

Iterated Inversion System: An Efficient Algorithm to Visualize Kleinian Groups Based on Inversions

Kento Nakamura
Meiji University

1 Introduction

Kleinian group theory is one of the fields of mathematics studying Möbius transformation groups. Kleinian group theory is advanced by mathematicians in the nineteenth century. Felix Klein and his student, Robert Fricke studied Möbius transformation groups. Henri Poincaré named such groups *Kleinian Groups*. Moreover, composing Geometry, Algebra, and Analysis, Poincaré built foundations of Kleinian groups theory. Klein and Poincaré were rivals to study Kleinian group theory.

In the latter half of the twentieth century, according to the achievement of William Thurston, hyperbolic geometry is again realized as a part of modern mathematics, and Kleinian group is also studied from the viewpoint of modern geometry; Then the complexity of Kleinian groups is unraveled step by step by the technique of modern mathematics until at the present.

Möbius transformation group is well suited to visualization and experiment; Actually, Klein and his students also leave beautiful visualized images of a Kleinian group without a computer. There are aspects that mathematicians advance research from visualization and experiment.

After a computer appeared, various visualization and calculation are performed by a computer. Visualized images of Kleinian groups often have beautiful fractal shape. Thus, people who love computer graphics are fascinated by complicated shapes of fractals. For example, *Fractalforums*¹ community gathers many fractal enthusiasts and discusses fractals. The fractals generated by Kleinian group theory also come up for discussion. In a sense, Kleinian group theory is an interdisciplinary area between mathematics and arts.

Mumford, Series, and Write wrote a book called *Indra's Pearls* [1]. This book is written for non-mathematician and contains explanations about Kleinian group theory, many beautiful visualized images, and methods of visualization. Thus, not only math maniacs but also programmers enjoy the book.

However, the book deal with only a small part of Kleinian groups. It is difficult to visualize all of the types of Kleinian groups in real time in spite of the growth of the performance of a personal computer because the number of combinations of the elements of the group exponentially increases,

Our final goals are to visualize all of the Kleinian groups in real time by personal computer and help us understand the Kleinian group theory intuitively with software. As the first step to the goals, we invent an algorithm called *Iterated Inversion System (IIS)*. IIS is an algorithm to render Kleinian group based on circle or sphere inversions. It visualizes not only three-dimensional Kleinian group but also four-dimensional Kleinian group. In this paper, we introduce the basic usage of the IIS algorithm and its applications.

2 Preparation

In this section, we introduce some mathematical terms and prerequisites to understand the IIS algorithm and basic usage of IIS.

2.1 Transformation Group Theory

In this paper, we use terms about Kleinian groups used in Indra's Pearls [1]. The word *group* represents an algebraic group which is central concept about group theory. The algebraic group is a set which has multiplication, satisfies the

¹<https://fractalforums.org/>

associative law, has a unit, and has an inverse element for each element.

Also, a transformation group is an algebraic group consists of transformations on a topological set X . Here a transformation is a homeomorphism $f : X \rightarrow X$.

We will introduce the Möbius transformation group and Kleinian group as a subgroup of the Möbius transformation group. In this paper, we mainly consider the cases $X = \tilde{\mathbb{C}}$, which is the union of the complex set and the infinity, and $X = \tilde{\mathbb{R}}^3$, which is the union of \mathbb{R}^3 and the infinity. In our context, we suppose proper topologies on $\tilde{\mathbb{C}}$ by the complex topology of $P^1(\mathbb{C})$ and on $\tilde{\mathbb{R}}^3$ by one-point compactification from \mathbb{R}^3 respectively.

We will introduce Möbius transformation group on $\tilde{\mathbb{C}}$ and $\tilde{\mathbb{R}}^3$. First, the two-dimensional Möbius transformation group $Möb(\tilde{\mathbb{C}})$ is the set of linear fractional transformations $f(z) = \frac{az+b}{cz+d}$, where a, b, c , and d are complex numbers and they satisfy $ad - bc = 1$. Here we naturally extend a complex function $f(z)$ into a transformation on $\tilde{\mathbb{C}}$. A linear fractional transformation is a conformal orientation-preserving homeomorphic map. See also section 2.3. From another point of view, a Möbius transformation is the composition of an even number of circle inversions. About inversions, see section 2.2. In the same manner, we consider the three-dimensional Möbius transformation group $Möb(\tilde{\mathbb{R}}^3)$. In the three-dimensional case, we define a Möbius transformation as the composition of an even number of sphere inversions.

2.2 Inversions in Circles or Spheres

In the two-dimensional cases, it is well known that Möbius transformations on $\tilde{\mathbb{C}}$ are composed of an even number of circle-inversions and line-symmetries. Here, if a circle is centered at $C \in \mathbb{C}$ and it has a radius $R \in \mathbb{R}$ ($R > 0$). Then the formula of the inversion along the circle is given by $f(z) = \frac{R^2}{z-C} + C$. Since $f \circ f$ is the identity map, the inversion map is a homeomorphism on $\tilde{\mathbb{C}}$. Remark that the inversion is an orientation-reversing conformal map on $\tilde{\mathbb{C}}$.

We often distinguish the inversion from a line symmetry map. But in this context, by interpreting the line on the complex plane as a circle centered at the infinity, we may regard the line symmetry as a kind of the inversion. In the sequel, we will call inversion both of circle-inversions and line-symmetries. In this category, any inversion is orientation-reversing, conformal, and homeomorphic on $\tilde{\mathbb{C}}$.

Here we will make a short note on the Jacobian of inversion. If the inversion is a line symmetry, the determinant of the Jacobian is -1 . Otherwise, if the inversion is a circle-inversion, remarking that a circle inversion preserves any angles, we obtain that Jacobian mapping is given by the composition of multiplication by a complex constant and complex conjugate. Thus the determinant of the Jacobian at a point $P \in \mathbb{C}$ is given by the following formula:

$$\text{Det(Jacobian)} = R^4 / \text{distance}(P, C)^4$$

Here the Jacobian is a composition of a scaling and a rotation and a complex conjugate, and we obtain the following.

$$\text{scaling factor at } P = R^2 / \text{distance}(P, C)^2.$$

In a similar way as the definition of a circle-inversion, a sphere-inversion can be defined as follows. Fix a sphere C in \mathbb{R}^3 . Let its center be C and its radius be R . A sphere-inversion I_C is a homeomorphism on $\tilde{\mathbb{R}}^3$ determined by a map from a point P to a point $Q = I_C(P)$ such that (i) Q is on the half line CP and that (ii) $\overline{CP} \cdot \overline{CQ} = R^2$. As above, the definition of a circle-inversion can be extended to a plane-symmetry in the same way as a line-symmetry in the two-dimensional case.

2.3 Möbius Transformations and the Hyperbolic Geometry

First, we review the definition of a group action. Let G be a group and X be a set. A map $G \times X \rightarrow X : (g, x) \mapsto g \cdot x$ is called a G -action on X if it satisfies the following conditions.

- (1) $e \cdot x = x$ for all $x \in X$. (Here, e denotes the unit of G .)
- (2) $(gh) \cdot x = g \cdot (h \cdot x)$ for all $g, h \in G$ and for all $x \in X$.

In this study, we handle $PSL_2\mathbb{C}$ -action on $\tilde{\mathbb{C}}$. Here $PSL_2\mathbb{C}$ is the projective space of 2×2 complex matrices A with $\det(M) = 1$. Möbius transformation is defined as a linear fractional transformation $f(z) = \frac{az+b}{cz+d}$ for complex variable z where constants a, b, c , and d are complex numbers and satisfy $ad - bc = 1$.

If we consider a map $\varphi : \frac{az+b}{cz+d} \mapsto \begin{pmatrix} a & b \\ c & d \end{pmatrix}$, from the set of linear fractional transformations to $PSL_2\mathbb{C}$. This map φ is a group isomorphism, and it gives an induced action of $PSL_2\mathbb{C}$ on $\tilde{\mathbb{C}}$. In the sequel, we regard the action of linear

fractional transformations and that of $PSL_2\mathbb{C}$ without distinction. For more details about Möbius transformation, refer [1][2].

Möbius transformation is deeply related to the hyperbolic geometry. When all of the coefficients in $f(z) = \frac{az + b}{cz + d}$ are real number, $f(z)$ preserve upper-half plane. Moreover, $f(z)$ preserves the hyperbolic metric $\frac{dz^2}{(Imz)}$ on the upper-half plane. Such linear fractional transformations correspond to an element of $PSL_2\mathbb{R}$. Thus the $PSL_2\mathbb{R}$ -action on the upper-half plane is the isometric transformation group of the hyperbolic plane \mathbb{H}^2 .

When the coefficients a, b, c , and d are complex numbers, the action can be thought as an isometric transformation group on the hyperbolic space \mathbb{H}^3 . We consider the uppe-half model of the hyperbolic space \mathbb{H}^3 . In fact,

$$\mathbb{H}^3 = \{(z, t) \in \mathbb{C} \times \mathbb{R} \mid t > 0\} \cup \{\infty\}$$

is the upper-half model with the hyperbolic metric $\frac{dz^2 + dt^2}{t^2}$ and $\hat{\mathbb{C}} = \{(z, 0) \mid z \in \mathbb{C}\} \cup \{\infty\}$ is the set of infinity. The two-dimensional Möbius transformation group acts on $\hat{\mathbb{C}}$ naturally, and it is well known that this action can be extended to an action on the upper-half model as isometric transformations. This extension is called '*Poincaré extension*'. As above, researching structures of the Möbius transformation groups is heavily related to studying three-dimensional hyperbolic geometry and hence hyperbolic manifolds. For more details, refer [3][4].

In this paper, we also consider the three-dimensional Möbius transformation on $\hat{\mathbb{R}}^3 = \mathbb{R}^3 \cup \{\infty\}$. The definition of the three-dimensional Möbius transformation in the two ways as follows. One is the simple extension of the two-dimensional version. A Möbius transformation is defined as a composition of an even number of sphere-inversions and plane-symmetry. The other is the representation of the isometry group of the four-dimensional hyperbolic space. To represent three-dimensional Möbius transformation using matrix, we consider a quaternion 2×2 matrix group called $S p^k(1, 1)$. About this topic, refer [5][6].

Three-dimensional Möbius transformations are deeply related to the four-dimensional hyperbolic space. In fact, the upper-half space model of the four-dimensional hyperbolic space \mathbb{H} is homeomorphic to the four-dimensional open ball and its boundary (the infinity point set of the hyperbolic space) is $\hat{\mathbb{R}}^3 = \mathbb{R}^3 \cup \{\infty\}$. Using Poincaré extension, a sphere-inversion in $\hat{\mathbb{R}}^3$ is extended to an (orientation-reversing) isometric transformation of \mathbb{H}^4 . Thus the composition of an even number of sphere-inversions corresponds to an orientation-preserving isometry, and it is known that any isometry of \mathbb{H}^4 is given in this way. In this point of view, three-dimensional Möbius transformations are deeply related to the four-dimensional hyperbolic space and four-dimensional hyperbolic manifolds.

2.4 Classification of Möbius Transformations

Excluding the identical map, we will classify Möbius transformations $f(z) = \frac{az + b}{cz + d}$ into three types. The types are *elliptic*, *parabolic*, and *loxodromic*. It is well-known that the standard form of $PSL_2\mathbb{C}$ is either one of $\begin{pmatrix} 1 & \lambda \\ 0 & 1 \end{pmatrix}$, λ is a non-zero constant,) or $\begin{pmatrix} \lambda & 0 \\ 0 & \lambda^{-1} \end{pmatrix}$, λ is a non-zero constant,) from a famous theorem in linear algebra. If $X = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ is conjugate to the former one, then $f(z)$ is called a parabolic transformation. If $f(z)$ is parabolic, the number of fixed point is one, and the matrix X satisfy $\text{tr}^2 X = 4$.

When the standard form of X is $\begin{pmatrix} \lambda & 0 \\ 0 & \lambda^{-1} \end{pmatrix}$, we have the following two cases. One is the case $|\lambda| = 1$ and the other is the case $|\lambda| \neq 1$. If $|\lambda| = 1$ then $f(z)$ is called an elliptic transformation. When $f(z)$ is elliptic, the fixed point set of $f(z)$ consists of two distinct points, and $f(z)$ gives a rotation around these fixed points. And $f(z)$ is elliptic if and only if $\text{tr}^2 X > 4$. Remark that if the order of a Möbius transformation $f(z)$ is finite then it is elliptic.

If the standard form is $\begin{pmatrix} \lambda & 0 \\ 0 & \lambda^{-1} \end{pmatrix}$ with $|\lambda| \neq 1$, then $f(z)$ is called a loxodromic transformation. In a special case, if λ is a real number other than ± 1 , the transformation is called a *hyperbolic* transformation too. When $f(z)$ is loxodromic the number of the fixed point set of $f(z)$ is two. One of the fixed points is called an attracting fixed point and the other is called a repelling fixed point. This is because the orbit space of $f(z)$ of a general point z_0 has two limit point and one of two is the limit of $f^n(z_0)$ for $n \rightarrow \infty$, and the other is the limit of $f^n(z_0)$ for $n \rightarrow -\infty$. Also, $f(z)$ is loxodromic if and only if $\text{tr}^2 X < 4$.

The set of three-dimensional Möbius transformations also have a similar classification into six types. In three-dimensional case, each type ‘parabolic,’ ‘elliptic,’ ‘loxodromic’ has a modifier *simple* and *compound*. Thus there are six variations of classification types, for example, “simple elliptic” or “compound loxodromic.” For detailed information of classification of three-dimensional Möbius transformation, see chapter 4.3.2.

2.5 Kleinian Groups

A Kleinian group is a group originated from the name of Felix Klein who is a famous mathematician in the nineteenth century. A group G is a Kleinian group if it satisfies the following two conditions. First, G is a subgroup of the Möbius transformation group. Second, the G -action on \mathbb{H}^3 , is properly discontinuous. Here a G -action on \mathbb{H}^3 is properly discontinuous if and only if for any compact set $K \subset \mathbb{H}^3$ there are an only finite number of elements $g \in G$ such that $g(K) \cap K \neq \emptyset$.

When the G -action is properly discontinuous, the ϵ neighborhoods of points in the orbit space are disjoint for a positive ϵ . This follows that there exists a fundamental domain with positive volume in \mathbb{H}^3 . Because a Möbius transformation gives an isometric transformation of \mathbb{H}^3 , a Kleinian group gives a hyperbolic three-manifold as a quotient space of the group. Basic properties of a Kleinian group are described in chapter 2 of [2].

In this paper, we consider a Möbius transformations in a broad sense, that is, a map which is the composite of any number of inversions. In order to distinguish an original Möbius transformation, we call it *an extended Möbius transformation*. The composite of odd numbers of inversions is an orientation-reversing map represented by $f(z) = \frac{a\bar{z} + b}{c\bar{z} + d}$, where a, b, c , and d are complex numbers and $ad - bc = 1$.

An extended Möbius transformation induces an isometric mapping (not necessarily to be orientation-preserving) on the three-dimensional hyperbolic space \mathbb{H}^3 by Poincaré expansion. We call a properly discontinuous subgroup of the extended Möbius transformation group *an extended Kleinian group*.

2.6 Limit Set

Let G be a Kleinian group. For a general point $z_0 \in \hat{\mathbb{C}}$, a point $p \in \hat{\mathbb{C}}$ is a limit point of the orbit space of z_0 if and only if there exists a sequence $\{g_i\} \subset G$ such that $\lim_{i \rightarrow \infty} g_i(z_0) = p$. We call the closure of all limit points of an orbit space Gz_0 the *limit set* of G . The limit set is denoted by $\Lambda(G)$. Some of the properties are very useful to visualize the limit set of a Kleinian group.

- The limit set does not depend on a general point z_0 . That is, the closure of all limit points of any orbit space is $\Lambda(G)$.
- The limit set is preserved by G -action. That is, the image of action $g \in G$ of a point $p \in \Lambda(G)$ also is included in the limit set $\Lambda(G)$.
- For any element $g \in G$, the fixed point of g is contained in the limit set.

Much more properties of the limit set is also described in chapter 2.4.1 of [2].

Here we have some concepts to describe limit points of an orbit. One is an algebraic limit and the other is a geometric limit. When we fix a set of generators of a Kleinian group G , we represent any elements in G as a word of these generators and their inverses. Let a loxodromic element g be a word $a_1a_2 \cdots a_r$. Suppose that a point sequence $\{(a_1a_2 \cdots a_r)^n \cdot z_0\}$ converges to a point p in the limit set. This limit point p is one of the fixed points of g . Thus we consider a formula $p = (\overline{a_1a_2 \cdots a_r})z_0$, where $\overline{a_1a_2 \cdots a_r}$ is a circulating decimal of generators (and their inverses.) However $\overline{a_1a_2 \cdots a_r}$ is not an element in G , supposing an abstract infinite word, we may consider a point on the limit set as a point in an orbit. This way of thinking is called an algebraic limit of G . On the other hand, the original definition of a limit point is obtained in a geometrical way of thinking. Thus such limit point is also called a geometric limit.

3 Visualization of Kleinian Groups

3.1 Basic Methods for Visualization

In this sub-section, we introduce basic methods for visualizing extended Kleinian groups. We explain a simple example of an extended Kleinian group generated by four inversions.

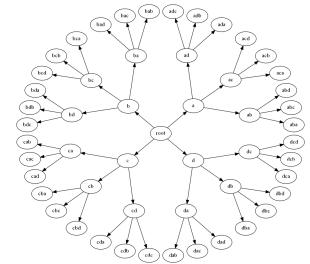


Figure 1: *Cayley Graph.*

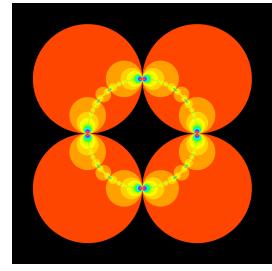


Figure 2: *Orbit of the disks.*

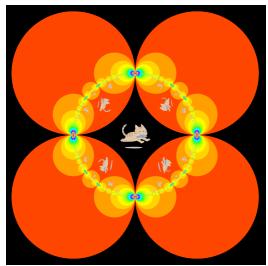


Figure 3: *The orbit of the cat.*

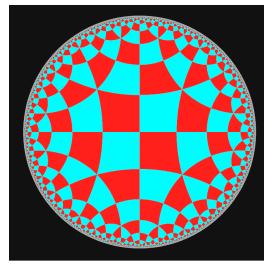


Figure 4: *Hyperbolic Tessellation.*

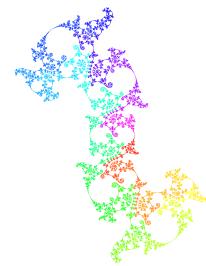


Figure 5: *Limit set of the Kleinian group.*

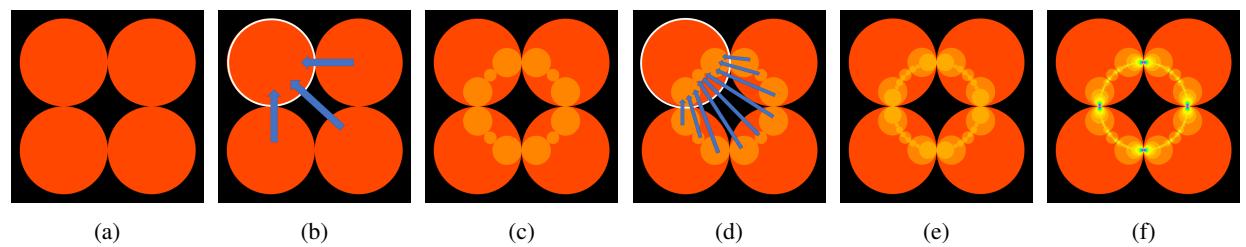


Figure 6: *The process of rendering the orbit of disks.*

First of all, to visualize a group, we consider a *Cayley graph*. A Cayley graph is defined as follows. Let G be a group, and we fix a set of generators. A node of a Cayley graph is an element of G , and an edge of a Cayley graph connects two nodes which are related by right-multiplication of a generator. For an easy example, we assume a group generated by four involutions, which is a map whose square coincides to the identity. In short, it is the following group.

$$G = \langle a, b, c, d \mid a^2 = b^2 = c^2 = d^2 = \text{id} \rangle$$

The Cayley graph of the group G is shown in Figure 1. The element of the group is a node. Four edges correspond to a, b, c , and d are emitted from each node. Because a, b, c , and d are involution, $a = a^{-1}$, and an edge corresponds to an inverse map is not needed. In Figure 1, we put the identity element id to the center and four edges are stretching symmetrically. In a Cayley graph, the graph distance from the identity to a node gives a word representing the node.

We can find that the number of words of the same length grows multiplied by three, that is to say words with length one is four, the number of words with length two are twelve, the number of words with length three are thirty-six, and the number of words with length n are $4 \cdot 3^{n-1}$.

The visualized image is Figure 2. In this image, we draw four circles on the plane, and we call their inversion a, b, c , and d respectively. Four circles come in touch but do not cross, and hence there is no relational expression between a, b, c , and d . Thus, the extended Möbius transformation group generated by these four inversions is group isomorphic to above G . In Figure 3, the cat is drawn in the center is moved by inversions, and “the orbit space of the cat” is visualized. We can find that the orbit space of the cat corresponds to nodes of the graph.

In order to draw the orbit space of the cat, there is breadth first search algorithm. In the following, we explain breadth first search algorithm. First of all, we draw the cat in the center, and next, we draw transformed cats by the words whose length is one, and we draw transformed cats by the word whose length is two. We continue iterating these processes. This is breadth first search algorithm. The tessellation shown in Figure 4 is also drawn by breadth first search algorithm.

Visualization by this algorithm is easy for us to implement and understand. We can roughly understand the actions of the groups. However, this method has a fault because the computational complexity is easy to increase and it takes too much time.

Also, there is another method of traversing the graph. It is called depth first search. Using this algorithm, we can draw only the limit set directly. See Figure 5. This shows an only the limit set of a Kleinian group. However, we do not use this algorithm in this paper. For more details of the above methods, read [1].

3.2 Iterated Inversion System (IIS)

To solve the problems we discussed in the previous sub-section, we focus circle inversions and invent an efficient algorithm to visualize the circle inversion fractals shown in Figure 6(f). The algorithm is called *Iterated Inversion System (IIS)*. It can visualize not only two-dimensional circle inversion fractals but also three-dimensional sphere inversion fractals.

The fractal in Figure 6(f) shows the orbit of the first four circles in Figure 6(a). It is also a Kleinian group composed of four circle inversions. Especially, it is called a *fuchsian group* when the limit set shapes into a perfect circle. On the other hand, the group called *quasi-fuchsian group* when the limit set shapes into a closed curve excluding a perfect circle. The process of generation of circle inversion fractals is as follows.

1. We need some disjoint disks to obtain circle inversion fractals. For example, we assume there are four orange disks as shown in Figure 6(a). We call orange disks *initial disks* and their boundary *initial circles*.
2. First of all, we focus on the white circle in Figure 6(b). The inversion in the white circle moves the other three disks into the interior of the white circle.
3. After we apply each inversion in the initial circle to the outer disks, we obtain twelve small disks. They are shown in Figure 6(c).
4. Next, we invert the twelve small disks in the initial circles. The inversion in the white circle moves the outer nine small disks into the interior of the white circle as shown in Figure 6(d). Each inversion in the initial circle generates smaller disks, and we obtain Figure 6(e).
5. We continue iterating these processes, that is, we continue applying each inversion in the initial circle to resulting smaller disks. Finally, we get Figure 6(f).

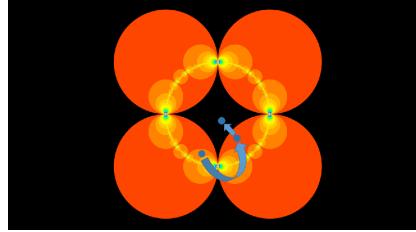


Figure 7: Orbit of blue point by IIS.

Algorithm 1 Iterated Inversion System (IIS.)

```

Require: count = 0 and coordinates = position determined by pixel
for  $i = 0$  to MAX_INVERSION do
    inOutside  $\leftarrow$  true
    for each Map  $G$  in circles do
        if  $G$  is available to coordinates then
            coordinates  $\leftarrow G(\text{coordinates})$ 
            INCREMENT count
            inOutside  $\leftarrow$  false
        end if
    end for
    if inOutside then
        BREAK for
    end if
end for
RETURN count

```

3.2.1 Two Dimensional IIS

IIS computes the depth of the circles point by point. Thus, we can perform parallel processing and render the images efficiently. All of the images in this paper are rendered using *OpenGL Shading Language (GLSL)*.

IIS can be applied to each point on the plane and computes nesting depth of the disk which contains the point. The process of the algorithm is as follows. First of all, if the point is contained in initial disks, we invert the point in the boundary circle of the disk. We continue applying inversions until the transformed point is in the outside of the initial disks. Figure 7 shows the orbit of the blue point transformed by iterations of inversions.

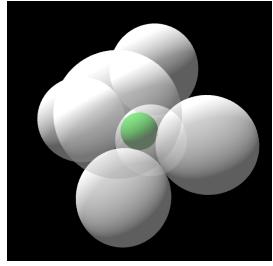
Furthermore, a point actually at the limit set never reaches the outside. So, we have to determine the maximum number of iterations in advance to prevent the algorithm from running indefinitely. The points except for the limit set are guaranteed that they are transformed to outside because inversions are involution.

Pseudo-code of IIS is shown in Algorithm 1. Later in section 4.3, we will introduce generators other than simple inversions. Thus, we consider a map $\gamma : \mathbb{C} \rightarrow G$ such that $\gamma(p)$ is the unit for a point p in the *fundamental domain* and that $G \rightarrow \gamma(p)$ is a composition of inversions for other points. The fundamental domain is the terminal area of transformations. In the above case, the black area consists of the outside and the boundary of all circles is the fundamental domain.

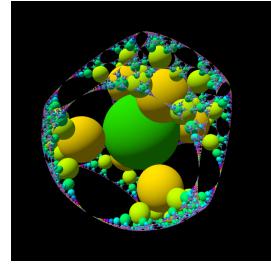
3.2.2 Three Dimensional Extension

In a similar manner to the two-dimensional algorithm, we extend the IIS to visualize three-dimensional Kleinian fractals. We extend the IIS algorithm by replacing circle inversions to sphere inversions easily, and we compute the nesting depth of the sphere voxel by voxel.

We use *ray tracing* to visualize three-dimensional objects. Ray tracing computes an intersection between a ray and objects algebraically. When we consider three-dimensional sphere inversion fractals in a similar manner to circle inversion fractals, we can obtain an image of an infinite number of nesting spheres in a sphere. There are two ways to render the nesting spheres. First one is to draw the nesting spheres as transparent spheres. Second one is *volume*



(a) Generator



(b) The orbit of spheres

Figure 8: The orbit of the sphere inversion fractal.

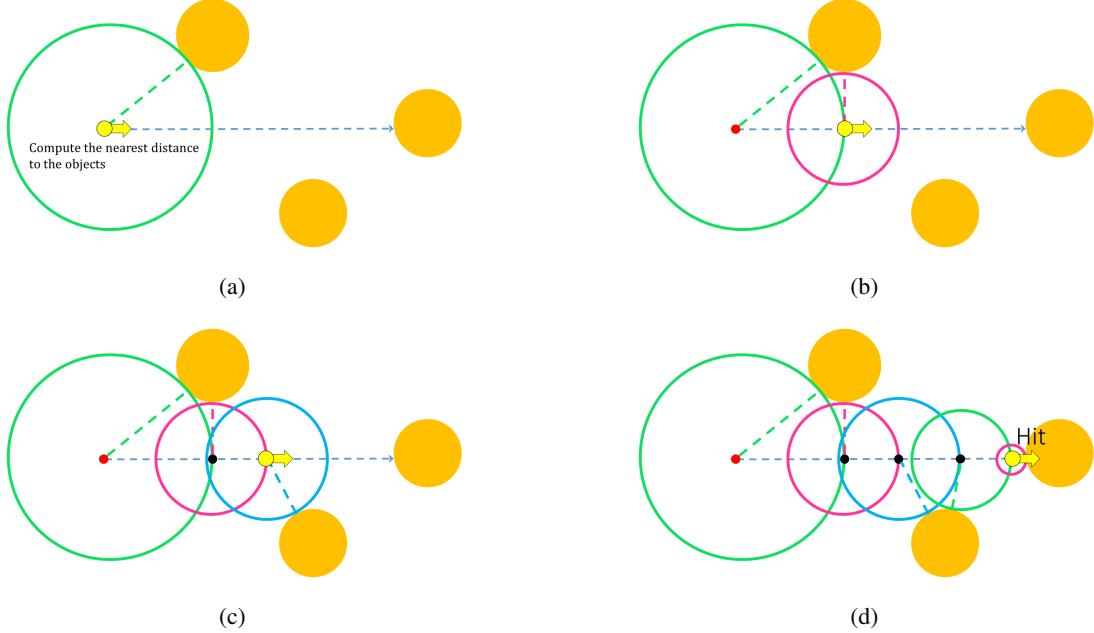


Figure 9: Sphere tracing.

rendering. However, it is difficult to render these two way of imaging efficiently, and visualized images are not helpful to study Kleinian groups.

Therefore we render the orbit of the sphere in a similar way to the two-dimensional circle inversion fractals by using IIS. See Figure 8(a). It shows white six inversion spheres and a green seed sphere. Figure 8(b) shows the orbit of the green sphere transformed by inversions in white spheres.

We use *sphere tracing* [7] to render three-dimensional fractals and orbit of the seed sphere. Sphere Tracing is one of the algorithms to render implicit surfaces using ray tracing. In the first place, we work a ray as something like a vector. We set the origin of the ray to the position of the camera and direction of the ray to the direction to each pixel on the screen from the camera. Each pixel is colored according to the first object the ray hits. In the following paragraphs, we introduce ray tracing and sphere tracing.

In the regular ray tracing, we calculate the intersections between the ray and the object algebraically, but it is impracticable to compute the intersections to fractal objects. As an alternating method, in sphere tracing, we make the tip of the ray marching along the direction of the ray step by step. To check how far the tip of the ray is from the objects, we need a *distance function*. The distance function is a function which returns the minimum distance between a given point and the object. For example, a distance function of a sphere can be obtained easily. Let P be a tip of a ray, let C and R be the center and the radius of the sphere respectively. A distance function of sphere $f(P)$ is given by $f(P) = \text{distance}(P, C) - R$. If there are many spheres, we use the minimum of the distance functions of the spheres.

Figure 9 shows a sphere tracing algorithm. The yellow arrow is a ray and orange disks are the visible objects. First of all, in Figure 9(a), we use the distance function and find the minimum distance between the tip of the ray and the disks. Next, we put the tip forward and apply distance function to the position of the ray as in Figure 9(b). Again, we apply the distance function and make the tip marching, and we obtain Figure 9(c). We continue iterating this processes until the value of the distance function is less than a small positive threshold as shown in Figure 9(d) or sufficiently large.

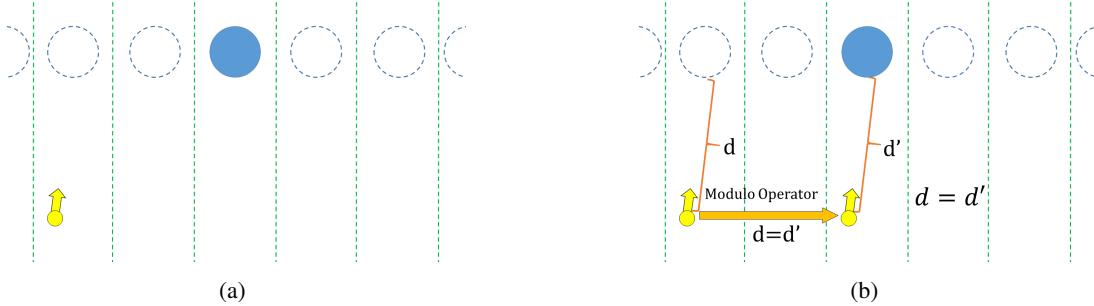


Figure 10: *Fold space by modulo operator*.

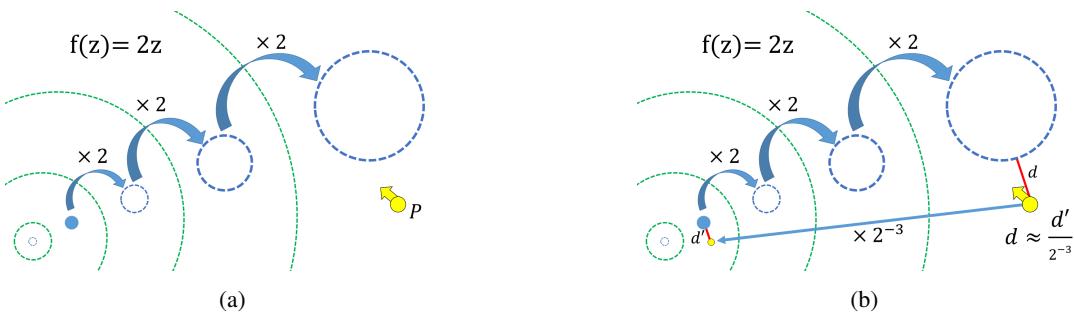


Figure 11: *Fold space by scaling*.

However, in regard to fractal rendering, it is difficult to get an actual distance from a given point to the object. So, we use a lower estimated distance as a return value of the distance function. The technique to approximate a distance is

called *distance estimation*. For more details about fractal rendering and distance estimation, see also the blog post² by Christensen.

Before we introduce details of distance estimation, we show a well-known technique to render many objects in sphere tracing. Instead of preparing distance functions for many objects lined up in a row, we can use modulo operator to get the distance to the lined up objects.

See Figure 10(a). We assume there are the blue disk, the yellow ray, and dotted circle and lines. Now, we want to draw all of the dotted circles. So, we want the minimum distance between the tip of the ray and dotted circles. If we only know the position of the tip of the ray and the blue disk, then how can we get the minimum distance? Thus, we assume the nearest circle to the ray is in the same dotted region, and we fold up the regions using the modulo operator. We can measure the distance to blue disk as in Figure 10(b). Finally, we can obtain the minimum distance to line upped disks.

Next, we consider scaling example. See Figure 11(a). There are a yellow ray and a blue disk, and orbit of scaling as dotted blue circles. The green dotted lines show scaling regions. We want to get the minimum distance between dotted blue circles and the tip of the ray. If we only know the coordinates of the original blue disk and tip of the ray, then how can we get the minimum distance? We assume that the nearest sphere to the tip of the ray is in same regions. Thus, we scale the tip of the ray to the same region to the blue disk but, the distance between scaled ray and disk is also scaled. So, we correct the distance by the ratio determined by Jacobian (sometimes referred to as the Jacobian determinant) of scaling transformation. See Figure 11(b). If the scaling of the circle is $f(z) = 2z$, we divide the distance by 2^{-3} because the tip of the ray is moved to the three regions away from the original.

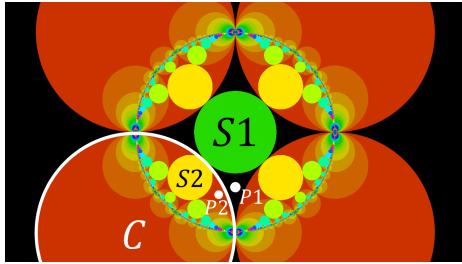


Figure 12: XY-slice image of Figure 8(b).

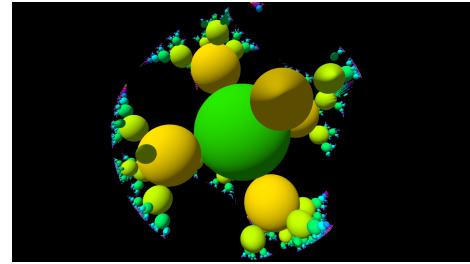


Figure 13: Artifact.

In a similar manner to scaling, we can render sphere inversion fractals. For the sake of simplicity, we consider a slice image of Figure 8. See Figure 12. This image shows the XY-slice of the orbit of spheres. Orange disks in the background are slices of the orbit of the initial inversion spheres. Slices of the orbit of the base sphere are colored in the same color as the orbit shown in Figure 8(b). In Figure 12, C is the white circle, the boundary of the initial inversion sphere. $S1$ is the base sphere and the inversion of $S2$ in the circle C . The white point $P1$ is the inversion of $P2$ in the circle C .

Now we assume that the tip of the ray is at $P2$. Let's calculate the minimum distance between $P2$ and the orbit of base spheres. The nearest sphere to $P2$ is $S2$. So, we have to calculate the distance between $P2$ and $S2$. We call the distance d . However, we do not know the center and radius of $S2$. So, we calculate d from a distance between $S1$ and $P1$. Inversions in spheres and Möbius transformations do not preserve Euclidean distance. Thus we use the Jacobian to estimate the distance. We accumulate the Jacobian of inversions by multiplying the Jacobian for every inversion.

Finally, we divide the distance between the base sphere and the point on the fundamental domain by the accumulated Jacobian, and we can get the approximated distance between the tip of the ray and the nearest sphere. For the above case, we get an inequality $d \geq \text{distance}(P1, S1)/\text{Jacobian}$. The formula gives a lower bound for spheres. For more details on the derivation of this estimation formula, see the blog post³ by Inigo Quilez.

We have one more thing to consider because the above calculation is a rough estimate. For example, if a given point is in the outer area of the orbit, the distance function returns unintentionally large distance, and the ray can pass through the real objects. This causes artifact shown in Figure 13. The fore part of the fractals is not rendered. In order to avoid this kind of trouble, we shrink the length of the estimated distance. Shrinking the length of the distance

²Mikael H Christensen, Distance Estimated 3D Fractals (Part I):
<http://blog.hvidtfeldts.net/index.php/2011/06/distance-estimated-3d-fractals-part-i/>

³Inigo Quilez, distance estimation: <http://www.iquilezles.org/www/articles/distance/distance.htm>

Algorithm 2 Distance Function.

```
Require: count = 0, d = MAX_DISTANCE, dr = 1.0, and coordinates = tip of the ray
for i = 0 to MAX_INVERSION do
    inFundamentalDomain ← true
    for each Map G in Maps do
        if G is available to coordinates then
            dr ← dr* (Jacobian of G(coordinates))
            coordinates ← G(coordinates)
            INCREMENT count
            inFundamentalDomain ← false
        end if
    end for
    if inFundamentalDomain then
        BREAK for
    end if
end for
for each BaseSphere S in BaseSpheres do
    d ← min(d, scalingFactor * (distance(coordinates, S .center) – S .radius) / (absolute value of dr))
end for
return d
```

increases the number of steps of sphere tracing, but we can eventually obtain the intersection of the ray and the spheres. The scaling factor is determined experimentally according to the size of the spheres.

The generalized pseudo-code for a distance function is in Algorithm 2.

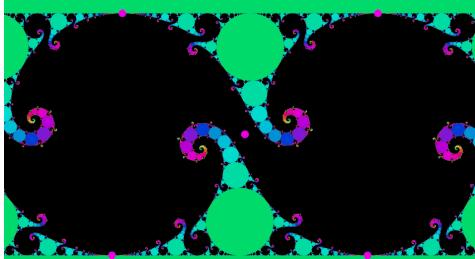


Figure 14: The limit set of the Kleinian groups with Maskit parametrization.

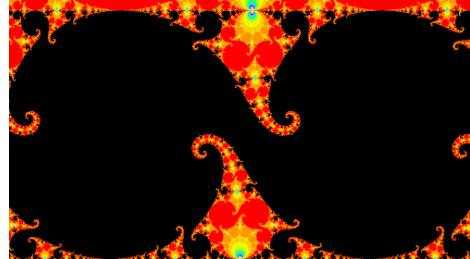


Figure 15: The limit set applied inversions of the circles.

3.3 Related Works

Before IIS is developed, there is an algorithm to draw a fractal called *pseudo-kleinian*, and there are visualization techniques and variations related to pseudo-kleinian. The algorithms are fast same as IIS, but pseudo-kleinian is not a Kleinian group. Pseudo-kleinian uses Möbius transformation. So, it is similar to IIS in terms of to draw three-dimensional fractals efficiently. Thus, there is a relationship between pseudo-kleinian and Kleinian groups. However, pseudo-kleinian differs from Kleinian groups which the mathematicians study.

Aaron Montag uses texture based approach to visualize a limit set of the Kleinian groups [8]. We prepare initial seed circle in the texture. Next, we apply generators of the group to each pixel. If the transformed pixel is on the seed circle or filled pixel, we fill the original pixel. We continue iterating this process; we obtain an image of the limit set of the group. This algorithm needs high-resolution texture, and it is difficult to extend this algorithm to three-dimension.

Jos Leys invented efficient algorithm to draw a Kleinian group with Maskit parametrization shown in Figure 14⁴. The pink points are control points of the fractal. Although IIS uses circle or sphere inversions, Jos Leys used Möbius

⁴http://www.josleys.com/article_show.php?id=221

transformations algebraically. He observes the orbit of the Maskit parametrization group and successfully discovers the algorithm to visualize Maskit groups. We can apply circle inversions to his figure, and we obtain Figure 15.

Martin von Gagern and Jürgen Richter-Gebert introduced an algorithm called *Reverse Pixel Lookup* [9] to render two-dimensional hyperbolic tiling. The algorithm is similar to ours, but we explore the method not only tiling but also other varieties of images, for instance, three-dimensional objects.

4 Application

In this section, we introduce the advanced usage of IIS.

4.1 Render Internal Area



Figure 16: *Edge of the circle inversion fractal.*

In two-dimensional circle inversion fractals, when all of the circles touch each other, the limit set divides the plane into two parts as shown in Figure 16(a). The image generated by four inversions of circles. After applying IIS, we only fill the pixel when the transformed point is moved into an inner part of the black area, and we obtain the inner part of the circle inversion fractals. See Figure 16(b).

4.2 Render Circles

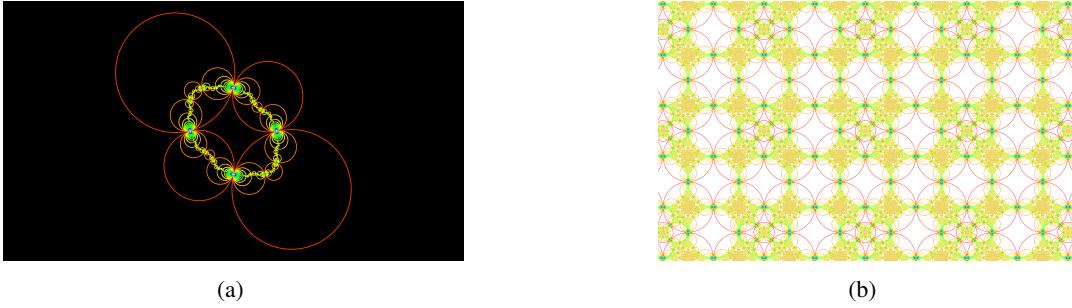


Figure 17: *Circle inversion fractal.*

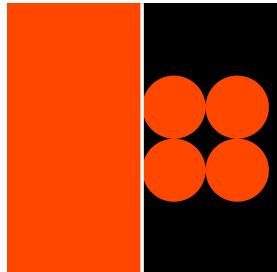
We can draw only edges of disks in circle inversion fractals as shown in Figure 17. We can estimate the distance from the circumference of the disks using Jacobian of circle inversions. When we apply circle inversions, we multiply and accumulate Jacobian of the inversions. When the transformed point is moved to the outside of the initial disks, we divide computed distance by accumulated Jacobian.

4.3 Geometrical Representation of Möbius Transformation Groups

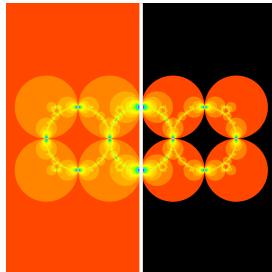
So far, we only use a simple circle or sphere inversions. Other interesting images can be generated using more complicated Möbius transformations. It is known that we can construct any Möbius transformation out of inversions. We can compose them by an even number of inversion. Thus, we can apply IIS to visualize fractals combining circle inversion fractals and extended Möbius transformation groups. Moreover, we can tweak the parameters of Möbius transformations by arranging geometrical objects like circles or lines on the plane. So, we can control parameters easily and intuitively.

The author is developing visualization software for inversion fractals. It is available at <https://schottky.jp>. This section is a revised version of [10].

4.3.1 Two Dimensional Generators

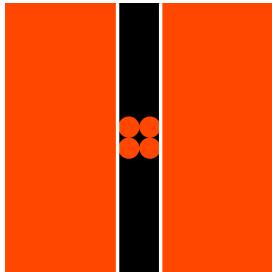


(a) Generator

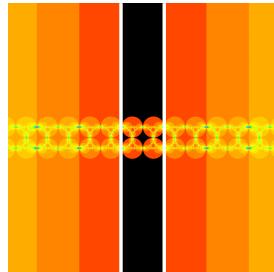


(b) Orbit

Figure 18: Inversion in the circle with infinite radius and four Schottky disks.

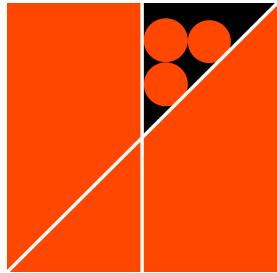


(a) Generator

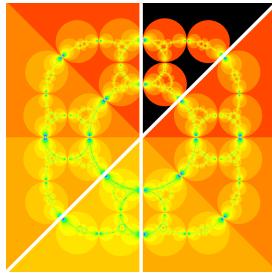


(b) Orbit

Figure 19: Parallel translation generator and four Schottky disks.

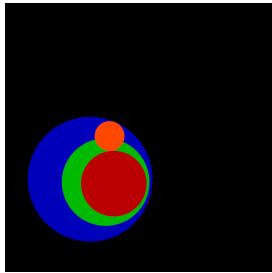


(a) Generator

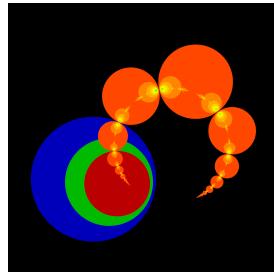


(b) Orbit

Figure 20: Rotation generator and three Schottky disks.



(a) Generator



(b) Orbit

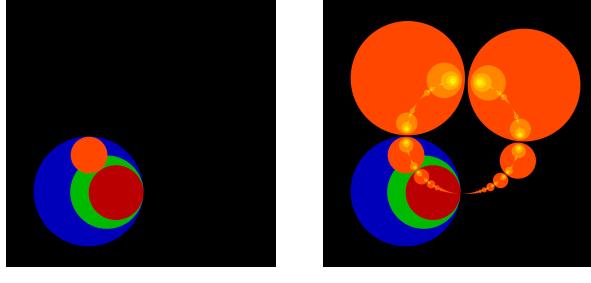
Figure 21: Hyperbolic generator and a Schottky disk.

Inversion in a Circle with Infinite Radius. An inversion in a circle with infinite radius is treated as a reflection over a borderline of half plane. See Figure 18(a). The four inversion circles are lying on the right side, and there is the orange region on the left side. The region is a half plane, that is, a disk with infinite radius. Its boundary is colored with a white line. The orbit is shown in Figure 18(b). As we can see, the circles are reflected over the white line.

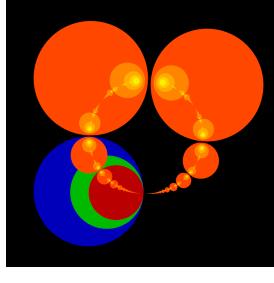
Parallel Translation. A composition of reflections over two parallel half planes facing each other generates a parallel translation. See Figure 19(a). There are two orange half planes on the right and left sides. The orbit is shown in Figure 19(b).

Rotation. A composition of reflections over two crossing half planes generates a rotation. The generator is shown in Figure 20(a). The two half planes are crossing. The rotation axis is a crossing point of white borderlines. The orbit is shown in Figure 20(b). It has a rotation symmetry and it is an elliptic transformation. Crossing angle of two borderlines should be rational angles; otherwise, the orbit will overlap each other.

Composition of Two Circles. Next, we use a composition of inversions in two circles. See Figure 21(a). There are one inversion disk and three regions colored with red, green, and blue. We call the boundary of red disk C_1 , the outer circle of green region C_2 , and let C_1' be the inversion of C_1 in C_2 . The outer circle of the blue region is C_1'' . The

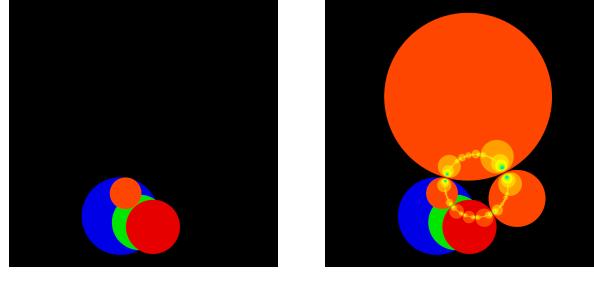


(a) Generator



(b) Orbit

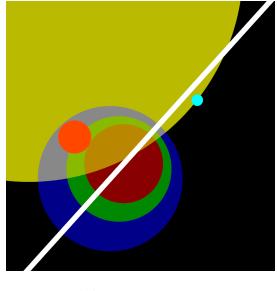
Figure 22: The orbit of Parabolic generator and a Schottky disk.



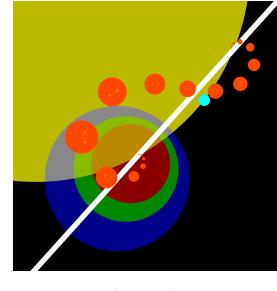
(a) Generator

(b) Orbit

Figure 23: Elliptic generator and a Schottky disk.



(a) Generator



(b) Orbit

Figure 24: Loxodromic generator and a Schottky disk.

generator is composed of C_1 , C_2 , and C_1' . While C_1 and C_2 have no intersection, the composition of inversions in C_1 and C_2 represents hyperbolic transformations⁵. The orbit is shown in Figure 21(b). The orbit of the disk converges to two fixed points. We compose the map G as follows. The prefix I represents an inversion, for example, I_{C_1} represents an inversion in C_1 .

$$G = \begin{cases} I_{C_2} \circ I_{C_1} & (\text{The point is inside of } C_1) \\ I_{C_1} \circ I_{C_2} & (\text{The point is outside of } C_1') \end{cases}$$

In the process of IIS, we can transform the point to the fundamental domain by applying G repeatedly. The fundamental domain of this type of generators is the blue and green area in Figure 21(a).

Then, we displace C_1 . When C_1 and C_2 are kissing as shown in Figure 22(a), this generator becomes a parabolic transformation⁶. The fixed points overlap each other, and the orbit converges to the point as shown in Figure 22(b).

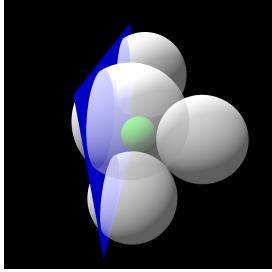
When C_1 and C_2 are crossing as in Figure 23, the generator becomes an elliptic transformation. The orbit is as shown in Figure 23(b). The disks are rotated around crossing line. Note that a crossing angle of C_1 and C_2 should be rational angles; otherwise, the orbit of the disks will overlap each other.

Loxodromic. We can twist the orbit by adding another two inversions. See Figure 24(a). The yellow disk and the white line are added to the hyperbolic generator. The white line is a line with two centers of C_1 and C_2 . We call the line L , the boundary of the yellow disk C_3 , and light blue point P . P is a user-defined control point, and the circle C_3 is determined by three points, one is the point P , and the others are inversions of P in C_1 and C_2 . L and C_3 are perpendicular to C_1 and C_2 . Thus, a composition of the reflection over L and the inversion in C_3 represents a rotation, and the orbit of the group is twisted as shown in Figure 24(b). This is a loxodromic transformation⁷. The map G is as follows.

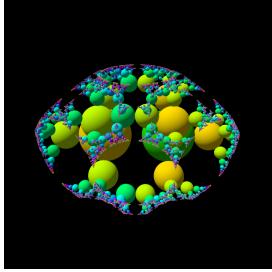
$$G = \begin{cases} (I_{C_2} \circ I_{C_1}) \circ (I_{C_3} \circ I_L) & (\text{The point is inside of } C_1) \\ (I_L \circ I_{C_3}) \circ (I_{C_1} \circ I_{C_2}) & (\text{The point is outside of } C_1') \end{cases}$$

⁵<https://www.shadertoy.com/view/MsScWW>⁶<https://www.shadertoy.com/view/XsBcDD>⁷<https://www.shadertoy.com/view/lSsyDW>

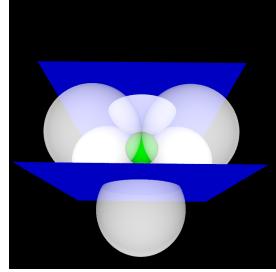
4.3.2 Three Dimensional Generators



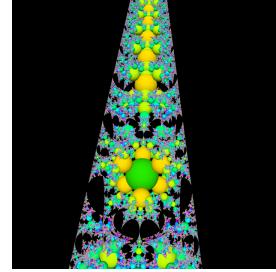
(a) Generator



(b) Orbit



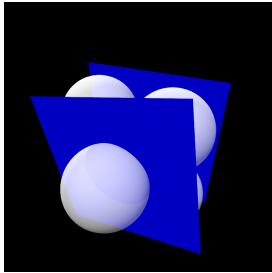
(a) Generator



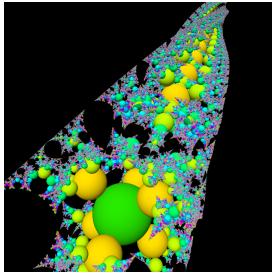
(b) Orbit

Figure 25: Inversion in the sphere with infinite radius, four Schottky spheres, and a base sphere.

Figure 26: Parallel translation generator, six Schottky spheres and a base sphere.

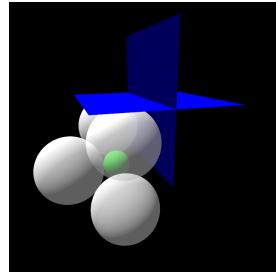


(a) Generator

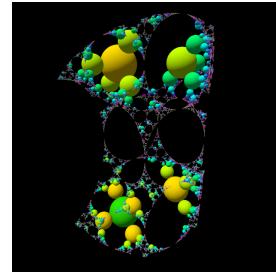


(b) Orbit

Figure 27: Compound parabolic generator, six Schottky spheres and a base sphere.

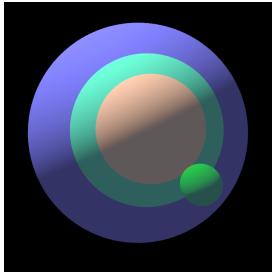


(a) Generator

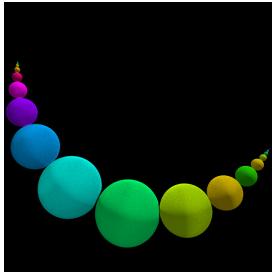


(b) Orbit

Figure 28: Rotation generator, four Schottky spheres, and a base sphere.

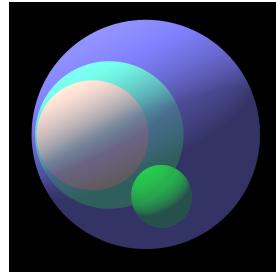


(a) Generator

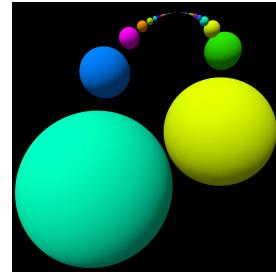


(b) Orbit

Figure 29: Hyperbolic generator and a base sphere.



(a) Generator



(b) Orbit

Figure 30: Parabolic generator and a base sphere.

Inversion in a Sphere with Infinite Radius. An inversion in a sphere with infinite radius is represented by a reflection through a plane. See Figure 25(a). There are four white inversion spheres, one green base sphere, and one blue plate. The blue plate is a part of a sphere with infinite radius. The orbit of the group is shown in Figure 25(b). The resulting orbit has reflection symmetry.

Parallel Translation⁸. A composition of inversions in two parallel planes represents a parallel translation along a normal vector of the planes. See Figure 26. This is a parabolic transformation. Moreover, in three-dimension, we can add a twist to the orbit. See Figure 27. The orbit is rotated around the normal vector of the planes for every translation. This operation is possible in three-dimensional space because we have gained a degree of freedom over two-dimensional space. The transformations yielding twisted orbits are called *compound parabolic* transformations.

⁸<https://www.shadertoy.com/view/lsjyzK>



Figure 31: Elliptic generator and a base sphere.

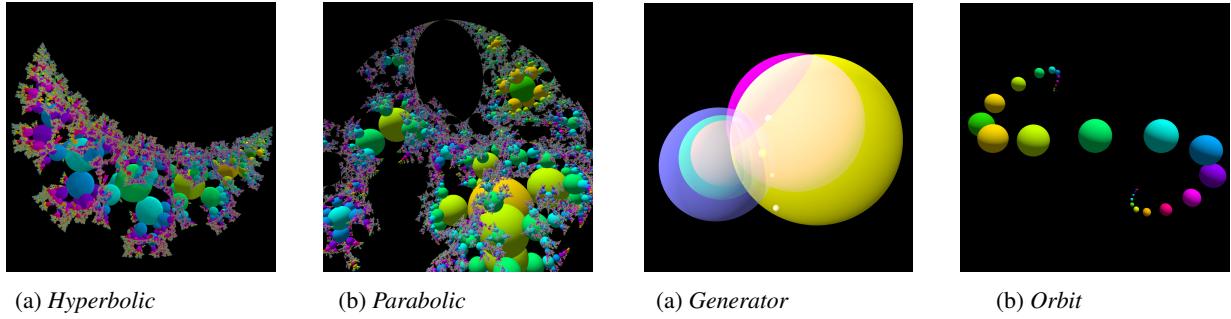


Figure 32: The orbit generated by a composition of two spheres and six Schottky spheres.

Figure 33: Compound loxodromic generator and a base sphere.

Rotation. A composition of reflections through two crossing planes generates a rotation. The axis of rotation is the intersection line of two planes. The generators and the orbit are shown in Figure 28.

Composition of Two Spheres⁹. We compose generators using two spheres. See Figure 29(a). We call the light red sphere S_1 , the light green sphere S_2 , and the blue sphere S_1' . The map G is as follows.

$$G = \begin{cases} I_{S_2} \circ I_{S_1} & (\text{The point is inside of } S_1) \\ I_{S_1} \circ I_{S_2} & (\text{The point is outside of } S_1') \end{cases}$$

While S_1 and S_2 have no intersection, the generator is hyperbolic transformation. The orbit of the base sphere is shown in Figure 29(b). When S_1 and S_2 come in contact with each other at one point as shown in Figure 30(a), it becomes a parabolic transformation. The orbit of spheres touches at the fixed point. It is shown in Figure 30(b).

When S_1 and S_2 are crossing as in Figure 31, the generator becomes an elliptic transformation. The orbit is as shown in Figure 31(b). Similarly to two-dimensional ones, the crossing angle of S_1 and S_2 should be rational angles; otherwise the orbit of the spheres will overlap each other.

Also, Figure 32 shows the example of the more complicated orbit of spheres generated by adding six inversion spheres to the group shown in Figure 29 and Figure 30.

Compound Loxodromic¹⁰. Finally, we add two spheres perpendicular to S_1 and S_2 . See Figure 33(a). We call pink sphere S_3 and yellow sphere S_4 , and there are three user-defined control points P , Q_1 , and Q_2 . S_3 and S_4 are determined by four points. Let P' and P'' be inversions of P in S_1 and S_2 . The spheres and the map G are as follows.

$$S_3 = \text{Sphere}(P, P', P'', Q_1) \quad S_4 = \text{Sphere}(P, P', P'', Q_2)$$

$$G = \begin{cases} (I_{S_4} \circ I_{S_3}) \circ (I_{S_1} \circ I_{S_2}) & (\text{The point is inside of } S_1) \\ (I_{S_2} \circ I_{S_1}) \circ (I_{S_3} \circ I_{S_4}) & (\text{The point is outside of } S_1') \end{cases}$$

The composition of inversions in S_3 and S_4 represents rotation. The twisted orbit shown in Figure 33(b) is analogous to the loxodromic transformations in two-dimension. Therefore, we call this generator *compound loxodromic*.

⁹<https://www.shadertoy.com/view/ldByDW>

¹⁰<https://www.shadertoy.com/view/MdjyRV>

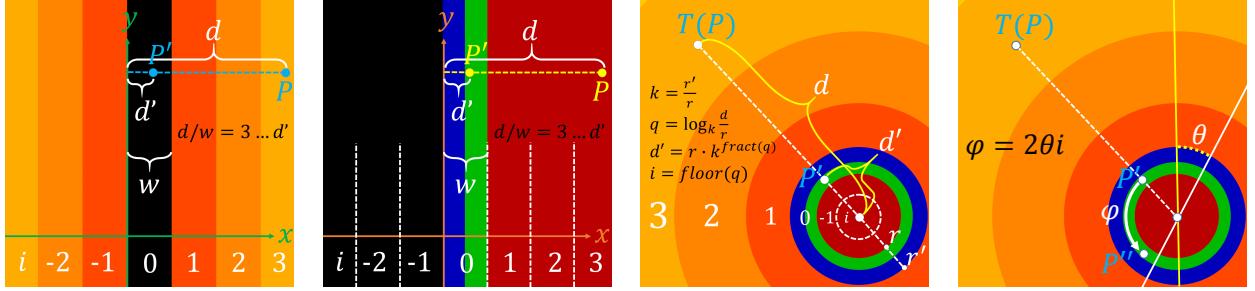


Figure 34: Parallel transla-
tion.

Figure 35: Inverted
parabolic generator.

Figure 36: Inverted hyper-
bolic generator.

Figure 37: Inverted loxo-
dromic genera-
tor.

4.3.3 Optimization

The maps of generators composed of some inversions can be optimized using proper conjugations. In this section, we introduce optimization techniques for two-dimensional generators. Optimized maps translate the points to the fundamental domain just one operation. Of course, we can also optimize three-dimensional generators in a similar manner to two-dimensional ones.

Parallel Translation¹¹. Suppose that there are two parallel lines similar to Figure 19. We can map the point using a remainder instead of applying inversions repeatedly. First of all, we conjugate a given point by parallel translations and rotations to make the parallel lines perpendicular to the X-axis and aligned with the Y-axis. The conjugated lines are shown in Figure 34. Let w be a distance between two lines, let i be an index of a series of bands as in Figure 34, let P be a given point, let P' be a mapped point, and let d and d' be distances from the Y-axis. We divide d by w and get the remainder d' and the quotient i . Finally, we calculate P' using d' and restore the P' to the original geometry.

Parabolic¹². Next, we consider parabolic transformations like Figure 22. Let T be an inversion of a circle centered on the fixed point (the contact point of $C1$ and $C2$.) Applying T to $C1$, $C2$, and $C1'$, we get parallel lines because the tangential point moves to the infinite point. We call the lines $TC1$, $TC2$, and $TC1'$, and they are shown in Figure 35. The red line $TC1$ and the blue line $TC1'$ represent a parallel translation. The process of the map is as follows. Firstly, we apply T to a given point P and obtain $T(P)$. Then we translate $T(P)$ in the same way as parallel translations and get P' . Finally, we apply $T (= T^{-1})$ to P' again.

Loxodromic¹³. Finally, we consider hyperbolic transformations like Figure 21. Let T be an inversion of a circle centered on one of the fixed points. We apply T to $C1$, $C2$, and $C1'$. We call the images of them $TC1$, $TC2$, and $TC1'$. They are concentric circles centered on the inverted image of the other fixed point. They represent the real scaling. See Figure 36. Let P be a given point, let P' be a mapped point, let r and r' be a radius of $TC1$ and $TC1'$, let d and d' be a distance from a center of $TC1$, let k be a scaling factor, let q be an exponential quotient, and let i be an index of series of circles. They are calculated as follows: $k = \frac{r'}{r}$, $q = \log_k \frac{d}{r}$, $d' = r \cdot k^{fracionalPart(q)}$, $i = floor(q)$. We calculate P' using d' . When there is a loxodromic transformation similar to Figure 24, we also apply T to L and $C3$, and we call them TL and $TC3$. See Figure 37. It represents a scaling by a complex number. The white line TL and the yellow line $TC3$ are crossing through a center of the concentric circles. After we get P' , we calculate P'' by applying a rotation to P' . Let θ be the angle between TL and $TC3$ and let φ be the rotation angle. φ is calculated as, $\varphi = 2\theta i$. The process is shown below. Firstly, we apply T to a given point and get $T(P)$. Next, we calculate P' using d' . If there are crossing lines, we rotate P' by φ and get P'' . Finally, we apply T to P' or P'' again.

4.4 Sphairahedra and Three-dimensional Fractals

In this section, we introduce an example of three-dimensional tiling using IIS. We discussed this topic in [11]. The paper is revised and included for this subsection. The author gathers images, animation, and renderer for the fractals at <https://sphairahedron.net>.

¹¹IIS Parallel Translation Example: <https://www.shadertoy.com/view/MtySzC>

¹²IIS Parabolic Transformation Example: <https://www.shadertoy.com/view/lIIVSzD>

¹³IIS Loxodromic Transformation Example: <https://www.shadertoy.com/view/4IGXDy>

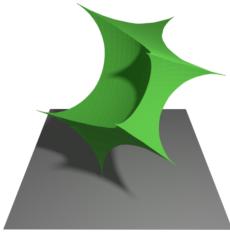
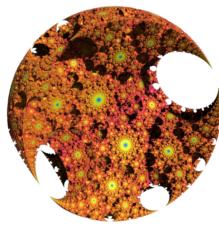
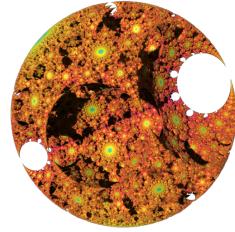


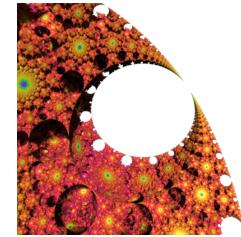
Figure 38: *Cube-type sphairahedron.*



(a)



(b)



(c)

Figure 39: *Images of a quasi-sphere rendered in different viewpoints.*

4.4.1 Introduction

In 2003, Kazushi Ahara and Yoshiaki Araki invented a new mathematical concept called *Sphairahedron* to introduce new kind of three-dimensional fractals [12].

Sphairahedron is a coined word combining two words *sphaira-* (a prefix that means 'spherical') and *-hedron* (a suffix comes from 'polyhedron.') In short, sphairahedron is a polyhedron with spherical faces. See Figure 38. It shows cube-type sphairahedron. As we can see, each face of the cube is a part of a sphere.

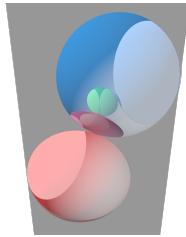
We can make a tiling pattern of sphairahedra using inversion about their spherical faces. In many cases, the boundary of the tiling converges to a three-dimensional fractal shape as shown in Figure 39. The union of all the tiles is mostly homeomorphic to a three-dimensional ball. Thus, this is called a *quasi-sphere*. Also, the boundary of the fractal is the limit set of the sphere inversion group.

Mathematically speaking, we consider a Coxeter-like group generated by the inversions in all of the spherical faces of the sphairahedron, and we obtain the tiling after transforming the original sphairahedron by each element of the group. Also, under some technical conditions, the group is called a *quasi-fuchsian group* and the limit set a *quasi-fuchsian fractal*.

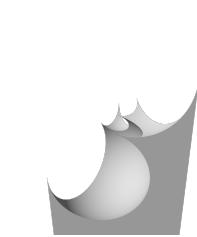
A quasi-fuchsian fractal is one of the three-dimensional fractals at an early era of visualizing fractals by computer. The video¹⁴ posted by Ahara and Araki shows the fractal based on a cube-type sphairahedron in the case of only quasi-fuchsian. However, there are sphairahedra based on other types of polyhedra, and we can allow tiles to self-intersect each other, that is, the group is not quasi-fuchsian. In this way, we can see more varieties of fractal patterns than proposed by Ahara and Araki.

The three-dimensional tiling of hyperbolic polyhedra is well known. In comparison to this, sphairahedra and their tiling patterns are originated from four-dimensional hyperbolic geometry. The rise in dimension brings complexity and difficulty for visualization, but visualized shapes have impressive structures. In this paper, we will introduce a variety of sphairahedra and fractal shapes generated by them.

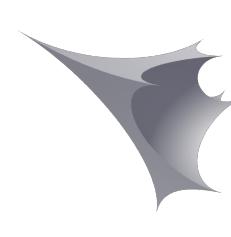
4.4.2 Sphairahedron



(a) A

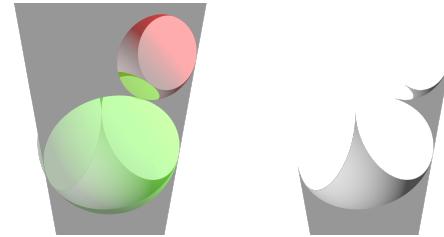


(b) *Infinite type*



(c) *Finite type*

Figure 40: *Sphairahedron.*



(a) A



(b) *Infi-
nite
type*

Figure 41: *Semi-sphairahedron.*

¹⁴Quasi-fuchsian fractals: <https://www.youtube.com/watch?v=3lcO9zRCv-4>

First of all, we will describe the definition of a sphairahedron. Let $S^3 = R^3 \cup \{\infty\}$ be a three-dimensional sphere and let D_1, D_2, \dots, D_p be some three-dimensional closed balls. We consider the complement A of the union of these balls, that is, $A = S^3 - (\overline{D_1} \cup \overline{D_2} \cup \dots \cup \overline{D_p})$. If A is composed of simply-connected two components; in other words, A has two connected components and the first homology group of each component is trivial, we call one side of A a sphairahedron.

The image in Figure 40(a) is an example of A . We hollow out the S^3 with six balls: we remove three half space (three balls with infinite radius,) from S^3 and obtain the prism of infinite length, and we scoop out the prism by the remaining three-colored transparent balls as in Figure 40(a). A is composed of two parts, and each of the components is simply connected. Since it has six faces, and these faces are arranged as those of faces of a cube, it is called a cube-type sphairahedron. Especially, the sphairahedron shown in Figure 40(b) is also called an *infinite type sphairahedron*, because one of the vertices of the sphairahedron is at the infinity. Similarly, the shape hollowed out by six finite balls in Figure 40(c) is called a *finite type sphairahedron*.

Moreover, we can loosen the definition of sphairahedron, that is, the case A has simply connected three or more components. Figure 41(a) shows an example of A with simply connected three components. It is the S^3 scooped out by five balls and a pentahedral prism type sphairahedron. We divide A so that one part of A has five faces as shown in Figure 41(b). We can regard the resulting shape as a singular case of a pentahedron, and we call it a *semi-sphairahedron*.

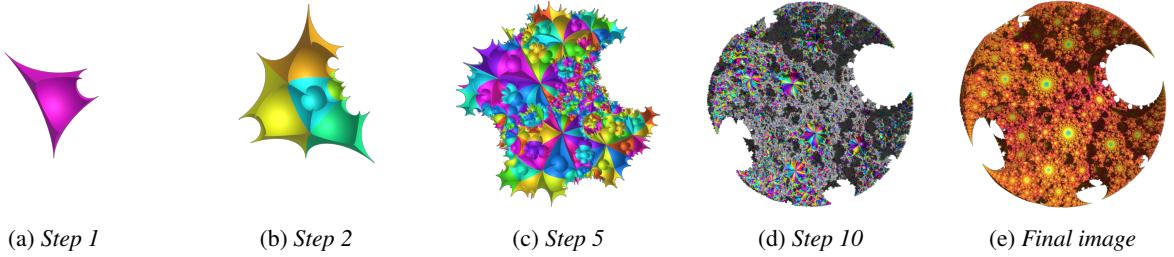


Figure 42: *Tiling of a finite cube-type sphairahedron.*

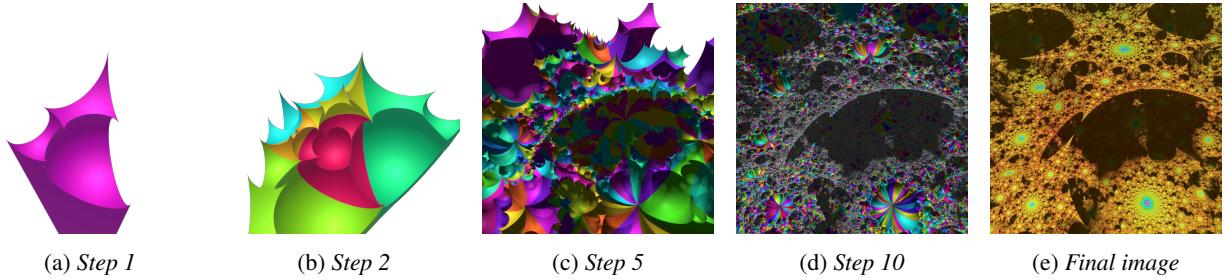


Figure 43: *Tiling of a cube-type infinite sphairahedron.*

4.4.3 Construct Fractal

In Figure 42 we show a process of the tiling of sphairahedra. The sphairahedron presented in Figure 42(a) has six spherical faces. We apply inversions in each spherical face to original sphairahedron, and we obtain new six sphairahedra surrounding the initial sphairahedron as shown in Figure 42(b). Next, we apply inversions in each of the new faces to the new sphairahedra, and we obtain more sphairahedra. We continue iterating inversions, and finally, we get a three-dimensional fractal shape as presented in Figure 42(e).

Also, an infinite type sphairahedron can be tiled as presented in Figure 43. The pattern converges to fractal terrain owing to reflections over side faces of the sphairahedron as shown in Figure 44. In the fractal terrain, we can find symmetry easily. For example, we can see hexagram-like terrain patterns in Figure 44. These patterns are originated from the dihedral angles of the side faces of $\pi/3$.

In the same way as the tiling of the sphairahedron, a semi-sphairahedron can be tiled by the inversions about its faces. The resulting fractal of the semi-sphairahedron is different from normal sphairahedron's one. Figure 45 shows

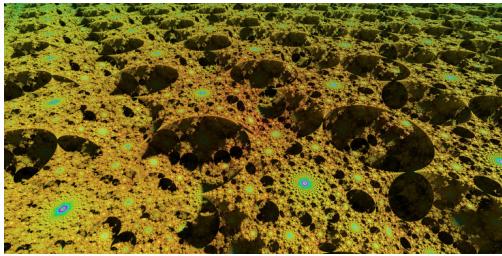


Figure 44: Fractal terrain based on the infinite sphairahe-dron in Figure 43.

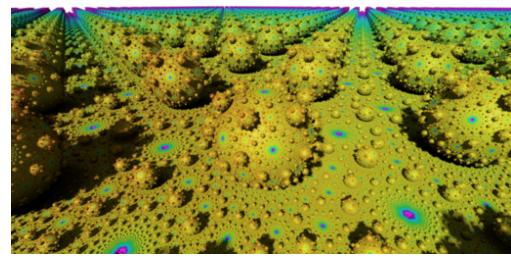


Figure 45: Fractal terrain based on the infinite semi-sphairahedron in Figure 41.

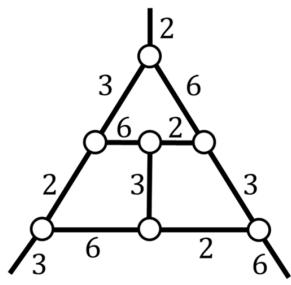


Figure 46: Polyhedral Graph for infinite sphairahe-dron.

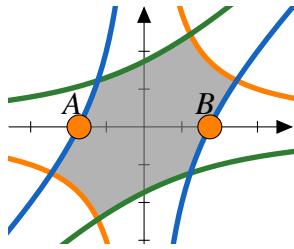
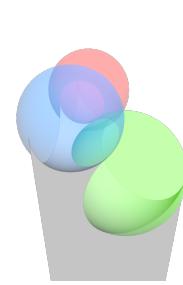
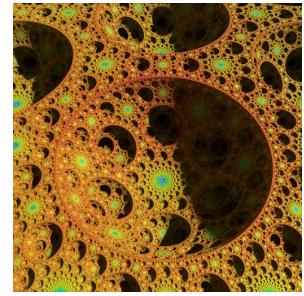


Figure 47: Parameter space of the cube-type sphairahedron.

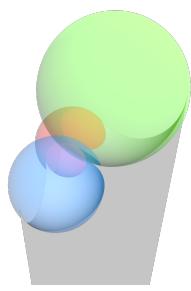


(a) Sphairahe-dron

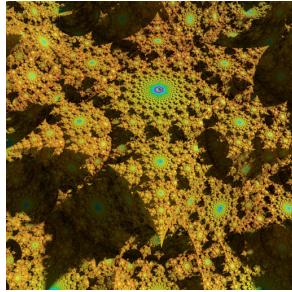


(b) Limit set

Figure 48: Sphairahe-dron corresponding to the parameter A in Figure 47.

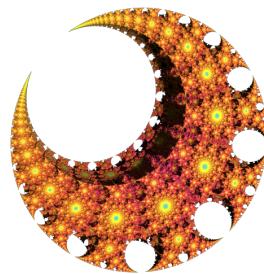


(a) Sphairahe-dron

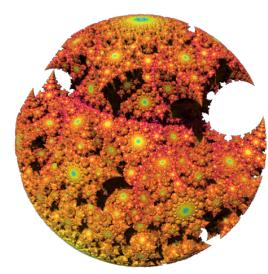


(b) Limit set

Figure 49: Sphairahe-dron corresponding to the parameter B in Figure 47.



(a) Limit set



(b) Limit set

Figure 50: Limit sets based on the sphairahe-dron in Figure 49. Each of them is transformed by two different inversion spheres.

the pattern generated by an infinite semi-sphairahedron shown in Figure 41. It is the union of an infinite number of balls circumscribing each other and no longer homeomorphic to a three-dimensional ball. Thus, It is not a quasi-sphere or a quasi-fuchsian. We will describe more about a tiling pattern of a semi-sphairahedron later.

In the images of fractals within this paper, each tile of the sphairahedra is colored according to the number of inversions. We use the color wheel to determine their color, and the tile's color varies in order of red, yellow, green, and blue. In Figure 42(a) ~ (d) and Figure 43(a) ~ (d), we refer color wheel with large steps to visualize each tile clearly. On the other hand, in the other images, we refer the wheel with smaller steps, and we find lots of tiles with many inversions in the blue parts of the fractal. We can also find that the blue parts themselves form the constant patterns. For instance, see Figure 44 and Figure 45. The blue parts are on the vertexes of sphairahedron.

Up to this point, We showed tiling patterns without gaps between the tiles and intersections of the tiles. However, not every sphairahedra can generate such proper tiling patterns. To obtain them, we have to consider two mathematical properties of the original sphairahedron, that is, the sphairahedron should be ideal and rational. In the next section, we will introduce these properties.

4.4.4 Ideality and Rationality

We consider two properties to characterize a sphairahedron: ideality and rationality. First, we introduce ideality. Let P be a sphairahedron. We say P is *ideal* when all of the edges of P are mutually tangent at their vertices. A standard polyhedron, that is, a polyhedron with planar faces, never have this property. The second property is rationality. We say P is *rational* if each of the dihedral angles of the edges is equal to π/n for a natural number n .

For instance, all of the dihedral angles of the sphairahedron in Figure 40 are $\pi/3$. That is to say, all of the dihedral angles are expressed as π/n , and the sphairahedron is rational. Then, as presented in Figure 40(a), each of the vertices is the point of contact between three balls. Thus, the sphairahedron is ideal.

4.4.5 Parameter Space of Ideal Rational Sphairahedron

Ahara and Araki worked on a classification problem of ideal rational sphairahedra and derived the parameter space of cube-type sphairahedra in their publication [12]. However, they showed only the limit set originated from a cube-type sphairahedron whose dihedral angles are $\pi/3$. In this section, we briefly show how to derive parametrization of ideal rational sphairahedra in general cases.

First of all, we have to determine the number of faces of a sphairahedron and how to arrange its edges between vertices. In order to represent a sphairahedron, we use a polyhedral graph as shown in Figure 46. The graph shows an infinite cube-type sphairahedron. The black lines represent edges, and the circles represent vertices. The three radial edges connect the infinite vertex and finite vertices. Each of the numbers beside the edges means a natural number n for the dihedral angle π/n .

Secondly, we enumerate the combinations of dihedral angles of each edge and choose one combination. In order to satisfy ideality, the sum of the dihedral angles at each vertex should be $(k - 2)\pi$ for the number of the edges k connected to the vertex. Thus, the total of the dihedral angles at each vertex of the cube-type sphairahedron is π . Ahara and Araki found that the cube-type sphairahedron has seven combinations of the angles [12].

After we choose a combination of dihedral angles, we fix some side faces and height of some balls. In the cube-type sphairahedron, we fix side faces of the prism so that their interior angles are determined angles by the graph, and we fix the height of one of the balls to zero.

Finally, we parametrize the ideal rational sphairahedron respect to the heights of the rest of the balls. The positions and radii of the balls are decided on the basis of ideality and rationality in relation to the fixed prism and balls.

For example, we show the outline of the parameter space for the infinite cube-type sphairahedra whose all dihedral angles are $\pi/3$ in Figure 47. It is derived by Ahara and Araki in their paper [12]. Figure 48 and Figure 49 show sphairahedra and their limit sets corresponding to the points A and B on the parameter space. The coordinates of the parameter space represent the heights of the two balls for the sphairahedra in Figure 48(a) and Figure 49(a). The x-coordinate is the height of the green ball on the right side, and the y-coordinate is the height of the blue ball in front of the left side. The gray area surrounded by the three hyperbolas is the parameter space of ideal rational sphairahedra. In other words, if the parameter is contained in the gray area, the corresponding sphairahedron is ideal and rational.

It is ensured mathematically that the limit set of the sphairahedron is continuously deformed while the parameter varied in the parameter space. For instance, we find that there are many crater-like dents in the limit set in Figure 48(b). As we increase the height of the green ball on the right side, the dents rise, and we can see the hexagram-like terrain in Figure 49(b).

After we obtain an infinite sphairahedron, we can get finite sphairahedra from it. We fix a point which does not match vertices of the infinite sphairahedron, and we consider another sphere centered at the point. We call the new sphere an *inversion sphere*. We invert the infinite sphairahedron in the inversion sphere, and we get a finite sphairahedron. We can choose positions and radii of the inversion sphere, and the quasi-spheres are continuously deformed according to the configuration of the inversion sphere. In Figure 50, we show the two limit sets. Each of them is generated by the same sphairahedron in Figure 49 using two different inversion spheres. In this way, we can get many variations of sphairahedra and limit sets by choosing the inversion sphere besides we change the parameter.

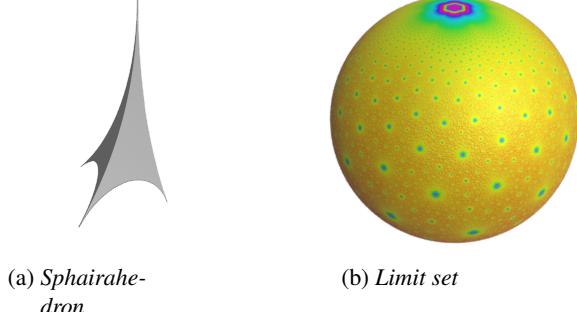


Figure 51: *Finite tetrahedron type.*

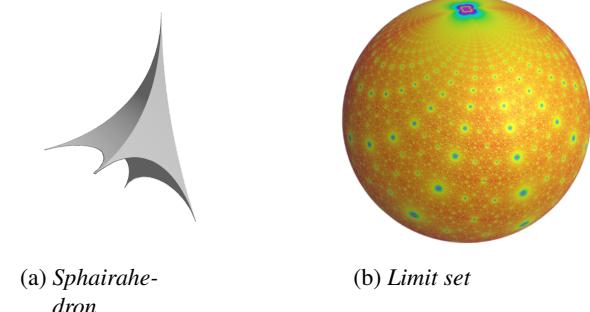


Figure 52: *Finite pentahedral pyramid type.*

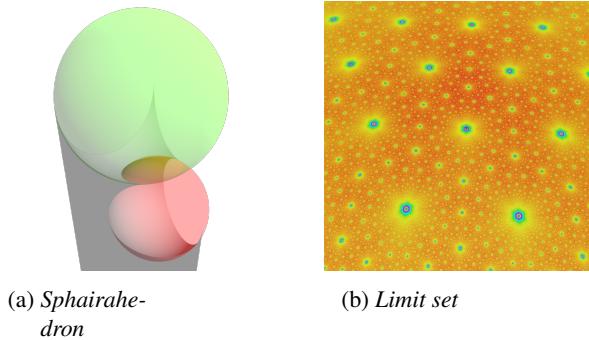


Figure 53: *Infinite pentahedral prism type.*

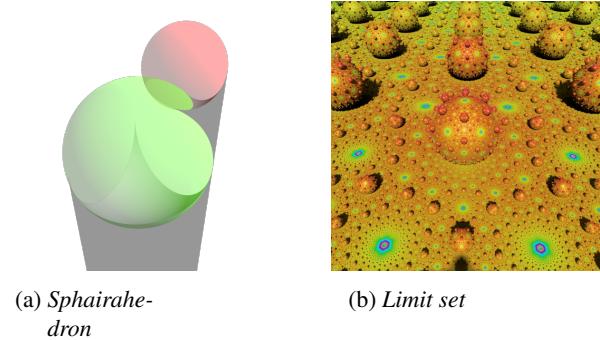


Figure 54: *Infinite pentahedral prism type with two components.*

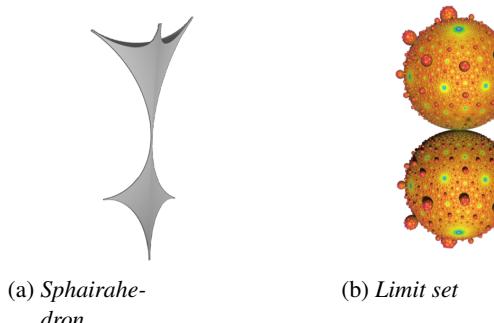


Figure 55: *Finite pentahedral prism type.*

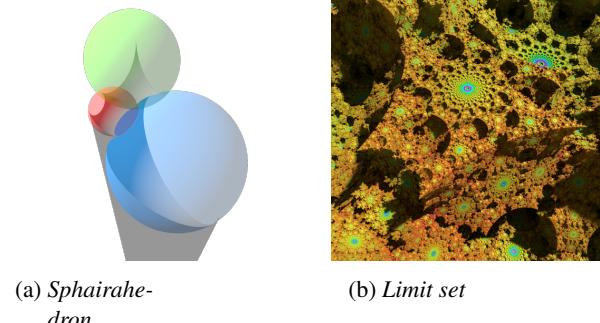
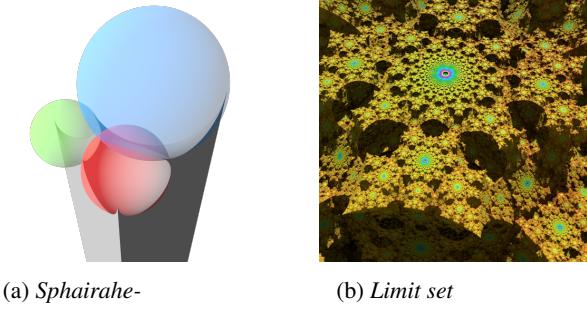


Figure 56: $(\pi/2, \pi/3, \pi/6)$ *infinite cube-type.*



(a) *Sphairahedron*

(b) *Limit set*

Figure 57: $(\pi/2, \pi/4, \pi/4)$ infinite cube-type.

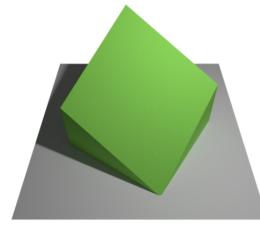
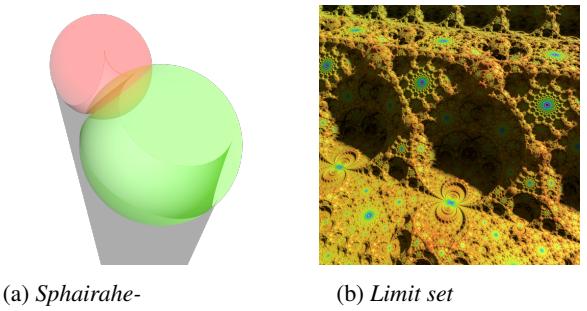


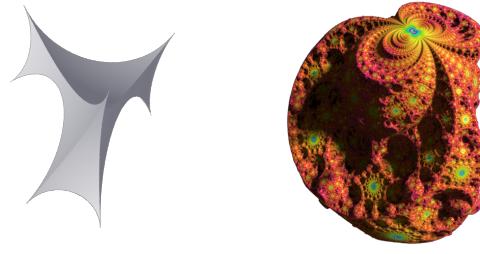
Figure 58: *Cake-type hexahedron*.



(a) *Sphairahedron*

(b) *Limit set*

Figure 59: *Infinite hexahedral cake-type*.



(a) *Sphairahedron*

(b) *Limit set*

Figure 60: *Finite hexahedral cake-type*.

4.4.6 Variations of Sphairahedra

In the same way as the parametrization of the cube-type sphairahedron, we can obtain the parameter spaces of other types of the ideal rational sphairahedron. In this section, we will show some more examples of sphairahedra and their limit sets.

Figure 51 and Figure 52 show sphairahedra and their limit sets based on a tetrahedron and a pentahedral pyramid. Each of them has the unique combination of dihedral angles and a single parameter. Both of the limit sets are actual spheres, but their patterns of color are different from each other.

Sphairahedra based on pentahedral prism have six combinations of the dihedral angles. It turns out that most combinations of this type become semi-sphairahedra. The pentahedral prism type sphairahedra have one parameter for the height of the green ball on the left side in Figure 41(a). The limit sets of them have an interesting character. See Figure 53(a). It shows one of the semi-sphairahedron whose dihedral angles are $\pi/3$. The limit set shown in Figure 53(b) seemed to be a plane, but there is an infinite number of the spherical hollows under the plane. When we decrease the height of the green ball, the semi-sphairahedron becomes to be composed of two components as shown in Figure 54(a). The hollows rise, and they emerge on the plane as balls as presented in Figure 54(b). As described in the previous section, the limit sets of semi-sphairahedra are no longer quasi-spheres or quasi-fuchsian. Figure 55 shows finite type semi-sphairahedron and its limit set. It is easy to find that the limit set is composed of an infinite number of balls circumscribing each other.

We already introduced the cube-type sphairahedron whose every angle is $\pi/3$. According to the combinations of the angles, the patterns of the limit set are greatly changed. Figure 56 and Figure 57 show the sphairahedra whose dihedral angles of the side faces are $\pi/2, \pi/3$, and $\pi/6$, and $\pi/2, \pi/4$, and $\pi/4$ respectively. Each of the limit sets forms a crater-like shape. The shapes of the limit sets result from triangular reflections of side faces and difference in the height of the spherical faces of the sphairahedra.

Finally, we show another sphairahedron based on the hexahedron called cake-type in Figure 58. The infinite sphairahedron in Figure 59(a) have four side faces, and one parameter for the height of the smaller red ball on the left side. The limit set shown in Figure 59(b) has parallel translation symmetry along the vertical directions of the faces, and the difference between the heights of two spherical faces of the sphairahedron causes the difference of elevation of

the terrain. A finite sphairahedron and its limit set are shown in Figure 60.

4.4.7 Breaking Ideality or Rationality

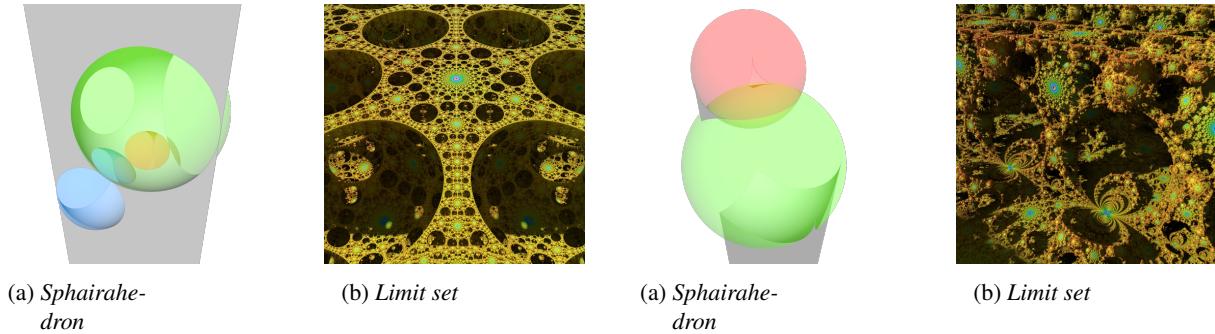


Figure 61: *Cube-type sphairahedron corresponding to the Figure 62: Cake-type sphairahedron same type as Figure outside of the parameter space in Figure 47.*

59.

In the previous section, we dealt with ideal rational sphairahedra. If we increase the number of the faces of a sphairahedron more than six, it becomes difficult to obtain an ideal rational sphairahedron. The reasons are because the constraint of dihedral angles owing to rationality becomes more strict as the number of the faces increases and the number of the edges connected to one vertex is limited to at most four owing to ideality.

In this way, we try to think about sphairahedra not always satisfying ideality and rationality. For instance, the parameter space in Figure 47 is a subset of the plane. In this section, we will examine sphairahedra and quasi-spheres corresponding to the parameter on the plane and outside of the parameter space. The sphairahedra may not be ideal or rational nor even sphairahedra, but their limit sets are meaningful shapes.

For example, we consider the cube-type sphairahedron in Figure 49. The limit set forms a hexagram-like shape, and we call six elongated parts of the hexagram arms. The arms get longer as the parameter approaches to the edge of the parameter space. When the parameter is at the edge (the point B in Figure 47,) the arms come into contact with neighboring arms. Then the parameter is outside of the parameter space, and the arms overlap each other as presented in Figure 61(b). In this case, the sphairahedron corresponding to the limit set may not be rational, and the other part of the sphairahedron gets a hole as shown in Figure 61(a).

However, if the arms overlap each other in rational angle, the sphairahedron is rational. Thus, there are ideal rational parameters outside of the parameter space, but we don't know whether they are points or lines.

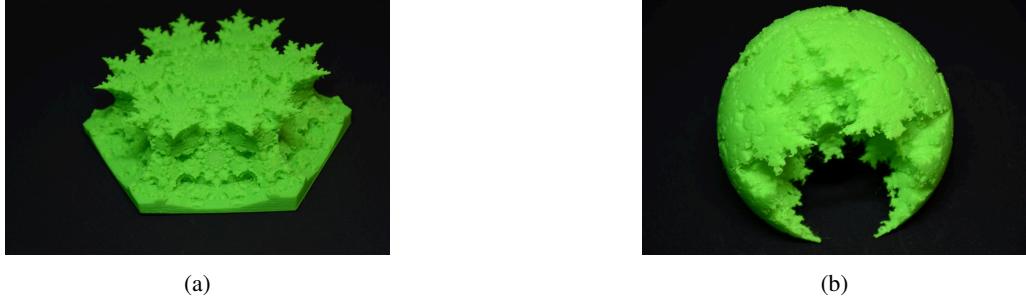
Another example is in Figure 62. It is based on the cake-type sphairahedron in Figure 59. If we make the height of the smaller red ball too high, the resulting sphairahedron will be broken as shown in Figure 62(a). However, the resulting limit set in Figure 62(b) keeps the meaningful shape, and there are holes everywhere in the shape.

4.4.8 Three-Dimensionanl Printing

Yoshiaki Araki also tried to materialize these fractals in 2006 [13]. He succeeded to materialize quasi-sphere with some methods including three-dimensional printing. However, recently, three-dimensional printing has become more popular. The printer has become cheaper than before, and we can easy to experiment for three-dimensional printing. Now, we can also make full-colored three-dimensional printing objects. We use monochrome printing by home use printer and full-colored printing by three-dimensional printing service.

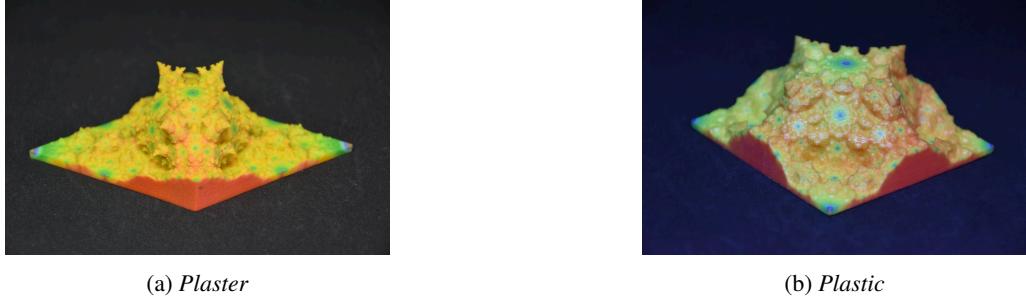
Generally, there are troublesome problems in three-dimensional printing of fractals. For example, we have to generate mesh data with IIS. IIS can generate voxel data, and we use algorithms like the *marching cubes* to make mesh data from voxel data. Moreover, we have to determine the appropriate level of details of meshes of the fractals object because fractals have self-similarity. For full-colored printing, we have to generate texture coordinates and color of the fractal as texture data. In this sub-section, we introduce how to generate fractal data for three-dimensional printing.

First of all, we have to generate polygonal fractal data. We can use IIS to generate voxel fractal data. Once we obtain voxel data, we can use a marching cubes method to convert voxel data to polygonal mesh data. In our case, we use the *OpenVDB* library to represent voxel fractal data and generate polygonal mesh data.



(a)

(b)

Figure 63: *Monochrome printing by Makerbot Replicator Z18 with PLA resin.*(a) *Plaster*(b) *Plastic*Figure 64: *Full-colored printing by DMM.make 3D print.*

Secondly, we decimate the mesh data generated in the former step because it often has too many polygons. To decimate meshes, we use three-dimensional modeling softwares called *Blender* or *ZBrush*. For monochrome printing, we complete data conversion. See Figure 63. The objects are printed by FDM three-dimensional printer. We use Makerbot Replicator Z18 with PLA resin.

Thirdly, if we want to print full-colored fractal data, we need image texture and texture coordinates of the fractal object. To obtain texture coordinates, we apply UV unwrapping to the object. After UV unwrapping, we obtain a mapping between world coordinates and UV coordinates. We create image texture using the coordinates. In this case, we used ZBrush to unwrap UV. Optionally, we can hollow the object out to reduce the printing cost.

We can place an order for full-colored three-dimensional printing service. We use *DMM.make 3D print* service. See Figure 64(a). The material of this model is plaster. Figure 64(b) shows another plastic model.

5 Conclusion

In this paper, we introduced an efficient algorithm called Iterated Inversion System (IIS) to visualize Kleinian groups based on circle inversions and sphere inversions. IIS is applied to each point on the plane in parallel. Then we can visualize Kleinian groups efficiently.

The application range of IIS is broad. We can render many images related to Kleinian groups and tiling using IIS. IIS and mathematics are developed close together. Therefore IIS is taken notice by researchers who study Kleinian groups as pure mathematics and computer graphics.

On the other hand, there is also a certain family of Kleinian groups, which we can not visualize using IIS. Our final goal is that for all Kleinian groups, we develop this kind of efficient algorithm and get mathematical results from obtained images.

Acknowledgement

I am deeply grateful to Professor Kazushi Ahara who provided wide-ranging constructive discussion and valuable comments. I would also like to express my gratitude to my family for their continuing moral support and warm

encouragements. Finally, I gratefully appreciate the financial support of Meiji University that made it possible to publish my work and complete my thesis.

References

- [1] David Mumford, Caroline Series, and David Wright. *Indra's Pearls: The Vision of Felix Klein*. 1st ed. Apr. 2002.
- [2] Albert Marden. *Hyperbolic Manifolds: An Introduction in 2 and 3 Dimensions*. 2016.
- [3] A. Marden. *Outer Circles: An Introduction to Hyperbolic 3-Manifolds*. 1st ed. May 2007.
- [4] 谷口 雅彦・奥村 善英. 双曲幾何学への招待 - 複素数で見る. Oct. 1996.
- [5] Keita Sakugawa. “On Limit Sets of 4-dimensional Kleinian Groups with 3 Generators”. In: *Tokyo J. of Math.* 33.1 (June 2010), pp. 165–182.
- [6] 佐久川恵太. “二元生成四次元クライン群における潜伏的放物型変換について”. MA thesis. 明治大学, 2007.
- [7] John C Hart. “Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces”. In: *The Visual Computer* 12.10 (1996), pp. 527–545.
- [8] Aaron Montag. “Interactive Image Sequences Converging to Fractals”. Bachelor Thesis. Technical University of Munich, 2014.
- [9] Martin von Gagern and Jürgen Richter-Gebert. “Hyperbolization of Euclidean Ornaments.” In: *Electr. J. Comb.* 16.2 (2009).
- [10] Kento Nakamura and Kazushi Ahara. “A Geometrical Representation and Visualization of Möbius Transformation Groups”. In: *Proceedings of Bridges 2017: Mathematics, Art, Music, Architecture, Education, Culture*. Ed. by Carlo H. Séquin David Swart and Kristóf Fenyvesi. 2017, pp. 159–166.
- [11] Kento Nakamura and Kazushi Ahara. “Sphairahedra and Three-Dimensional Fractals”. In: *Proceedings of Bridges 2018: Mathematics, Art, Music, Architecture, Education, Culture*. 2018, pp. 171–178.
- [12] Kazushi Ahara and Yoshiaki Araki. “Sphairahedral approach to parameterize visible three dimensional quasi-Fuchsian fractals”. In: *Computer Graphics International, 2003. Proceedings*. IEEE. 2003, pp. 226–229.
- [13] Yoshiaki Araki. “Materializing 3D quasi-Fuchsian fractals”. In: *Forma* 21.1 (2006), pp. 19–27.