorama and walkaround animations, the user can gain insight into the type of the universal cover. Both the spherical space and the hyperbolic space can bring unfamiliar experience to someone new to the concept. For example, inside the spherical space, objects do not always appear smaller when they are further away. Specifically, when the viewer is at the south pole, objects near the north pole appear much wider than the same object at the equator (Figure 14 (a)). On the other hand, objects farther away always look smaller in the hyperbolic space. However, with the seemingly same distance to the viewers, an object can look much smaller in the hyperbolic space than in the Euclidean space (Figure 14 (b-c)). As the objects move around the scene such as the avatar



(*a*) **$*222$** (*b*) **$*2323$** (*c*) **$*22222$**

Fig. 14: An avatar walks along the walls of the orbifolds. The universal cover is the sphere for (a) and the hyperbolic space for (b) and (c).

in Figure 14 and the accompanying video, the way the reflections of the objects deform in non-Euclidean spaces is rather different from that in the Euclidean space. For example, a reflection of the avatar may suddenly grow much bigger and then shrink quickly in the spherical space. With a relatively simple scene (the orbifold), the user can gain insight into the entire universal cover through mirror reflections.

7 Rendering

Once the scene has been constructed, we can either display it interactively during the design stage (Figure 3) or send it to Mitsuba [19] for high-quality off-line rendering. For the latter, we make use of a ray-tracing type of approach with Mitsuba. Recall that our non-Euclidean room is modelled by a subset in $\mathbb{D} \times [0,h]$ where $\mathbb{D}$ is the unit disk and h is the height of the room. Thus, any geodesic γ in the space satisfies that its horizontal projection is a geodesic in the spherical or hyperbolic space. Consequently, γ takes the form of a 3D spiral. To generate the correct rendering of such scenes, we modify the ray-triangle intersection algorithm in Mitsuba from line-plane intersection to spiral-plane intersection.

Given a point p_0 in the room and a tangent direction v, the spiral emanating from p_0 in the direction of v is the tensor product of horizontal projection and the vertical projection of the spiral. The horizontal projection is a circle (can also be a line) in the floor that starts with the projection of p_0 onto the floor and travels in the direction of v_H, the projection of v onto the floor. The vertical projection of the spiral is a line that starts at p that travels in the direction of v_Z, the z-component of v. This leads to a parametric form of γ which we use to find the intersection of the spiral γ with a given triangle on an object in the scene. When there are multiple intersections with the triangle, we find the one that has the smallest positive t value, which represents the closest intersection from the reference point p.

During the design stage, we make use of scan conversion for interactive feedback. Recall that in this case, we also render the universal cover since we do not model mirror reflections during design. For each vertex on an object in the scene or a virtual copy, we project it to the image plane by using the same spiral-triangle intersection as in the case of Mitsuba rendering. We then perform barycentric interpolation to find the footprint of any triangle in the image plane by interpolating their vertices' locations. Given enough mesh resolution, the error in this interpolation-based approach is relatively small since all of its vertices are projected correctly using the spiral-triangle intersection.

8 Performance

Our interactive design system is evaluated on a computer featuring an i7-8700K @3.70GHz CPU and an NVIDIA GeForce GTX 2080 GPU. For an orbifold $*22222$ with 26,022 primitives at an 800×600 resolution, our system can render its universal cover approximated by 166 copies of the original room at 11 frames per second. When the number of copies is reduced to 61, the frame rate is increased to 30 frames per second.

For off-line rendering, we make modifications to Mitsuba 0.6. The rendering was performed on a Linux Cluster, comprised of machines equipped with 12-core 2.67 GHz processors and 48 GB SDRAM, to generate high-resolution images and videos based on the scenes that we created. Due to the increased computational cost from the spiral-triangle intersections, the rendering is considerably slower for non-Euclidean orbifolds than Euclidean orbifolds. As an example, for the single frame of the spherical orbifold $*222$ that contains $12,572$ primitives (shown in Figure 14 (a)), it required 7.69 minutes when rendered at a resolution of 800×600.

9 Conclusion and Future Work

In this paper, we propose the use of mirror reflections as a visual metaphor for orbifolds and provide a system in which the user can interactively design any two-dimensional kaleidoscopic orbifold. At the core of our interactive design system is the ability to determine the configuration of the room (locations of the corners, shapes of the walls) given the orbifold notation provided by the user. Our system can handle not only Euclidean orbifolds, but also spherical orbifolds and hyperbolic orbifolds. In addition, we provide an enumeration of two-dimensional kaleidoscopic orbifolds based on the combination of the cardinality of the underlying polygon and the type of the universal cover. As part of the design system, we also enable the interactive construction of the universal cover of the orbifold as well as movement of the objects in the scene. The user can also generate high-quality photorealistic renderings of the scene, panorama, and animations with Mitsuba, which we have modified to account for the geodesics in spherical and hyperbolic geometry along which light travels.

Making rendering more efficient with the off-line rendering with Mitsuba is important, and we plan to investigate efficient spatial hashing data structures for the non-Euclidean spaces. In addition, the quality of the meshes used to represent the floor and the ceiling can impact the rendering speed, and we plan to explore optimal meshing structures for our purpose.

For future directions, we wish to expand our rendering system to arbitrary three-dimensional orbifolds. In addition, not all orbifolds are kaleidoscopic, and we would like to incorporate the visualization for non-kaleidoscopic orbifolds, such as the ones involving gliding reflections. Finally, we wish to explore the visualization of non-orientable orbifolds, whose universal cover is a non-orientable surface such as the Projective plane and the Klein bottle.

Supplemental Materials

The supplement materials of this paper include an appendix and a video.

Acknowledgments

The authors wish to thank our anonymous reviewers for their constructive feedback. We appreciate the help from Peter Oliver during video production. Botong Qu has provided valuable suggestions during the initial discussion phase. This work was supported in part by the NSF award (# 1619383).

References

[1] Irrlicht engine. `http://irrlicht.sourceforge.net/`. Created: 2016-03-18. 5

[2] N. Aigerman, S. Z. Kovalsky, and Y. Lipman. Spherical orbifold tutte embeddings. *ACM Trans. Graph.*, 36(4), jul 2017. doi: 10.1145/3072959.3073615 1, 2

[3] N. Aigerman and Y. Lipman. Orbifold tutte embeddings. *ACM Trans. Graph.*, 34(6):190:1–190:12, Oct. 2015. doi: 10.1145/2816795.2818099 1, 2

[4] N. Aigerman and Y. Lipman. Hyperbolic orbifold tutte embeddings. *ACM Trans. Graph.*, 35(6):217–1, 2016. doi: 10.1145/2980179.2982412 1, 2

[5] J. W. Anderson. *Hyperbolic geometry*. Springer Science & Business Media, 2006. doi: 10.1007/1-84628-220-9 8

[6] A. Basmajian and M. Zeinalian. Möbius transformations of the circle form a maximal convergence group, 2006. doi: 10.48550/arXiv.math/0603457 6

[7] P. Berger, L. A., and L. Velho. An image-space algorithm for immersive views in 3-manifolds and orbifolds. *Vis Comput.*, 31:93–104, 2015. doi: 10.1007/s00371-013-0913-2 2

[8] D. Bommes, H. Zimmer, and L. Kobbelt. Mixed-integer quadrangulation. *ACM Trans. Graph.*, 28(3), jul 2009. doi: 10.1145/1531326.1531383 1

[9] J. Conway, H. Burgiel, and C. Goodman-Strauss. *The Symmetries of Things*. Ak Peters Series. Taylor & Francis, 2008. doi: 10.1201/b21368 2

[10] J. H. Conway, P. G. Doyle, J. Gilman, and W. P. Thurston. Geometry and the imagination in minneapolis. *arXiv preprint arXiv:1804.03055*, 2018. doi: 10.48550/arXiv.1804.03055 5

[11] D. Cooper, C. D. Hodgson, and S. P. Kerckhoff. *Three-dimensional Orbifolds and Cone-Manifolds*, vol. Volume 5 of *MSJ Memoirs*. The Mathematical Society of Japan, Tokyo, Japan, 2000. doi: 10.2969/msjmemoirs/00501C020 3

[12] H. S. M. Coxeter. *Non-euclidean geometry*. Cambridge University Press, 1998. 3, 5, 6

[13] S. Giaccari and R. Volpato. A fresh view on string orbifolds. *Journal of High Energy Physics*, 2023(1), jan 2023. doi: 10.1007/jhep01(2023)173 1

[14] G. Greaves. Cyclotomic matrices over the eisenstein and gaussian integers. *Journal of Algebra*, 372:560–583, 2012. doi: 10.1016/j.jalgebra.2012.09.006 4

[15] A. J. Hanson. Visualizing quaternions. In *ACM SIGGRAPH 2005 Courses*, SIGGRAPH '05. ACM, New York, NY, USA, 2005. doi: 10.1145/1198555.1198701 2

[16] A. Hertzmann and D. Zorin. Illustrating smooth surfaces. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, p. 517–526. ACM Press/Addison-Wesley Publishing Co., USA, 2000. doi: 10.1145/344779.345074 1

[17] M. P. Hitchman. *Geometry with an introduction to cosmic topology*. Jones & Bartlett Learning, 2009. 5, 6

[18] J. F. Humphreys. *A Course in Group Theory*, vol. Volume 5 of *Oxford Science Publications*. Oxford University Press, 1996. 3

[19] W. Jakob. Mitsuba renderer, 2010. http://www.mitsuba-renderer.org. 8, 9

[20] F. Kälberer, M. Nieser, and K. Polthier. Quadcover - surface parameterization using branched coverings. *Computer Graphics Forum*, 26(3):375–384, 2007. doi: 10.1111/j.1467-8659.2007.01060.x 1

[21] P. Miley. *A Study of Orbifolds*. PhD thesis, University of British Columbia, 1998. 3, 7

[22] M. Nieser, J. Palacios, K. Polthier, and E. Zhang. Hexagonal global parameterization of arbitrary surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 18(6):865–878, 2012. doi: 10.1109/TVCG.2011.118 1

[23] M. Nieser, K. Poelke, and K. Polthier. Automatic generation of riemann surface meshes. In *Advances in Geometric Modeling and Processing, 6th International Conference, GMP 2010, Castro Urdiales, Spain, June 16-18, 2010. Proceedings*, pp. 161–178, 2010. doi: 10.1007/978-3-642-13411-1_11 2

[24] T. Novello, V. da Silva, and L. Velho. Global illumination of non-euclidean spaces. *Computers and Graphics*, 93:61–70, 2020. doi: 10.1016/j.cag.2020.09.014 2

[25] D. O'Shea. *The Poincare Conjecture: In Search of the Shape of the Universe*. Walker Books, 2007. doi: 10.5860/choice.45-0926 1

[26] J. Palacios, L. Roy, P. Kumar, C.-Y. Hsu, W. Chen, C. Ma, L.-Y. Wei, and E. Zhang. Tensor field design in volumes. *ACM Trans. Graph.*, 36(6), nov 2017. doi: 10.1145/3130800.3130844 1

[27] J. Palacios, H. Yeh, W. Wang, Y. Zhang, R. S. Laramee, R. Sharma, T. Schultz, and E. Zhang. Feature surfaces in symmetric tensor fields based on eigenvalue manifold. *IEEE Transactions on Visualization and Computer Graphics*, 22(3):1248–1260, 2016. doi: 10.1109/TVCG.2015.2484343 1

[28] J. Palacios and E. Zhang. Rotational symmetry field design on surfaces. *ACM Trans. Graph.*, 26(3):55–es, jul 2007. doi: 10.1145/1276377.1276446 1, 2

[29] B. Qu, L. Roy, Y. Zhang, and E. Zhang. Mode surfaces of symmetric tensor fields: Topological analysis and seamless extraction. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):583–592, 2021. doi: 10.1109/TVCG.2020.3030431 1

[30] L. Roy, P. Kumar, S. Golbabaei, Y. Zhang, and E. Zhang. Interactive design and visualization of branched covering spaces. *IEEE Trans. Vis. Comput. Graph.*, 24(1):843–852, 2018. doi: 10.1109/TVCG.2017.2744038 2

[31] L. Roy, P. Kumar, Y. Zhang, and E. Zhang. Robust and fast extraction of 3d symmetric tensor field topology. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):1102–1111, 2019. doi: 10.1109/TVCG.2018.2864768 1

[32] W. Thurston and S. Levy. *Three-dimensional Geometry and Topology*. Number v. 1 in Luis A.Caffarelli. Princeton University Press, 1997. doi: 10.1515/9781400865321 1, 2, 5, 6, 7

[33] J. van Wijk and A. Cohen. Visualization of the genus of knots. p. 72, 01 2005. doi: 10.1109/VISUAL.2005.1532843 2

[34] F. Williams and E. Zhang. Rendering kaleidoscopic scenes using orbifold theory. In *SIGGRAPH ASIA 2016 Technical Briefs*, SA '16. Association for Computing Machinery, New York, NY, USA, 2016. doi: 10.1145/3005358.3005368 8

[35] R. Zeller, O. Delgado-Friedrichs, and D. H. Huson. Tegula–exploring a galaxy of two-dimensional periodic tilings. *Computer Aided Geometric Design*, 90:102027, 2021. doi: 10.1016/j.cagd.2021.102027 8

[36] E. Zhang, J. Hays, and G. Turk. Interactive tensor field design and visualization on surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 13(1):94–107, 2007. doi: 10.1109/TVCG.2007.16 1