

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

Kento Nakamura
Meiji University

Abstract

Kleinian group is one of the fields of mathematics. Visualized Kleinian group shows us beautiful fractal structure, and clues to understand their mathematical properties. However, it often takes much time to render Kleinian group. Thus, we invent an efficient algorithm to visualize Kleinian groups. The algorithm is called Iterated Inversion System (IIS.) It can render not only two dimensional objects but also three dimensional objects. We summarize the usage of the IIS and its applications.

Contents

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 Poincare named such groups *Kleinian Groups*. Moreover, composing Geometry, Algebra, and Analysis, Poincare built foundation of Kleinian groups theory. Klein and Poincare were rivals to study Kleinian group theory.

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 computer. Visualized images of Kleinian groups often have beautiful fractal shape. Thus, some people enjoy rendered images as arts. 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 Wright wrote a book called *Indra's Pearls* [MumfordSeriesWright200204]. The book is written for non-mathematician and contains explanation about Kleinian group theory, many beautiful visualized images, and methods of visualization. Thus, not only math enthusiast but also programmers enjoy the book.

However, the book deal with a small part of Kleinian group. Because of the properties of the group, we can not visualize all of the types of Kleinian groups in real time in spite of the growth of the performance of a personal computer.

Our 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 two-dimensional Kleinian group but also three-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 Terminology

In this paper, we use terms about Kleinian groups used in Indra's Pearls [MumfordSeriesWright200204]. The word *group* represents an algebraic group which is central concept about group theory. Algebraic group is a set which has multiplication and identity element, satysfies associative law, and whose each element of the group has inverse element.

Also, a transformation group is an algebraic group consists of transformations about plane or space. However, in this case, because we assume Möbius transformation group, we add infinity ∞ to complex plane \mathbb{C} (or three dimensional space \mathbb{R}^3) and apply one-point compactification to the set. We consider a group composed of the element of homeomorphic mapping on the set $\tilde{\mathbb{C}}$ (or $\tilde{\mathbb{R}}^3$).

We assume that transformations $f(z)$ and $g(z)$ are complex functions whose parameter is complex number z and homeomorphic mappings on $\tilde{\mathbb{C}}$. What transformation group G is generated by $f(z), g(z)$ is any element of G are represented by some composite mappings of $f(z), g(z), f^{-1}(z)$, and $g^{-1}(z)$.

In this paper, for simplicity, we use lower case alphabets such as a and b instead of $f(z)$ and $g(z)$ and upper case alphabet such as A and B instead of $f^{-1}(z)$ and $g^{-1}(z)$.

From this point, we assume transformation group G is generated by two elements a and b . Arbitrary elements of G is represented by four alphabet a, b, A , and B and we follow the rule of inverse of a is A and inverse of b is B . Thus, composite mapping $f(z) \circ g(z) \circ f^{-1}(z)$ simply is represented by abA .

¹<https://fractalforums.org/>

Following the rule of the words, we represent circular infinite words as bar, that is, $aaaa\cdots$ is represented by \bar{a} and $abABabAB\cdots$ is represented by $abAB$. These infinite words are not element of G . However, when we consider the orbit by G , we use such notations to express the limit set.

2.2 Inversions in Circles or Spheres

It is known that Möbius transformations on $\tilde{\mathbb{C}}$ are composed of even number of inversions in circles. Here, it is assumed that the inverse mapping about circle centered at $C \in \mathbb{C}$ and radius $R \in \mathbb{R}$ ($R > 0$). The formula is given by $f(z) = \frac{R^2}{z-C} + C$. According to the definition, inverse mapping is a homeomorphic mapping on $\tilde{\mathbb{C}}$.

In this context, the circles do not center infinity, but by interpreting the line on complex plane as “the circle whose center is infinitely far point and radius is infinity”, line symmetry transformation (but infinitely far point is transformed to infinitely far point) on complex plane is also inversion in the circle. Inversion mapping (including line symmetry transformation) does not preserve direction of complex plane. Thus compositions of even number of inversion mapping are homeomorphic mappings preserving direction of complex plane.

Later, we compute Jacobian of inverse mapping. This is Jacobian matrix as mapping from complex plane to complex plane. Generally, inverse mapping preserve angles. From this property, Jacobian mapping is given by multiplication of complex number. Concretely, Jacobian is composed of rotations and constant scaling and absolute value of Jacobian is as follows $Jacobian = R^2/distance(P, C)^2$ where P is a point before applying the inversion.

In the similar manner to circle inversions we can determine inversion mapping about spherical surface S^2 included in $\tilde{\mathbb{R}}^3$. Here, image of the inversion mapping is determined by center of the sphere and distance to the center. Concretely, $f(z) = \frac{R^2}{z-C} + C$. Also, a plane α included in \mathbb{R}^3 is considered as sphere whose center and radius are infinity. In this case, the inversion mapping about the plane α is plane symmetry transformation about α .

2.3 Möbius Transformations

In this study, we handle actions on $PSL_2\mathbb{C}$ of $\hat{\mathbb{C}}$. Möbius transformation is defined on $\hat{\mathbb{C}}$ and linear fractional transformation $f(z) = \frac{az+b}{cz+d}$ for complex variable z where constants a, b, c, d are complex number and satisfy $ad - bc = 1$. Such linear fractional transformation $f(z)$ is conformal homeomorphic mapping and preserves direction of $\hat{\mathbb{C}}$.

As a group acting on $\hat{\mathbb{C}}$, a set of linear fractional transformation $f(z) = \frac{az+b}{cz+d}$ and a set of 2×2 matrix $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ $PSL_2\mathbb{C}$ are same type. So, in this place, we use linear fractional transformation and complex two-dimensional projective special linear group $PSL_2\mathbb{C}$ without distinction.

Also it is known that any Möbius transformation can be represented by even number of circle inversions. The compositions of Möbius transformations is also Möbius transformations. A set of all of the Möbius transformations makes groups. For more details about Möbius transformation, refer [**MumfordSeriesWright200204**][**marden'2016**].

For a group G , mapping $G \times X \rightarrow X$ is actions of a group satisfying following conditions.

1. $e \cdot x = x$ for all x in X . (Here, e denotes the identity element of the group G .)
2. $(gh) \cdot x = g \cdot (h \cdot x)$ for all g, h in G and all x in X . (Here, gh denotes the result of applying the group operation of G to the elements g and h .)

This is also called group G acts on space X . In this sense, a set of the linear fractional transformations acts on $\hat{\mathbb{C}}$

When all of the constant numbers of Möbius transformations are real number, that is, $a, b, c, d \in \mathbb{R}$, $f(z)$ preserve upper-half plane. Moreover, Metric of upper-half plane preserves $\frac{ds^2}{(Imz)}$. Thus, a set of linear fractional transformations whose all of constants are real number acts on hyperbolic plane as isometric transformation group.

When the constants a, b, c, d are complex number, it is thought that $f(z)$ acts on hyperbolic space. As the model of hyperbolic space $mathbb{H}^3$, we consider upper half space model, but we consider three-dimensional space as the coordinates (z, t) , and z is complex coordinates and t is real coordinates. Upper-half space model is a model of hyperbolic space whose universal set is $\{(z, t) \mid t > 0\} \cup \{\infty\}$. $\hat{\mathbb{C}} = \{(z, 0) \mid z \in \mathbb{C}\} \cup \{\infty\}$ is its set of infinitely far points. Möbius transformations are groups acting on the set of infinitely far points. However, it is known that by Poincare expansion, Möbius transformations are extended to isometric mapping in hyperbolic space. For the relationship between Möbius transformations and hyperbolic space, refer [**Marden200705outerCircles**][**taniguchi'okumura199610invitation**]

From the above, researching Möbius transformation groups is heavily related to studying three-dimensional hyperbolic geometry and hyperbolic polyhedra.

Also, in this paper, we also use a concept extending a Möbius transformation to a $\mathbb{R}^3 \cup \{\infty\}$. Concretely, we define composition of even number of inversion mapping about a sphere as a three-dimensional Möbius transformation. The definition is extended relation between inversions in a circle and Möbius transformation. In that sense, Möbius transformation should be called two-dimensional Möbius transformation, but simply they are called differently Möbius transformation and three-dimensional Möbius transformation without confusion.

Three-dimensional Möbius transformations are deeply related to four-dimantional hyperbolic space. Actually, we assume four-dimensional hyperbolic upper half space model, its infinite far point set is $\mathbb{R}^3 \cup \{\infty\}$, and it is known that three-dimensional Möbius transformation gives orientation preserving isometric transformation in four-dimensional hyperbolic space via poincare expansion. In this sense studying three-dimensional Möbius transformation is deeply related to four-dimensional hyperbolic space or four-dimensional hyperbolic space and four-dimensional hyperbolic polyhedra. To represent three-dimensional Möbius transformation using matrix there are Quaternion matrix sub-group of 2×2 matrix called $Sp^k(1, 1)$. About this topic, refer [sakugawa2010limit][sakugawa2007master].

2.4 Classification of Möbius Transformations

Excluding the identical mapping, Möbius transformations $f(z) = \frac{az + b}{cz + d}$ are classified as three types. They are *Elliptic*,

Parabolic, and *Loxodromic*. By the knowledge of linear algebra, the standard form of $PSL_2\mathbb{C}$ is either one of $\begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix}$ or $\begin{pmatrix} \lambda & 0 \\ 0 & \lambda^{-1} \end{pmatrix}$. The former one of $f(z)$ is parabolic transformations. For parabolic transformation the number of fixed point is one, and 2×2 square matrix X satisfy $\text{tr}^2 X = 4$

When the standard form is $\begin{pmatrix} \lambda & 0 \\ 0 & \lambda^{-1} \end{pmatrix}$ or absolute value of lambda is one, that is $|\lambda|$ is 1, $f(z)$ is called elliptic transformations. The properties of elliptic transformation are they have two fixed points and the transformation gives rotations around fixed points. Also, the matrix representation $\text{tr}^2 X > 4$ is necessary and sufficient conditions. In addition, the möbius transformation having finite order is elliptic transformation.

When the standard form is $\begin{pmatrix} \lambda & 0 \\ 0 & \lambda^{-1} \end{pmatrix}$ and complex number $|\lambda| \neq 1$, $f(z)$ is called loxodromic. If λ is real number exclude in 1, the transformation is called *hyperbolic* too. The properties of loxodromic transformation are they have two fixed points. One of the fixed points is attracting and the other one is repelling. Also, $\text{tr}^2 X < 4$ is necessary and sufficient conditions.

For three-dimensional transformations also have this kind of classification. In this case, it is added *simple* or *complex* and there are six variations of classification like “simple elliptic” or “complex loxodromic”. For detailed definitions of them, see chapter 4.3.2.

2.5 Kleinian Groups

Kleinian groups are groups originated from name of Felix Klein who is a mathematician in nineteenth century. The group G is Kleinian group when it satisfies following two conditions. Firstly, G should be a sub-group of Möbius transformation group. Secondly, about a fixed point $x \in \mathbb{H}^3$ in hyperbolic space, its orbit space $Gx = \{gx \mid g \in G\} \subset \mathbb{H}^3$ is properly discontinuous. In this place, sub-set A of hyperbolic space \mathbb{H}^3 is properly discontinuous means for any compact set K of \mathbb{H}^3 , there are only finite number of g which is element of A and becomes $g(K) \cap K \neq \emptyset$.

When a point x 's orbit space is properly discontinuous, we take fundamental domain whose volume is plus. Considering Möbius transformation gives isometric transformation of hyperbolic space, we also consider tiling of hyperbolic space by Kleinian groups.

In this paper, we consider Möbius transformation widely and think about a mapping which is represented by a composite mapping of inversion mapping (not limited to even number). As a term, to distinguish original Möbius transformation we call them *extended Möbius transformation*. A composite mapping composed of odd number of inversion mapping is represented by $f(z) = \frac{a\bar{z} + b}{c\bar{z} + d}$ where a, b, c, d are complex number and $ad - bc = 1$. However, \bar{z} represents conjugation of complex number.

Extended Möbius transformation introduced isometric mapping (not necessarily preserve direction) on three-dimensional hyperbolic space by poincare expansion. Properly discontinuous sub-group of extended Möbius transformation group is called *extended Kleinian groups*. Basic property of Kleinian group is described in chapter 2 of [marden'2016].

2.6 Limit Set

Let Kleinian group be Γ . We call closed set of all of the limit point *limit set*. Also, We represent it as $\Lambda(\Gamma)$. There are some known properties about limit set. For example,

- The orbit space of fixed point of generators becomes limit set.
- A point on the limit set transformed by element of the group also a point on the limit set.
- Fixed points of generators become limit set.

infinite words \overline{ab} converge to it is called *algebraic limit*. *geometric limit* is a point moved to The property of the limit set is also described in chapter 2.4.1 of [marden'2016].

3 Visualization of Kleinian Groups

3.1 Basic Methods for Visualization

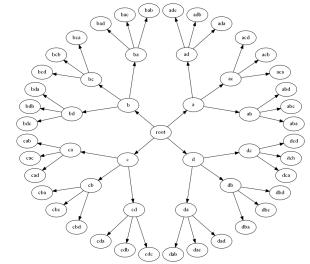


Figure 1: Cayley Graph

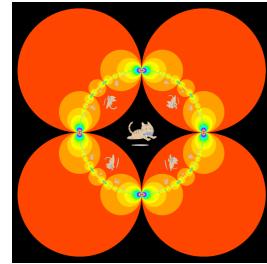


Figure 2: Orbit of the image

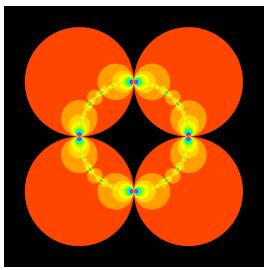


Figure 3: The orbit of the disks

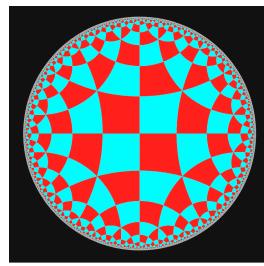


Figure 4: Hyperbolic Tessellation

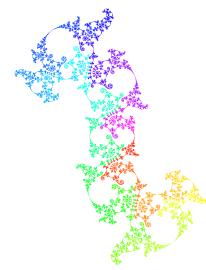


Figure 5: Limit set of the Kleinian group

In this sub-section, we introduce basic methods for visualizing extended Kleinian groups. For simple example, we explain an extended Kleinian group generated by four involution mappings.

First of all, to visualize a group we consider *Cayley graph*. Cayley graph is for a group G given a generators all elements of G are nodees and multiplication of generators by right to be related two elements connected by edges. For easy example, we assume a group generated by four involution (A mapping becomes identity mapping by square.) In short, it is a 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 is shown in Figure??.

The element of the group is nodes. Four edges corresponds to a, b, c, d are emitted from each node. Because a, b, c, d are involution, $a = a^{-1}$ and an edge corresponds to inverse mapping is not needed. In Figure ??, we put identity element id to center and four edges are stretching symmetrically. In Cayley graph, length of an edge from identity give the shortest length of a word.

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

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

In order to draw orbit space of the cat, there are breadth first search algorithm and depth first search algorithm. In the following, we explain breadth first algorithm.

Firstly, in this way, we draw inverted image of original cat by shorter words. First of all, we draw the center cat. Next, we draw transformed cat by the word whose length is one, and we draw transformed cat by the word whose length is two. We continue iterating these processes. This is breadth first search algorithm. Visualization of this algorithm are easy to implement and understand. roughly understand its actions of the groups. However, there are defective what the computational complexity is easy to increase.

Also, there is one more method of traversing graph. It is called depth first search. This one can draw only limit set directly. See Figure ???. This shows an only limit set of a Kleinian group. However, we do not use this algorithm in this paper. For more details of the algorithm, read [**MumfordSeriesWright200204**].

3.2 Iterated Inversion System (IIS)

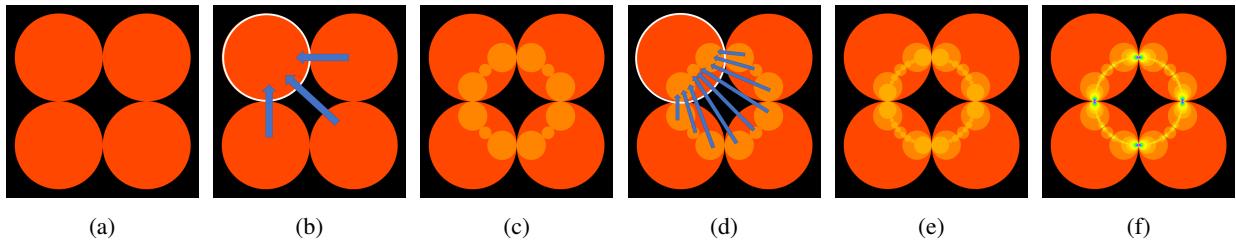


Figure 6: The process of rendering the orbit of Schottky disks

To solve the problems we discussed in the previous sub-section, we focus circle inversions and invent an efficient algorithm to visualize circle inversion fractals shown in Figure ??(??). 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 ??(??) shows the orbit of the first four circles in Figure ??(??). It is also a Kleinian group composed of four circle inversions. Especially, it is called a *quasi-fuchsian group*. 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 ??(??). We call orange disks *initial disks* and their boundary *initial circles*.
2. First of all, we focus on the white circle in Figure ??(??). 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 ??(??).
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 ??(??). Each inversion in the Schottky circle generates smaller disks, and we obtain Figure ??(??).

5. We continue iterating these process, that is, we continue applying each inversion in the initial circle to resulting smaller disks. Finally, we get Figure ??(??).

3.2.1 Two Dimensional IIS

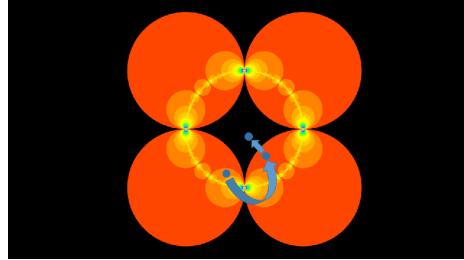


Figure 7: *Orbit of blue point by IIS*

Algorithm 1 Iterated Inversion System (IIS)

```

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

```

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

IIS is 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 ?? 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 ???. Later, we will introduce generators other than simple inversions. Thus, we consider a map G such that G is an identity for a point in the *fundamental domain* and that G is a composition of inversions for other points. The fundamental domain is the terminal area of transformations. In the above case, the black area where outside of all circles.

3.2.2 Three Dimensional Extension

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



Figure 8: The orbit of the sphere inversion fractal

We use *ray tracing* to visualize three-dimensional objects. Ray tracing computes an intersection between a ray and objects algebraically. There are two ways to render nesting spheres. First one is to draw nesting spheres as transparent spheres. Second one is *volume rendering*. However, these are difficult to render them efficiently, and visualized images are not interesting.

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

We use *Sphere Tracing* [hart1996sphere] 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 following paragraphs, we introduce ray tracing and sphere 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 regular ray tracing, we calculate the intersections algebraically, but we can not compute intersection to fractal objects. On the other hand, in sphere tracing, we march the “tip” of the ray 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 given point and objects. For example, a distance function of a sphere is as follows. Let P be a tip of a ray, let C and R be center and radius of the sphere. Distance function $f(P)$ is $f(P) = \text{distance}(P, C) - R$. If there are many spheres, we use minimum distance to the sphere.

However, in regard to fractal rendering, it is difficult to get an actual distance to its shape. So, we use a lower estimated distance as a return value of the distance function. The technique to approximate distance is called *distance estimation*. For more details about fractal rendering and distance estimation, see also the blog post² by Christensen.

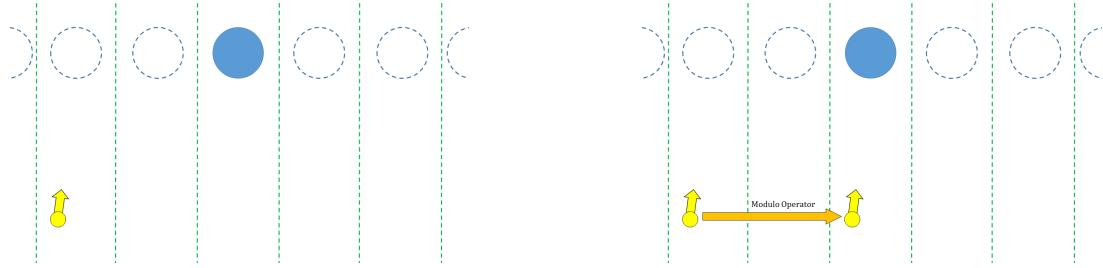


Figure 9: modulo1

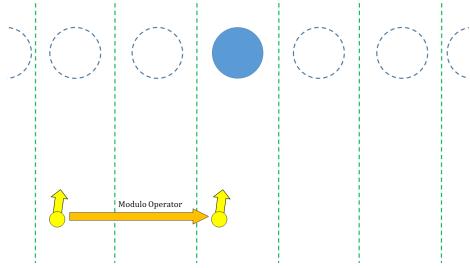


Figure 10: Fold space by modulo operator

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

²Mikael H Christensen, Distance Estimated 3D Fractals (Part I):

<http://blog.hvidtfeldts.net/index.php/2011/06/distance-estimated-3d-fractals-part-i/>

See Figure ???. 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 a minimum distance between the ray and dotted circles. However, we only know the position of the tip of the ray and the blue disk. 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 ???. Finally, we can draw line upped disks.

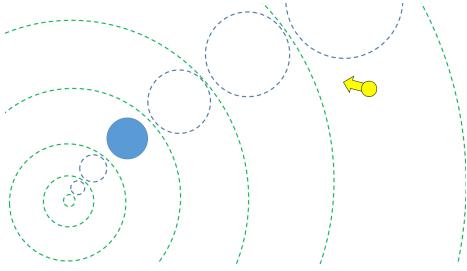


Figure 11: *scaling1*

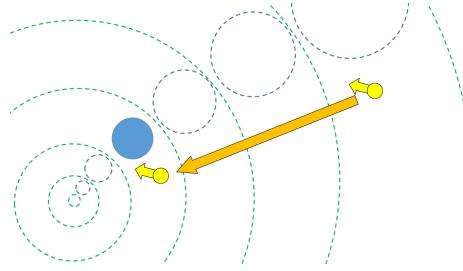


Figure 12: *scaling2*

Next, we consider scaling example. See Figure ???. 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 minimum distance to dotted blue circles. We assume that the nearest sphere to the tip of the ray is in same regions. However, we only know the coordinates of the original blue disk and tip of the ray. 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 scale dividing computed distance by Jacobian (sometimes referred to as the Jacobian determinant) of scaling. See Figure ???. If the scaling of the circle is $f(x) = 2x$, we divide the scale by eight because the ray is moved three green scaling regions.

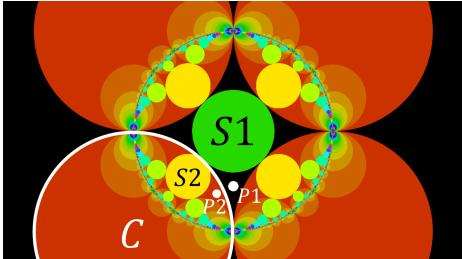


Figure 13: *XY-slice image of Figure ??(??)*

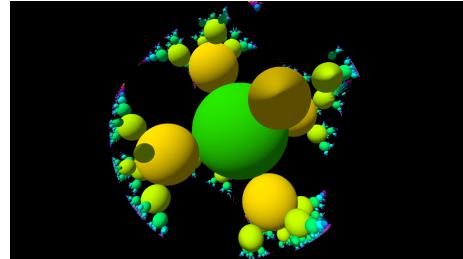


Figure 14: *artifact*

In a similar manner to scaling, we can render sphere inversion fractals. For the sake of simplicity, we consider a slice image of Figure ???. See Figure ???. This image shows the XY-slice of the orbit of spheres. Orange disks in the background are slices of the orbit of initial inversion spheres. Slices of the orbit of the base sphere are colored in the same color as the orbit shown in Figure ??(??). C is the white circle, the boundary of the initial inversion sphere. S_1 is the base sphere and the inversion of S_2 in the circle C . The white point P_1 is the inversion of P_2 in the circle C .

Now we assume that the tip of the ray is at P_2 . Let's calculate the minimum distance between P_2 and the orbit of base spheres. The nearest sphere to P_2 is S_2 . So, we have to calculate the distance between P_2 and S_2 . We call the distance d . However, we do not know the center and radius of S_2 . So, we calculate d from a distance between S_1 and P_1 . 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 $d \geq \text{distance}(P_1, S_1)/\text{Jacobian}$. 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

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

through the real objects. This causes artifact shown in Figure ???. The fore part of the fractals is not rendered. In order to avoid this kind of problems, we shrink the length of the estimated distance. It 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.

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  $\leftarrow$  true
    for each Map  $G$  in Maps do
        if  $G$  is available to coordinates then
             $dr \leftarrow dr * (\text{Jacobian of } G(\text{coordinates}))$ 
            coordinates  $\leftarrow G(\text{coordinates})$ 
            INCREMENT count
            inFundamentalDomain  $\leftarrow$  false
        end if
    end for
    if inFundamentalDomain then
        BREAK for
    end if
    end for
    for each BaseSphere  $S$  in BaseSpheres do
         $d \leftarrow \min(d, \text{scalingFactor} * (\text{distance(coordinates, } S.\text{center}) - S.\text{radius}) / (\text{absolute value of } dr))$ 
    end for
    return  $d$ 

```

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

3.3 Related Works

Aaron Montag uses texture based approach to visualize a limit set of the Kleinian groups [Montag2014hyperbolicIFS]. 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.

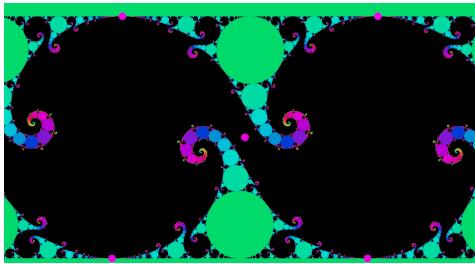


Figure 15:

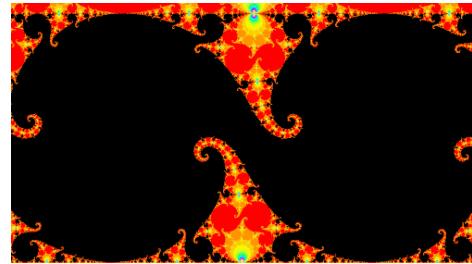


Figure 16:

Jos Leys invented efficient algorithm to draw a Kleinian group with Maskit parametrization shown in Figure ??⁴. IIS uses circle or sphere inversions, on the other hand, Jos Leys used Möbius transformations algebraically. He observes the orbit of the maskit parametrization group and We can apply circle inversions to his figure, and we obtain Figure ??.

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

Martin von Gagern and Jürgen Richter-Gebert introduce a similar algorithm called *Reverse Pixel Lookup* [journals/combinatorics/0] to render two-dimensional hyperbolic tiling. 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 17: 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 ??(??). The image generated by four inversions of circles. After applying IIS, we only fill the pixel when the transformed point is moved into inner part of the black area, and we obtain inner part of the circle inversion fractals. See Figure ??(??).

4.2 Render Circles



Figure 18: Edge of the circle inversion fractal

We can draw only edges of disks in circle inversion fractals as shown in Figure ???. 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 outside of the initial disks, we divide computed distance by accumulated Jacobian.

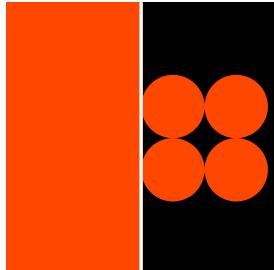
4.3 Geometrical Representation of Möbius Transformation Groups

In this paper, we mainly use circle or sphere inversions. 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

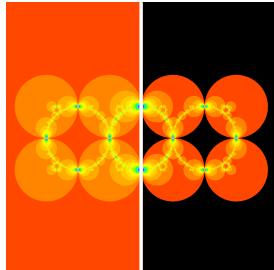
any Möbius transformation out of inversions. We can compose them by an even number of inversion. Thus, we can apply IIS to visualize fractals combining circle inversion fractals and Möbius transformation groups. Moreover, we can tweak the parameters of Möbius transformations by arranging geometrical 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 [bridges2017:159].

4.3.1 Two Dimensional Generators

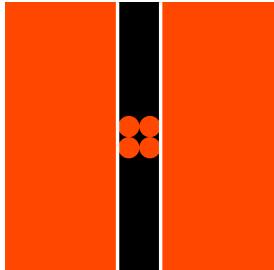


(a) Generator

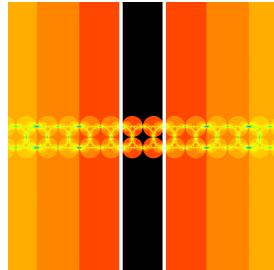


(b) Orbit

Figure 19: Inversion in the circle with infinite radius and four Schottky disks

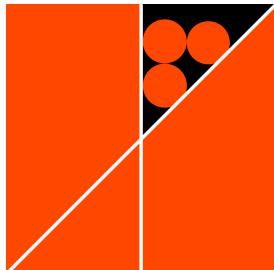


(a) Generator

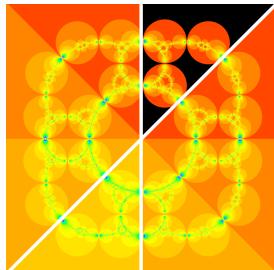


(b) Orbit

Figure 20: Parallel translation generator and four Schottky disks

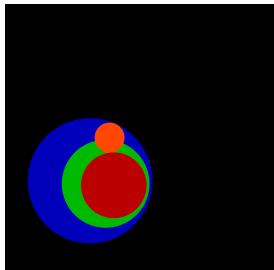


(a) Generator

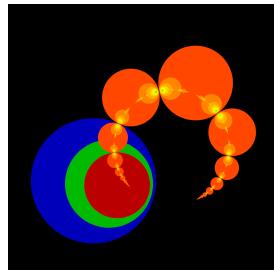


(b) Orbit

Figure 21: Rotation generator and three Schottky disks

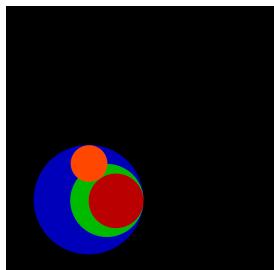


(a) Generator

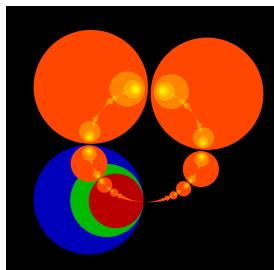


(b) Orbit

Figure 22: Hyperbolic generator and a Schottky disk

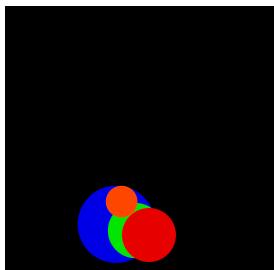


(a) Generator

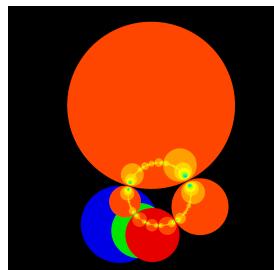


(b) Orbit

Figure 23: The orbit of Parabolic generator and a Schottky disk



(a) Generator



(b) Orbit

Figure 24: Elliptic 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 border line of half plane. See Figure ??(??). The four inversion circles are lying on the right side, and there is the



Figure 25: *Loxodromic generator and a Schottky disk*

orange region on the left side. The region is a half plane, that is, a disk with infinite radius. Its boundary is colored with white line. The orbit is shown in Figure ??(??). 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 ??(??). There are two orange half planes on the right and left sides. The orbit is shown in Figure ??(??).

Rotation. A composition of reflections over two crossing half planes generates a rotation. The generator is shown in Figure ??(??). The two half planes are crossing. The rotation axis is crossing point of white border lines. The orbit is shown in Figure ??(??). It has a rotation symmetry and it is an elliptic transformation.

Composition of Two Circles. Next, we use a composition of inversions in two circles. See Figure ??(??). 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 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 ??(??). 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 .

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

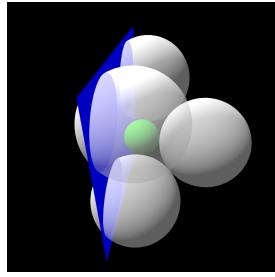
In the process of IIS, we can transform the point to the fundamental domain by applying G repeatedly. The fundamental domain of this type of generators is the blue and green area in Figure ??(??)..

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

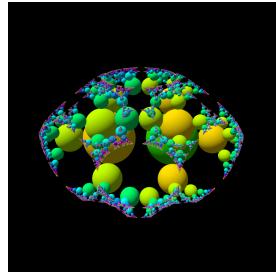
When C_1 and C_2 are crossing as in Figure ??, the generator becomes an elliptic transformation. The orbit is as shown in Figure ??(??). The disks are rotated around crossing line. Note that crossing angle of C_1 and C_2 should be rational angles; otherwise, the orbit of the disks will overlap each other.

Loxodromic. We can twist the orbit by adding another two inversions. See Figure ??(??). 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 ??(??). This is a loxodromic transformation⁷. The map G is as follows.

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

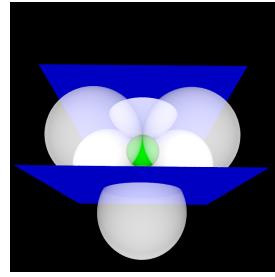


(a) Generator

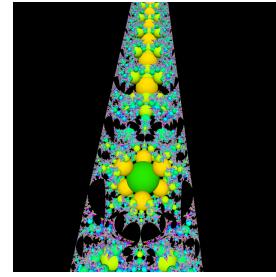


(b) Orbit

Figure 26: *Inversion in the sphere with infinite radius, four Schottky spheres, and a base sphere*

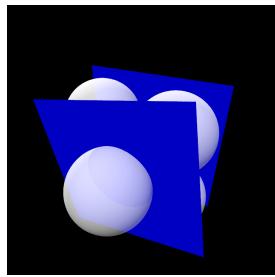


(a) Generator

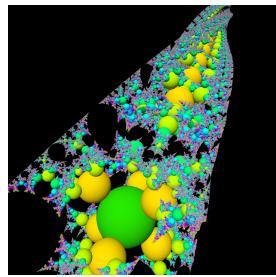


(b) Orbit

Figure 27: *Parallel translation generator, six Schottky spheres and a base sphere*

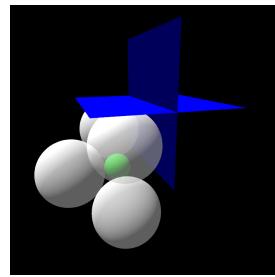


(a) Generator

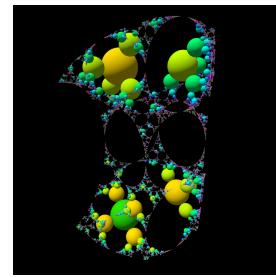


(b) Orbit

Figure 28: *Compound parabolic generator, six Schottky spheres and a base sphere*

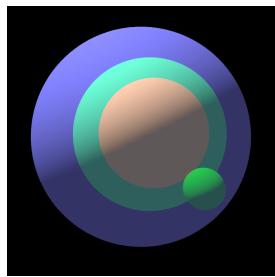


(a) Generator

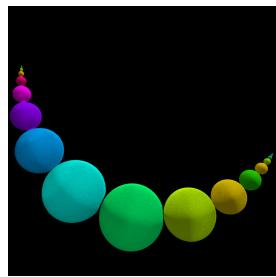


(b) Orbit

Figure 29: *Rotation generator, four Schottky spheres, and a base sphere*

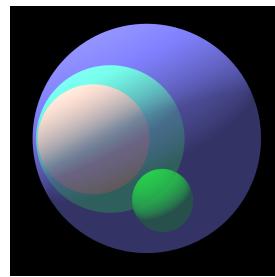


(a) Generator

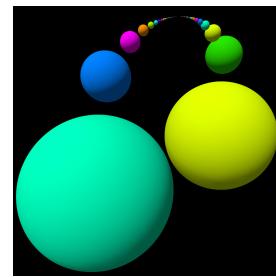


(b) Orbit

Figure 30: *Hyperbolic generator and a base sphere*

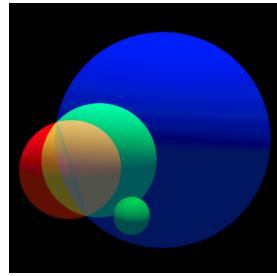


(a) Generator

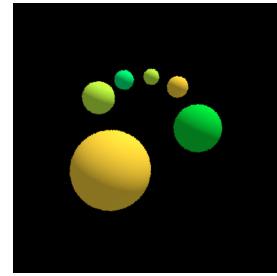


(b) Orbit

Figure 31: *Parabolic generator and a base sphere*

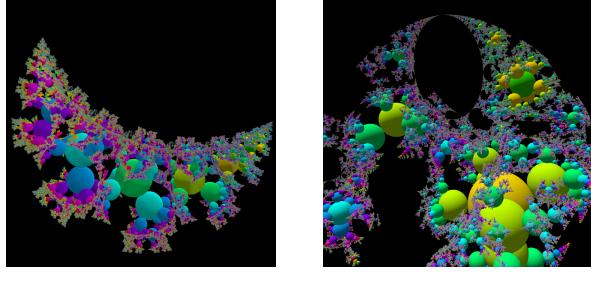


(a) Generator

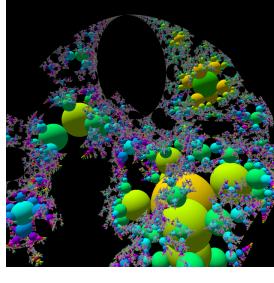


(b) Orbit

Figure 32: *Elliptic generator and a base sphere*

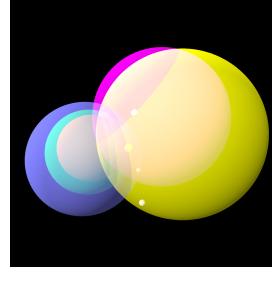


(a) *Hyperbolic*

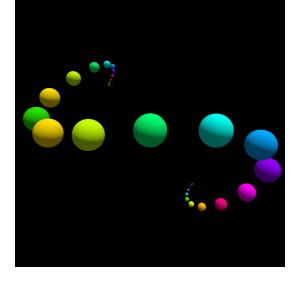


(b) *Parabolic*

Figure 33: *The orbit generated by a composition of two spheres and six Schottky spheres*



(a) *Generator*



(b) *Orbit*

Figure 34: *Compound loxodromic generator and a base sphere*

4.3.2 Three Dimensional Generators

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 ??(??). 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 ??(??). The resulting orbit has a 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 ???. This is a parabolic transformation. Moreover, in 3D, we can add a twist to the orbit. See Figure ???. The orbit is rotated around the normal vector of the planes for every translation. This operation is possible in 3D space, because we have gained a degree of freedom over 2D space. The transformations yielding twisted orbits are called *compound parabolic* transformations.

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 ??.

Composition of Two Spheres⁹. We compose generators using two spheres. See Figure ??(??). 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 ??(??). When S_1 and S_2 come in contact with each other at one point as shown in Figure ??(??), it becomes a parabolic transformation. The orbit of spheres touches at the fixed point. It is shown in Figure ??(??).

When S_1 and S_2 are crossing as in Figure ??, the generator becomes an elliptic transformation. The orbit is as shown in Figure ??(??). 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 ?? shows the example of the more complicated orbit of spheres generated by adding six inversion spheres to the group shown in Figure ?? and Figure ??.

Compound Loxodromic¹⁰. Finally, we add two spheres perpendicular to S_1 and S_2 . See Figure ??(??). 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.

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

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

⁵<https://www.shadertoy.com/view/MsScWW>

⁶<https://www.shadertoy.com/view/XsBcDD>

⁷<https://www.shadertoy.com/view/lsSyDW>

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

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

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

The composition of inversions in $S3$ and $S4$ represents rotation. The twisted orbit shown in Figure ??(??) is analogous to the loxodromic transformations in 2D. Therefore, we call this generator *compound loxodromic*.

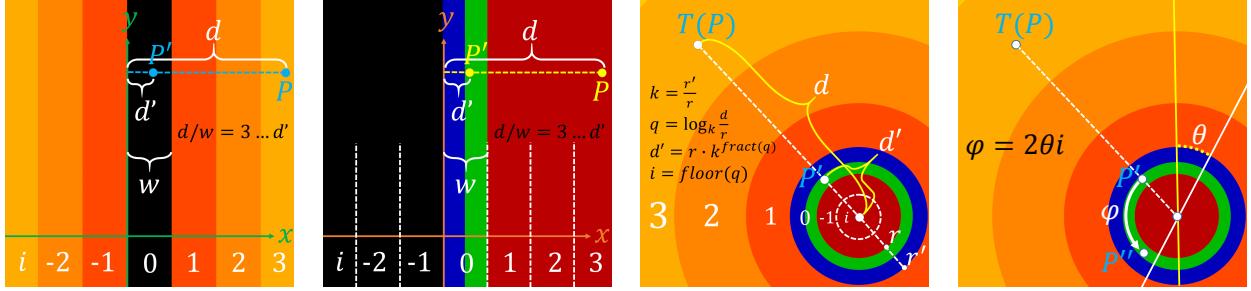


Figure 35: Parallel translation

Figure 36: Inverted parabolic generator

Figure 37: Inverted hyperbolic generator

Figure 38: Inverted loxodromic generator

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 Translation.¹¹ Suppose that there are two parallel lines similar to Figure ???. 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 ???. Let w be a distance between two lines, let i be an index of a series of bands as in Figure ???, 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.

Parabolic.¹² Next, we consider parabolic transformations like Figure ???. 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 ???. 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.

Loxodromic.¹³ Finally, we consider hyperbolic transformations like Figure ???. 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 ???. 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: $k = \frac{r'}{r}$, $q = \log_k \frac{d}{r}$, $d' = r \cdot k^{fracalPart(q)}$, $i = floor(q)$. We calculate P' using d' . When there is a loxodromic transformation similar to Figure ???, we also apply T to L and $C3$, and we call them TL and $TC3$. See Figure ???. 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.

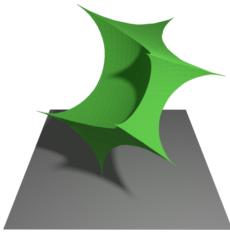
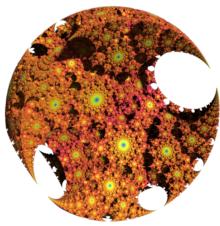
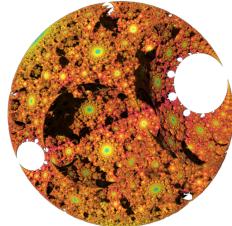


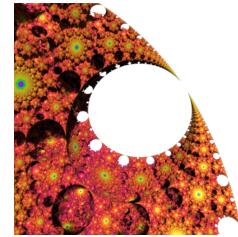
Figure 39: Cube-type sphairahedron.



(a)



(b)



(c)

Figure 40: Images of a quasi-sphere rendered in different viewpoints.

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 [**bridges2018NakamuraAhara**]. The paper is revised and included for this subsection.

The author gathers images, animation, and renderer for the fractals at <https://sphairahedron.net>.

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 [**ahara2003sphairahedral**].

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 ???. 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 ???. The union of all the tiles is mostly homeomorphic to a three dimensional ball. Thus, this is called a *quasi-sphere*. Also, boundary of the fractal is 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

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 $\overline{D_1}, \overline{D_2}, \dots, \overline{D_p}$ 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.

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

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

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

¹⁴Quasi-fuchsian fractals: <https://www.youtube.com/watch?v=3lc09zRCv-4>

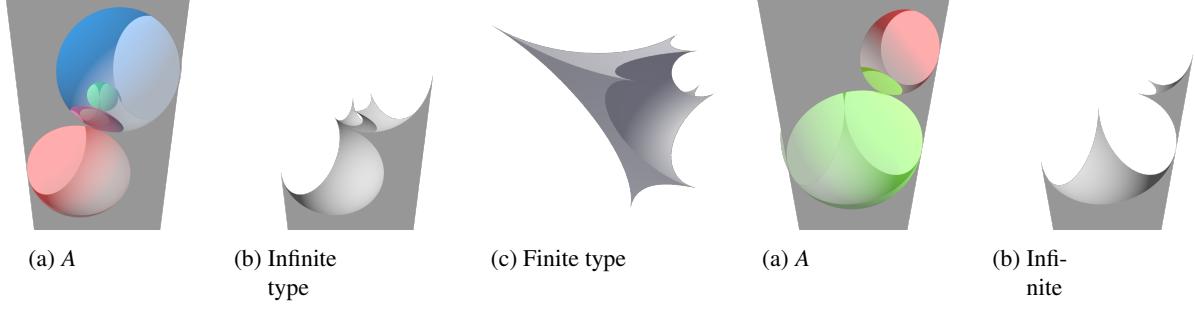


Figure 41: Sphairahedron.

(a) A

(b) Infinitely
type

Figure 42: Semi-sphairahedron.

(c) Finite type

(a) A

(b) Infinitely
type

The image in Figure ??(??) 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 ??(??). 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, it 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 ??(??) 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 ??(??) 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 ??(??). We can regard the resulting shape as a singular case of a pentahedron, and we call it a *semi-sphairahedron*.

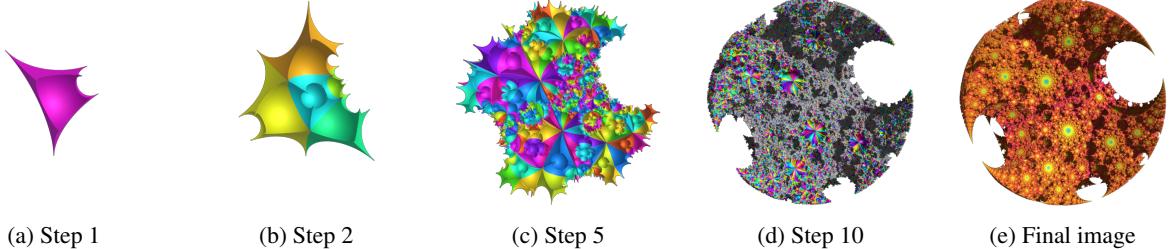


Figure 43: Tiling of a finite cube-type sphairahedron.

4.4.3 Construct Fractal

In Figure ?? we show a process of the tiling of sphairahedra. The sphairahedron presented in Figure ??(??) 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 ??(??). 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 ??(??).

Also, an infinite type sphairahedron can be tiled as presented in Figure ???. The pattern converges to fractal terrain owing to reflections over side faces of the sphairahedron as shown in Figure ???. In the fractal terrain, we can find symmetry easily. For example, we can see hexagram-like terrain patterns in Figure ???. 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 ?? shows the pattern generated by an infinite semi-sphairahedron shown in Figure ???. 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.

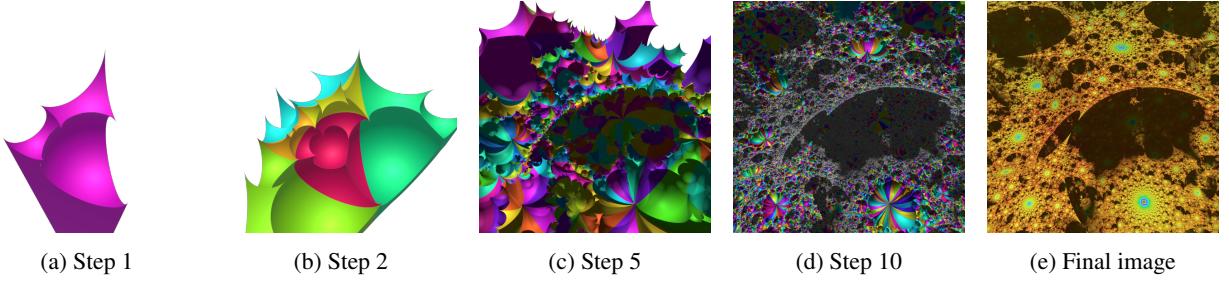


Figure 44: Tiling of a cube-type infinite sphairahedron.

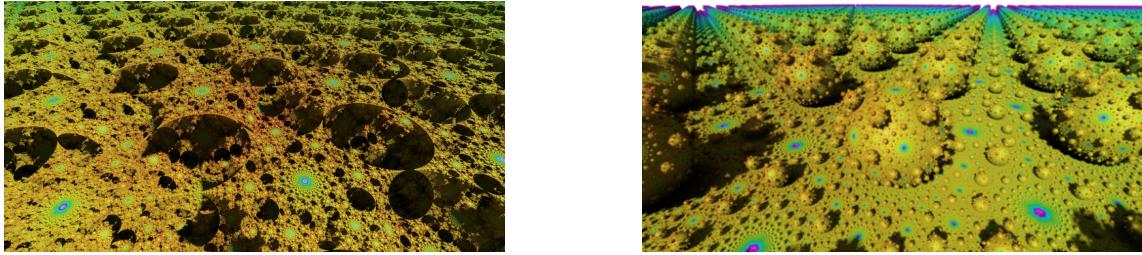


Figure 45: Fractal terrain based on the infinite sphairahedron in Figure ??.

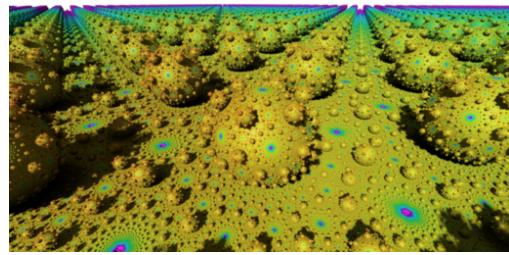


Figure 46: Fractal terrain based on the infinite semi-sphairahedron in Figure ??.

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 ?? (??) ~ (??) and Figure ?? (??) ~ (??), 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 n. 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 ?? 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 ??(??), 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 [ahara2003sphairahedral]. 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 ???. The graph shows an infinite cube-type sphairahedron. The black lines represent edges, and the circles represent vertices. The three radial

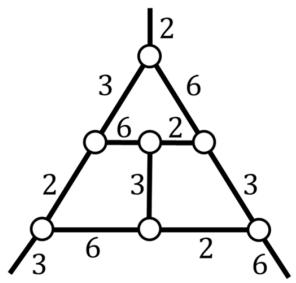


Figure 47: Polyhedral Graph for infinite sphairahedron.

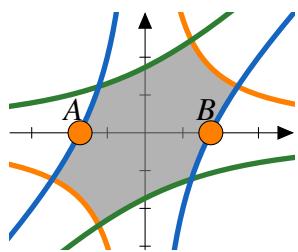
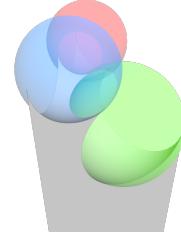
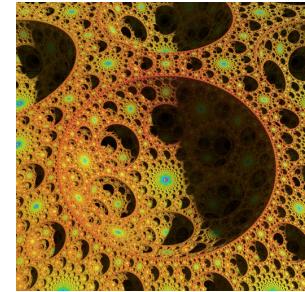


Figure 48: Parameter space of the cube-type sphairahedron.

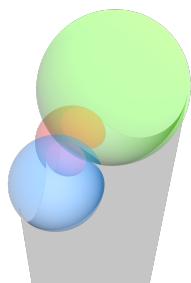


(a) Sphairahedron

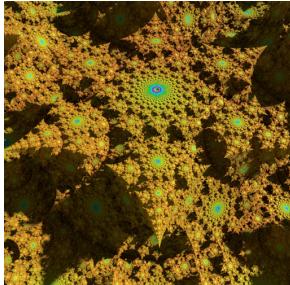


(b) Limit set

Figure 49: Sphairahedron corresponding to the parameter A in Figure ??.

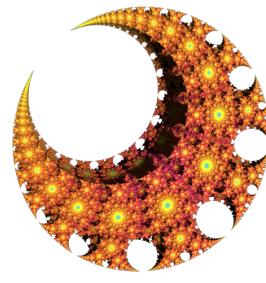


(a) Sphairahedron

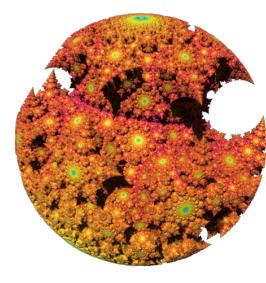


(b) Limit set

Figure 50: Sphairahedron corresponding to the parameter B in Figure ??.



(a) Limit set



(b) Limit set

Figure 51: Limit sets based on the sphairahedron in Figure ???. Each of them is transformed by two different inversion spheres.

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 [ahara2003sphairahedral].

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 ???. It is derived by Ahara and Araki in their paper [ahara2003sphairahederal]. Figure ?? and Figure ?? 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 ??(??) and Figure ??(??). 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 ??(??). 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 ??(??).

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 ??, we show the two limit sets. Each of them is generated by the same sphairahedron in Figure ?? 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.

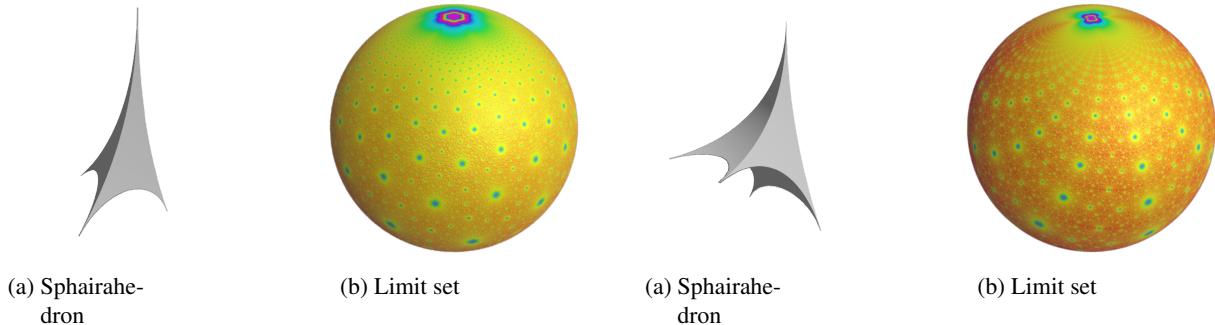


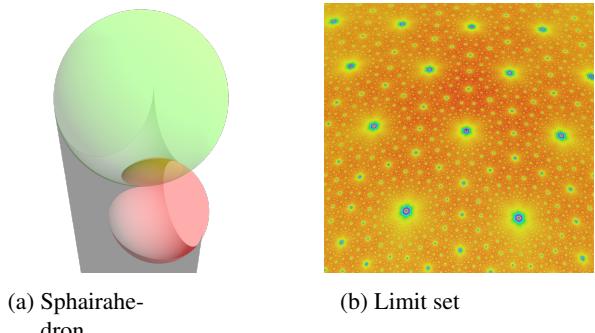
Figure 52: Finite tetrahedron type.

Figure 53: Finite pentahedral pyramid 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.

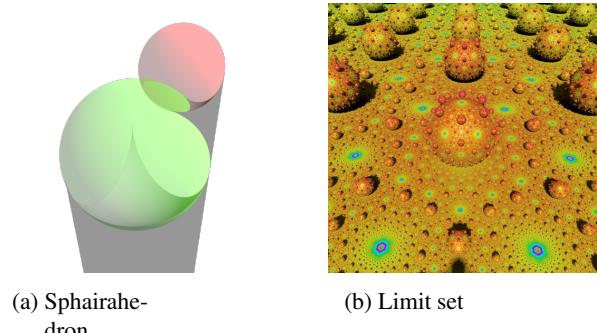
Figure ?? and Figure ?? show sphairahedra and their limit sets based on a tetrahedron and a pentahedral pyramid. Each of them has the unique combination of dihedral angles and a single parameter. Both of the limit sets are actual spheres, but their patterns of color are different from each other.



(a) Sphairahe-dron

(b) Limit set

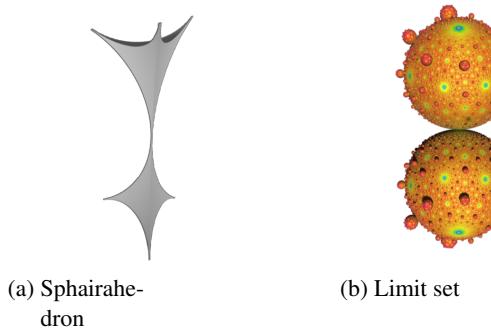
Figure 54: Infinite pentahedral prism type.



(a) Sphairahe-dron

(b) Limit set

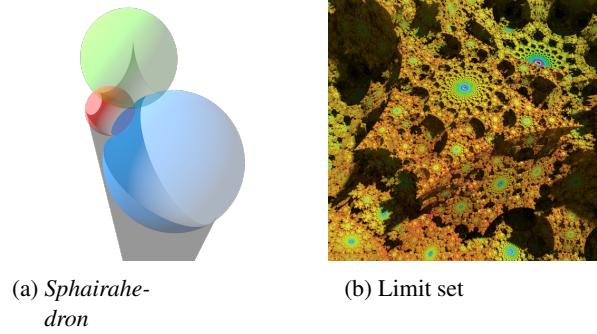
Figure 55: Infinite pentahedral prism type with two components.



(a) Sphairahe-dron

(b) Limit set

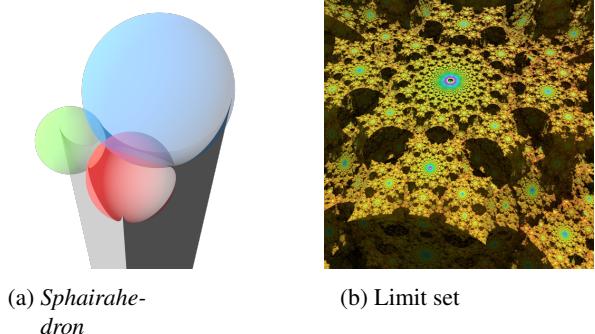
Figure 56: Finite pentahedral prism type.



(a) Sphairahe-dron

(b) Limit set

Figure 57: $(\pi/2, \pi/3, \pi/6)$ infinite cube-type.



(a) Sphairahe-dron

(b) Limit set

Figure 58: $(\pi/2, \pi/4, \pi/4)$ infinite cube-type.

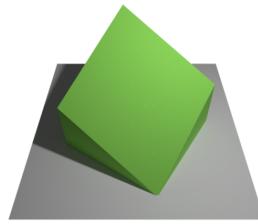
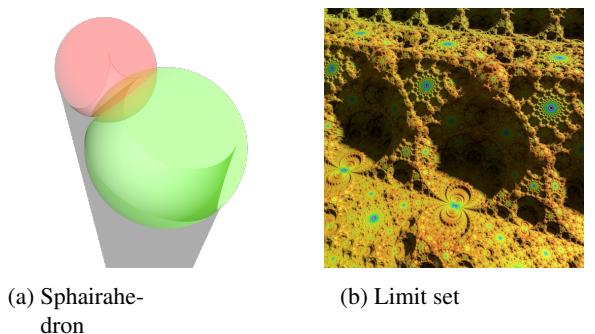


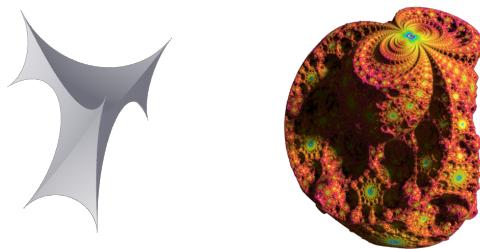
Figure 59: Cake-type hexahedron.



(a) Sphairahe-dron

(b) Limit set

Figure 60: Infinite hexahedral cake-type.



(a) Sphairahe-dron

(b) Limit set

Figure 61: Finite hexahedral cake-type.

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 ??(??). The limit sets of them have an interesting character. See Figure ??(??). It shows one of the semi-sphairahedron whose dihedral angles are $\pi/3$. The limit set shown in Figure ??(??) 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 ??(??). The hollows rise, and they emerge on the plane as balls as presented in Figure ??(??). As described in the previous section, the limit sets of semi-sphairahedra are no longer quasi-spheres or quasi-fuchsian. Figure ?? 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 ?? and Figure ?? 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 ???. The infinite sphairahedron in Figure ??(??) 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 ??(??) 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 ??.

4.4.7 Breaking Ideality or Rationality

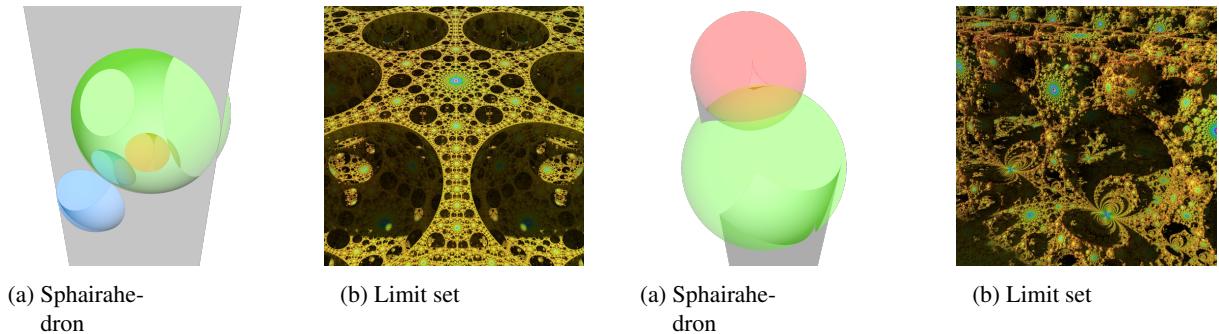


Figure 62: cube-type sphairahedron corresponding to the Figure 63: Cake-type sphairahedron same type as Figure outside of the parameter space in Figure ??.

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 ?? 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 ???. 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 ??), 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 ??(??). 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 ??(??).

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

4.4.8 Three-Dimensional Printing



Figure 64: Monochrome printing by Makerbot Replicator Z18 with PLA resin.



Figure 65: Full-colored printing by DMM 3D print.

Yoshiaki Araki also tried to materialize these fractals in 2006 [[araki2006materializing](#)]. He succeeded to materialize quasi-sphere with some methods including three-dimensional printing. However, recently, three-dimensional printing has become more popular. 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. 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.

Secondly, we decimate generated mesh because it often has too many polygon. To decimate meshes, we use *Blender* or *ZBrush*. For monochrome printing, we complete data conversion. See Figure ???. The object printed by Makerbot Replicator Z18 with PLA resin.

Thirdly, if we want to print full-colored fractal data, we have to generate texture for the fractal object. To obtain texture coordinates, we apply UV unwrapping. After UV unwrapping, we obtain a mapping between world coordinates and UV coordinates. We create texture using the coordinates. In this case, we used *ZBrush* to unwrap UV.

Optionally, we can hollow the object to reduce the printing cost.

We can place an order for full-colored three-dimensional printing service. We use *DMM 3D printing* service. See Figure ???. The material is plaster. Figure ?? shows plastic.

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 has a constraint that we have to compute in parallel; nevertheless it is useful. The application range of IIS is broad. We can render many images related to Kleinian groups and tiling using IIS.

On the other hand, there are also 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

We thank our super visor, Kazushi Ahara.

Our family.

Meiji University