

Iterated Inversion System: An Algorithm for Efficiently Visualizing Kleinian Groups and Extending the Possibilities of Fractal Art

Kento Nakamura

Graduate School of Advanced Mathematical Sciences, Meiji University, Tokyo, Japan.

ORCiD: 0000-0002-5628-4925

Email: somaarcr@gmail.com

Web: soma-arc.net

Iterated Inversion System: An Algorithm for Efficiently Visualizing Kleinian Groups and Extending the Possibilities of Fractal Art

ABSTRACT

Kleinian group theory is a branch of mathematics. A visualized Kleinian group often presents a beautiful fractal structure and provides clues for understanding the mathematical properties of the group. However, it often takes much time to render images of Kleinian groups on a computer. Thus, we propose an efficient algorithm for visualizing some kinds of Kleinian groups: the Iterated Inversion System (IIS), which enables us to render images of Kleinian groups composed of inversions as circles or spheres in real time. Real-time rendering has various applications; for example, the IIS can be used for experimentation in Kleinian group theory and for the creation of mathematical art. The algorithm can also be used to draw both two-dimensional and three-dimensional fractals. The algorithm can extend the possibilities of art originating from Kleinian groups. In this paper, we discuss Kleinian fractals from an artistic viewpoint.

KEYWORDS

fractals; visualization; Kleinian groups; circle inversion; sphere inversion

1. Background

1.1. *Emergence of Kleinian group theory and fractals*

The Kleinian group is a generic name for a properly discontinuous group composed of Möbius transformations. The origin of the Kleinian groups goes back to the nineteenth century, and Felix Klein, his apprentices, Robert Fricke, and Henri Poincaré studied the group. The Kleinian group is considered to be deeply related to hyperbolic geometry and is a geometrical concept that has attracted much attention from the beginning.

Klein himself was also deeply interested in the figures of fractals that originated from the Kleinian group. Fricke's students draw the Kleinian fractals by hand. These images are evidence of their quest to study Kleinian groups.

Regarding figures of fractals, Benoit Mandelbrot first used the term *fractal* in 1975. He also advocated the Mandelbrot set, and its wonderful and beautiful graphics were well known. Then, broad interest in the images of fractals grew. People with computers enjoyed their interesting shapes and colours. Additionally, many programmers developed renderers for the Mandelbrot set or other fractals with their computers.

Various fractals and their rendering methods were introduced with the growth of computer technology. There are some internet communities devoted to discussing fractals. For example, there is a bulletin board system about fractals called *Fractal Forums*¹. This forum is one of the largest communities around fractals. The participants display rendered fractals or compete with each other to make them, discuss the composition of the new fractals, and so on. Famous three-dimensional fractals called *Mandelbulb* and *Mandelbox* have been published in this forum. Of course, fractals based on Kleinian groups are also discussed.

In 2002, David Mumford, Caroline Series, and David Wright published a book called ‘Indra’s Pearls’[Mumford et al., 2002]. It is not a technical mathematics

¹<http://www.fractalforums.com/>

book, but it is written for the general public. The book also introduces how to render beautiful fractals related to Kleinian groups using a computer. Then, not only mathematics lovers but also programmers became strongly interested in the images of Kleinian groups. However, it is not easy to render all of the Kleinian fractals in real time using the methods introduced in Indra's Pearls. Additionally, the book addresses only a small part of Kleinian groups.

1.2. Iterated Inversion System and fractal rendering

The visualization of geometry became more artistic after the publication of Indra's Pearls because of its beautifully rendered images. Therefore, from the perspective of art, geometry is seen as a set of unstable objects changed by user operations or randomness, as in generative art. To handle geometry of this kind, it is necessary to render the targets of the geometry in real time. Therefore, the problem of real-time rendering for geometry and fractals involves subjects from arts to computer science. Rendering the Kleinian groups also requires methods of real-time rendering.

In the author's papers, [Nakamura & Ahara, 2016], [Nakamura & Ahara, 2017], and [Nakamura & Ahara, 2018], we propose a new algorithm called the *Iterated Inversion System (IIS)* for rendering fractals strongly depending on inversions in circles or spheres. Then, we argue that the algorithm contributes to rendering fractals based on Kleinian groups in real time.

For example, we showed that the IIS is applicable in rendering three-dimensional fractals generated by a sphairahedron in real time. A sphairahedron is a kind of polyhedron with spherical faces. Sphairahedra meeting some conditions can be tessellated with inversions in the faces. The three-dimensional tessellation often forms a three-dimensional fractal shape, and it can be rendered by the IIS in real time. For more details, see chapter 5.3, [Nakamura & Ahara, 2018], or chapter 4.4 of [Nakamura, 2018].

Approximately three-dimensional fractals originated from Kleinian groups, such as the limit sets introduced in [Sakugawa, 2010] and [Araki & Ito, 2008], but we have not verified the availability of the IIS for all of them.

1.3. Fractal rendering and the demoscene

To develop the IIS, we learned the rendering techniques used in the *demoscene*. The demoscene is a subculture of computer graphics. Demosceners aim to create beautiful movies and music in real time with a small-sized program. Furthermore, in a *demoparty*, demosceners gather and communicate with each other. They also show their works or compete. A demoparty is like an offline meeting among demosceners. The author also took part in a demoparty held in Tokyo, Japan, which is called *Tokyo Demo Fest (TDF)*²; it is the only demoparty in Japan.

Demosceners often use fractals in their artwork because fractals can generate complicated images from simple and short source codes. To visualize Kleinian

²<http://tokyodemofest.jp/2018/?lang=en>

groups in real time, we refer to their techniques. For more details on the demoscene, see the documentary film ‘Moleman 2 - Demoscene - The Art of the Algorithms (2012)³.’

The author took part in TDF and submitted OpenGL Shading Language (GLSL) artworks three times and PC 4K graphics once. Some of the author’s works using IIS are highly appreciated in the GLSL Graphics Compo and Combined Graphics Compo at TDF. The GLSL Graphics Compo is a competition in real-time graphics rendered in GLSL. The Combined Graphics Compo is a competition for single-image graphics files. In the Combined Graphics Compo, PC 4K Graphics is a competition for 4KB executable files that create graphics.

‘Indra’s Bubbles’ won 2nd place in the GLSL Graphics Compo at TDF 2016. At TDF 2017, ‘SchottkyWaltz’ won 1st place in the GLSL Graphics Compo, and ‘TokyoDemoFesTessellation’ won 2nd place in the Combined Graphics Compo. Finally, the author entered ‘Morph’ into the GLSL Graphics Compo at TDF 2018. These works are also described in chapter 5.4.

To render fractals using the IIS in real time, we have to use parallel processing. To prepare a parallel processing environment easily, we use a fragment shader in the GLSL. For more details about shaders, read ‘The Book of Shaders⁴.’

1.4. IIS from various perspectives

The IIS can be used for various mathematical objects. In this paper, we discuss the aesthetic properties and mathematical meanings of artworks rendered using the IIS.

From the artistic viewpoint of the IIS, the proposition of a new artificial expression is a dynamic drawing that is produced by real-time rendering. For example, we show the dynamic and generative generation of Kleinian fractals. When we make movies of fractals, we edit fractals or other components in real time to find good parameters.

The author has published interactive web applications using the IIS, such as *Schottky Link*⁵, the *Sphairahedron-based Fractal Renderer*⁶, *Flower*⁷, and so on. They are also introduced in the following sections.

On the other hand, from a mathematical perspective, the author’s web applications, such as Schottky Link, the Sphairahedron-based Fractal Renderer, and Flower, are dynamic geometrical visualization tools. They succeed in extracting visual geometric features from parabolic, elliptic, and loxodromic (hyperbolic) Möbius transformations. Moreover, it is useful to have a simple simulator for the tiling patterns or limit set generated by a finite generated Möbius transformation group. Such simulators are expected to support the discovery of new mathematical theorems and the study of Kleinian group theory.

The author also wrote a mathematical paper based on the sphairahedron in [Nakamura et al., 2020]. In the article, we experimentally render the tiling of

³<https://www.youtube.com/watch?v=iRkZcTg1JWU>

⁴<https://thebookofshaders.com/>

⁵<https://schottky.jp>

⁶<https://sphairahedron.net>

⁷<https://soma-arc.net/Flower>

the sphairahedron using the author's renderer. In a sphairahedron-based fractal renderer, we can change the viewpoint of the camera and the parameters interactively. From the experiment, we can classify the variations of sphairahedra, discover propositions and reformulate open problems.

The IIS provides a good example of how information technology bridges mathematics and art through the rendering of fractals based on Kleinian groups.

2. Preparation

In this section, we introduce some mathematical terms and prerequisites for understanding the IIS algorithm and the basic usage of the IIS. They are also discussed in chapter 2 of [Nakamura, 2018].

2.1. Transformation group theory

In this paper, we use the terms for Kleinian groups used in Indra's Pearls. The word *group* represents an algebraic group. Additionally, a transformation group is an algebraic group consisting of transformations.

We introduce the Möbius transformation group and Kleinian group as subgroups of the Möbius transformation group. In this paper, we mainly consider the cases $X = \hat{\mathbb{C}}$ and $X = \hat{\mathbb{R}}^3$. In our context, we assume proper topologies on $\hat{\mathbb{C}}$ by the complex topology of $P^1(\mathbb{C})$ and on $\hat{\mathbb{R}}^3$ by one-point compactification from \mathbb{R}^3 , respectively.

We next introduce the Möbius transformation group on $\hat{\mathbb{C}}$ and $\hat{\mathbb{R}}^3$. First, the two-dimensional Möbius transformation group $\text{Möb}(\hat{\mathbb{C}})$ is the set of linear fractional transformations $f(z) = \frac{az+b}{cz+d}$, where a, b, c , and d are complex numbers that satisfy $ad - bc = 1$. A linear fractional transformation is a conformal orientation-preserving homeomorphic map.

2.2. Inversions in circles and spheres

In the two-dimensional cases, it is well known that Möbius transformations on $\hat{\mathbb{C}}$ are composed of an even number of circle inversions and line symmetries. Here, if a circle is centred 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 $f(z) = \frac{R^2}{\overline{z-C}} + C$. Since $f \circ f$ is the identity map, the inversion map is a homeomorphism on $\hat{\mathbb{C}}$. Note that the inversion is an orientation-reversing conformal map on $\hat{\mathbb{C}}$.

We often distinguish the inversion from a line symmetry map. However, in this context, by interpreting the line on the complex plane as a circle centred at infinity, we also call it a *half-plane*. We may regard the line symmetry as a kind of inversion. In this category, any inversion is orientation-reversing, conformal, and homeomorphic on $\hat{\mathbb{C}}$. When we consider the IIS, we use the concept of the inner part of a circle and consider whether a given point is inside or outside of the circle. In the same way, in the case of line symmetry, the half-plane is also taken to be inside for convenience.

Here, we 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, noting that a circle inversion preserves angles, we obtain that the Jacobian mapping is given by the composition of multiplication by a complex constant and complex conjugate. Thus, the Jacobian determinant is as follows: let $P \in \mathbb{C}$, let R be the radius of the circle, and let C be the centre of the circle; then

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

In the same way as for the definition of a circle inversion, a sphere inversion and a Jacobian of the sphere inversion can be defined. Moreover, the definition of a sphere inversion can be extended to a plane symmetry in the same way as a line symmetry in the two-dimensional case. A sphere centred at infinity is called a *hyperplane*. We also regard the plane symmetry as a kind of inversion.

2.3. Möbius transformations and hyperbolic geometry

First, we review the definition of 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 $PSL_2\mathbb{C}$ -actions on $\hat{\mathbb{C}}$. Here, $PSL_2\mathbb{C}$ is the projective space of 2×2 complex matrices A with $\det(M) = 1$. A Möbius transformation is defined as a linear fractional transformation $f(z) = \frac{az + b}{cz + d}$ for the complex variable z , where the 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 $PSL_2\mathbb{C}$, we find that this map φ is a group isomorphism, and it yields an induced action of $PSL_2\mathbb{C}$ on $\hat{\mathbb{C}}$. In what follows, we regard the action of linear fractional transformations and that of $PSL_2\mathbb{C}$ without distinction. For more details about the Möbius transformation, refer to chapter 3 of [Mumford et al., 2002] or chapter 1.1 of [Marden, 2016].

The Möbius transformation is deeply related to hyperbolic geometry. When all of the coefficients in $f(z) = \frac{az + b}{cz + d}$ are real numbers, $f(z)$ preserves the upper half-plane. Moreover, $f(z)$ preserves the hyperbolic metric $\frac{dz^2}{(\text{Im}z)}$ 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 of as an isometric transformation group on the hyperbolic space \mathbb{H}^3 . We

now consider the upper-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 a set of isometric transformations. This extension is called a ‘Poincaré extension’. As above, researching the structures of the Möbius transformation groups is closely related to studying three-dimensional hyperbolic geometry and hence hyperbolic manifolds. For more details, refer to [Marden, 2016].

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 two ways is as follows: One is as a 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 definition is as a representation of the isometry group of the four-dimensional hyperbolic space. To represent a three-dimensional Möbius transformation using a matrix, we consider a quaternion 2×2 matrix group called $Sp^k(1, 1)$. Regarding this topic, refer to chapter 2 of [Sakugawa, 2010].

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}^4 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 a 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 can be given in this way. From 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 identity map, we classify Möbius transformations into three types: *elliptic*, *parabolic*, and *loxodromic*. Here, $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$, $ad - bc = 1$, and $\tau_A = \text{tr}A = \pm(a + d)$.

When A is elliptic, A is conjugate to $z \mapsto ze^{2i\theta}$, with $2\theta \not\equiv 2\pi$ and $\tau_A \in (-2, +2)$. The fixed points are two distinct points.

If A is parabolic, A is conjugate to $z \mapsto z + 1$, with $\tau_A = \pm 2$ and $A \neq id$. The number of fixed points is one.

When A is loxodromic, A is conjugate to $z \mapsto \lambda^2 z$, with $|\lambda| \neq 1$ and $\tau_A \in \mathbb{C} \setminus [-2, +2]$. The number of fixed points of A is two. Additionally, 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 points; one of the 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 $n \rightarrow -\infty$. Additionally, a loxodromic transformation whose trace is real is called a *hyperbolic* transformation.

There are detailed classification tables in Figure 3.8 of [Mumford et al., 2002] and chapter 1.1 of [Marden, 2016].

The set of three-dimensional Möbius transformations also has a similar classification into six types. In the three-dimensional case, each type—‘parabolic,’ ‘elliptic,’ and ‘loxodromic’—has a modifier: *simple* or *compound*. Thus, there are six variations of classification types, for example, ‘simple elliptic’ or ‘compound loxodromic,’ but ‘simple’ is often omitted. The classification of these six terms requires technical mathematics. Thus, for more details, see chapter 2.2 of [Sakugawa, 2010].

2.5. Kleinian groups

The Kleinian group is a group named after Felix Klein. 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 is only a finite number of elements $g \in G$ such that $\gamma(K) \cap K \neq \emptyset$.

When the G -action is properly discontinuous, the ϵ neighbourhoods of points in the orbit space are disjoint for a positive ϵ . It follows that there exists a fundamental domain with positive volume in \mathbb{H}^3 . Because a Möbius transformation yields an isometric transformation of \mathbb{H}^3 , a Kleinian group yields a hyperbolic three-manifold as a quotient space of the group. The basic properties of a Kleinian group are described in chapter 2 of [Marden, 2016].

In this paper, we consider Möbius transformations in a broad sense, that is, as maps that are a composite of any number of inversions. To distinguish them from the original Möbius transformation, we call them *extended Möbius transformations*. An extended Möbius transformation induces an isometric mapping (not necessarily orientation-preserving) on the three-dimensional hyperbolic space \mathbb{H}^3 by Poincaré extension. 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 $\Lambda(G)$. Useful properties for visualizing the limit set are described in chapter 2.4.1 of [Marden, 2016].

Here, we introduce some concepts to describe the limit points of an orbit: an algebraic limit and a geometric limit. When we fix a set of generators of a

Kleinian group G , we represent an element 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 point is called an algebraic limit of G .

On the other hand, the original definition of a limit point is obtained through a geometrical way of thinking. Thus, such a limit point is also called a geometric limit.

For more details about the algebraic limit and geometric limit, refer to chapters 4.1 and 4.4 of [Marden, 2016].

3. Basic methods for visualization

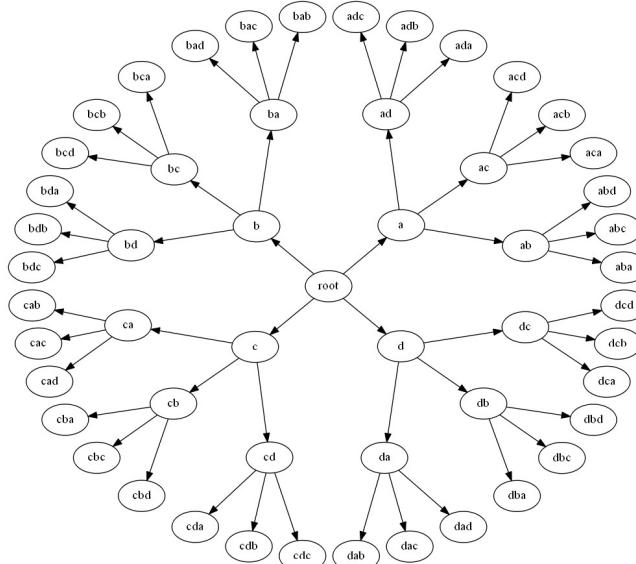


Figure 1.: Cayley Graph

In this section, we introduce basic methods for visualizing extended Kleinian groups.

First, to visualize a group, we consider a *Cayley graph*. A Cayley graph is defined as follows: Let G be a group, and fix a set of generators. A node of a Cayley graph is an element of G , and this element is also called a *word*. An edge of a Cayley graph connects two nodes that are related by the right-multiplication of a generator.

For example, regarding the group

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

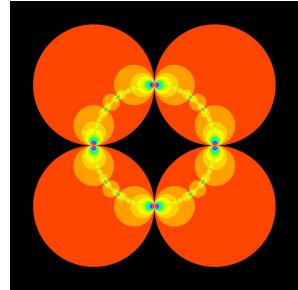


Figure 2.: Orbit space of the disks

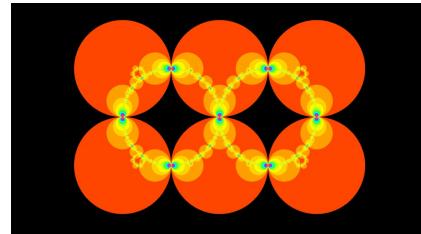


Figure 3.: Circle inversion fractal composed of six disks

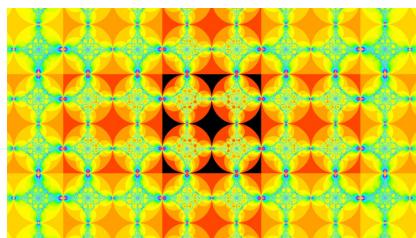


Figure 4.: Circle inversion fractal composed of four disks and four half-planes

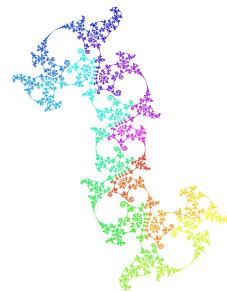


Figure 5.: Limit set of the Kleinian group

the Cayley graph of G is shown in Figure 1. The generators a , b , c , and d are inversions of circles. The visualized image is shown in Figure 2. In this image, the disks nest infinitely. We call the image *circle inversion fractals*. We draw the orbit space of the disks. These are the largest four disks in Figure 2, and we call their circle inversions a , b , c , and d . The four disks 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 isomorphic to the above G .

Figure 3 and Figure 4 show other examples of circle inversion fractals. For more details about circle inversion fractals, see the introduction of [Nakamura & Ahara, 2017].

To draw the orbit space of the disks, we use a *breadth-first search* algorithm. In what follows, we explain the breadth-first search algorithm. First, we draw the four disks, and next, we draw the transformed disks by words whose length is one, and we draw the new, transformed smaller disks by words whose length is two. We continue iterating these processes until the length of the words reaches a predetermined length. Finally, we have completely drawn the orbit space of the disks. This is a breadth-first search algorithm. In this paper, we mainly deal with fractals rendered by a breadth-first search.

Additionally, there is another method of traversing the graph, called a *depth-first search* algorithm. Using this algorithm, we can draw only the limit set directly; see Figure 5. This shows the limit set of a Kleinian group. In Indra's Pearls, this method is primarily used.

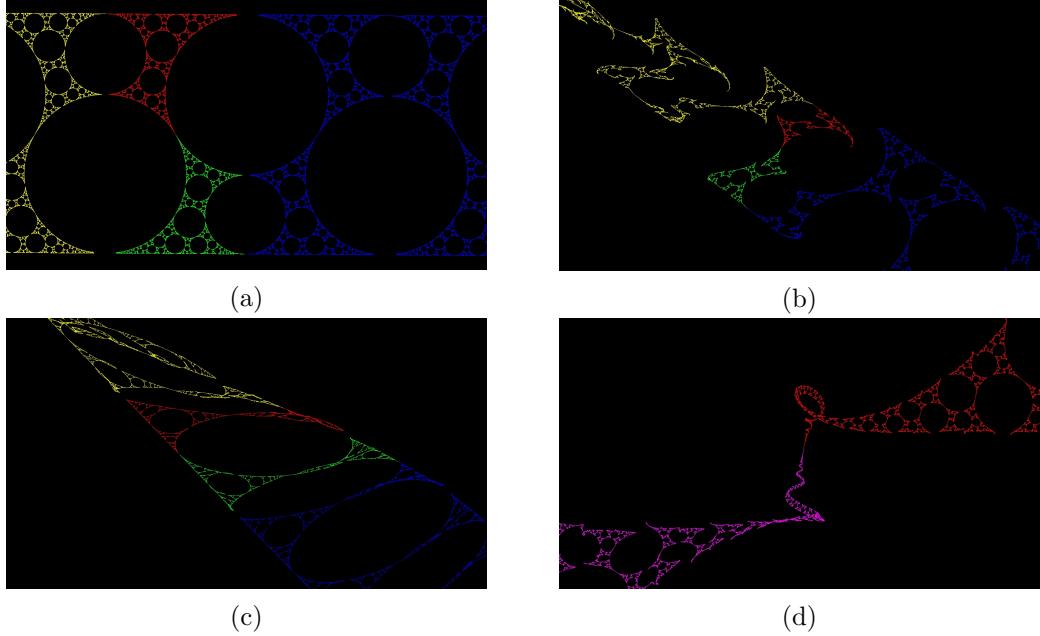


Figure 6.: The limit set of the four-dimensional Kleinian groups

Here is another example: Keita Sakugawa introduced a family of four-dimensional Kleinian groups in his master's thesis. However, he did not publish his paper, and it is written in Japanese. The limit sets of the four-dimensional

Kleinian groups have three-dimensional torsion, as shown in Figure 6. They are rendered using a depth-first search of the Cayley graph by the author.

Despite the attractiveness of the images, there are few examples of the families of four-dimensional Kleinian groups because of the complexity of the generators. Generally, Möbius transformations acting on three-dimensional space are composed of a 2×2 quaternion matrix called $Sp^K(1,1)$. The image of the limit set is more complicated than the group introduced in Indra's Pearls. The calculation of quaternions takes much time.

Visualization by these algorithms is easy for us to implement and understand. However, these methods have the drawbacks that the computational complexity of the Cayley graph increases easily, and they take too much time.

For more details on graph traversal methods, see chapter 4 of ‘Breadth-first Search’ at pp. 113-117 and chapter 5 of ‘Depth-first search: how to extricate oneself from a maze’ at pp. 141-150 of Indra’s Pearls.

4. IIS in two dimensions and three dimensions

The contents of sections 4.1 and 4.2 have already been described in [Nakamura & Ahara, 2017]. However, they are important; therefore, we introduce them again in the next sections.

4.1. Two-dimensional IIS

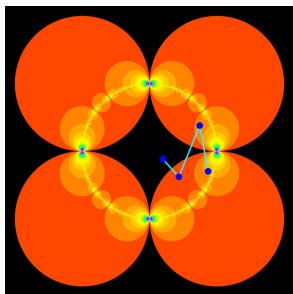


Figure 7.: Orbit of the blue point by the IIS

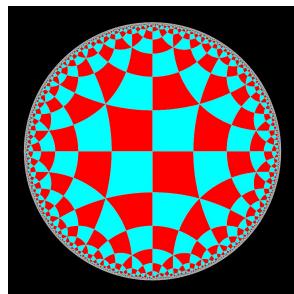


Figure 8.: Hyperbolic tessellation

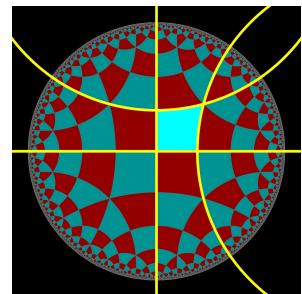


Figure 9.: Fundamental tile

To solve the problems of graph traversal approaches, we focus on circle inversions and propose an efficient algorithm to visualize the fractals originating from circle inversions. This algorithm is called the *Iterated Inversion System (IIS)*. It can visualize not only two-dimensional fractals but also three-dimensional fractals originating from sphere inversions.

The IIS calculates the depth of the overlapping disks point by point. The process of the IIS is as follows: For each point on the plane, if the point is contained in one of the initial inversion disks, we apply inversion in the circle. We continue iterating inversions until the point has moved to the outside of the initial disks. Finally, we colour the original point according to the number

of iterations of inversions or choose a colour for the final point. Note that we have to predetermine the maximum number of iterations because a point that is exactly on the limit set never reaches the outside of the circles.

Figure 7 shows the blue point being moved by the IIS. The point is inverted three times. Therefore, the depth of the point is three. The calculation is independent at each point. Therefore, we can easily parallelize the IIS and render the resulting fractal quickly.

The IIS has various applications other than circle inversion fractals. For example, we can render the hyperbolic tessellation shown in Figure 8 using the IIS. To render a hyperbolic tessellation with the IIS, first, we prepare the fundamental first tile and the circles forming the edges of the tile. Figure 9 shows the first tile in the case of Figure 8. The tile is formed by four circles (two normal circles and two half-planes.)

For each point on the plane, if the point is contained in one of the circles, we apply inversion in the circle. We continue iterating inversions until the transformed point is on the first tile, and we colour the original points according to the number of inversions. Finally, we obtain an image of the hyperbolic tessellation, as shown in Figure 8.

In summary, to use the IIS, we need a fundamental initial tile and a set of transformations that transform a point into the fundamental initial tile. The process of the IIS is the reverse of tiling. Because inversions are involution functions ($f(x) = f^{-1}(x)$), we simply apply inversions to the point. Additionally, the circle inversion fractal can be seen as a tiling of the black area.

The breadth-first search approach has to compute the centres and radii of circles and draw them. The number of circles increases exponentially. The number of circles is determined as follows: Let d be the depth of the reflections, and let n be the initial number of disks. The number of disks is as follows:

$$\sum_{k=1}^d n(n-1)^{k-1}.$$

On the other hand, the number of IIS calculations is a polynomial order of the number of iterations and the number of inversion circles for each pixel. Each computation is atomic; thus, we can perform parallel processing for the IIS.

For more details about the IIS, see ‘Iterated Inversion System’ in [Nakamura & Ahara, 2017] or chapter 3.2.1 of [Nakamura, 2018].

4.2. Three-dimensional extension

In a similar manner to the two-dimensional algorithm, we extend the IIS to visualize three-dimensional fractals related to Kleinian groups. We replace the circle inversions with sphere inversions in the IIS, and we compute the nesting depth of the spheres voxel by voxel using the IIS. To visualize the voxel data, there is a *volume rendering* algorithm. The volume-rendered sphere inversion fractals are shown in Figure 10. It has possibilities for artistic expression, but it is not easy to efficiently render the data, and the visualized images are not

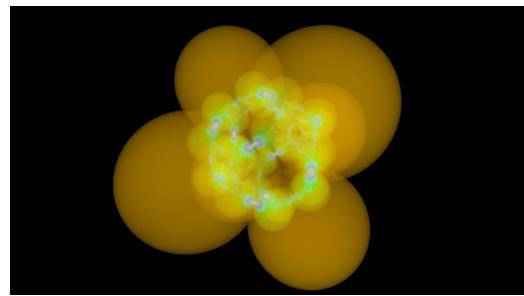
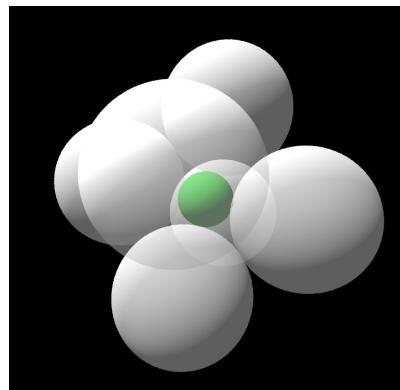
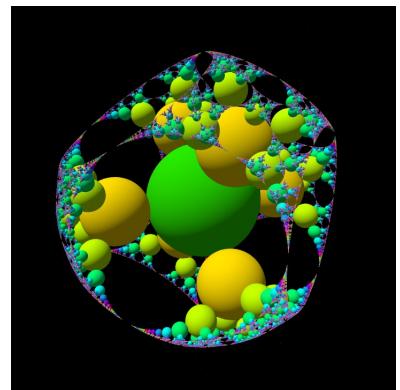


Figure 10.: Volume-rendered nesting spheres

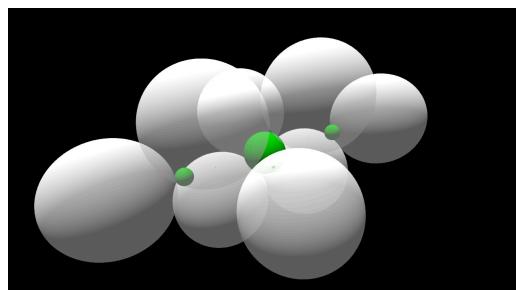


(a) Generator

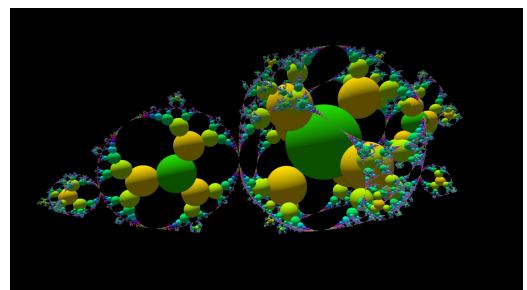


(b) The orbit of spheres

Figure 11.: The orbit of the sphere inversion fractal



(a) Generator



(b) The orbit of spheres

Figure 12.: Another orbit of the sphere inversion fractal

helpful in studying Kleinian groups.

Therefore, we render the orbit of the spheres; see Figure 11(a), which shows six white inversion spheres and a green seed sphere. Figure 11(b) shows the orbit of the green seed sphere transformed by inversions in the white spheres. The small orbit of the spheres is a part of the limit set of the Kleinian group. Additionally, Figure 12 shows other generators and orbits. It has three seed spheres and eight inversion spheres.

We use *ray tracing* to visualize three-dimensional objects. Ray tracing computes an intersection between a ray and objects algebraically, and we work with a ray like a vector. However, the fractal is composed of many spheres. Thus, it is difficult to compute all of the intersections. Therefore, we use the *sphere tracing* [Hart, 1996] algorithm, which is a kind of ray tracing technique.

In sphere tracing, we make the tip of the ray march along the direction of the ray step by step. To check the distance between the tip of the ray and the objects, we use a *distance function*. The distance function returns the minimum distance between a given point and the objects.

However, regarding fractal rendering, it is not easy to obtain an actual distance from a given point to the objects. Thus, we use a lower estimated distance. We approximate the distance using a technique called *distance estimation*.

When we compute the distance to the orbit of the sphere and the tip of the ray, we apply the IIS to the tip of the ray using sphere inversions instead of circle inversions. Finally, we compute the distance between the seed sphere and the transformed tip of the ray. However, the inversion transformations cause an error. Therefore, we use the Jacobian (Jacobian determinant) of the inversions. We accumulate the Jacobian by multiplying it for each inversion. Then, we divide the distance between the seed sphere and the transformed point at the outer area by the accumulated Jacobian. Finally, we obtain the approximate distance between the tip of the ray and the nearest sphere. For the above case, the distance is a lower bound for the spheres.

Additionally, the above computation is a rough estimate, which causes artefacts. For example, the transformed point is outside of the limit set, the distance is unintentionally long, and the ray can pass through the orbit of the spheres. To avoid artefacts, we shorten the estimated distance by multiplying by a scaling factor. Shortening the distance increases the number of steps of sphere tracing, but we can eventually obtain the intersection of the ray and the spheres.

For more details about sphere tracing and its rendering technique and pseudo-code, see ‘Iterated Inversion System’ in [Nakamura & Ahara, 2017] and chapter 3.2.2 of [Nakamura, 2018].

4.3. Related works

Aaron Montag introduced the texture-based approach in his bachelor’s thesis[Montag, 2014]. The algorithm is as follows: First, we prepare the initial seed disk in the texture. Next, we apply the generators of the group to each point on the plane. If the transformed point is on the seed disk or filled point, we fill the original point. We continue iterating the second process; we obtain

an image of the limit set of the group. His implementation can be seen in the gallery of CindyJS⁸.

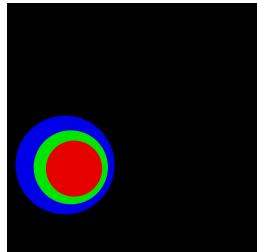
However, the algorithm is difficult to extend in three dimensions because three-dimensional voxel data require considerable computer memory. Additionally, it is difficult to draw enlarged images because they require more pixels.

Martin von Gagern and Jürgen Richter-Gebert introduced an algorithm called *Reverse Pixel Lookup*[Gagern & Richter-Gebert, 2009]. They aimed to render two-dimensional hyperbolic tessellations. This algorithm is similar to the IIS, but we aim to use the method to render Kleinian fractals and three-dimensional objects.

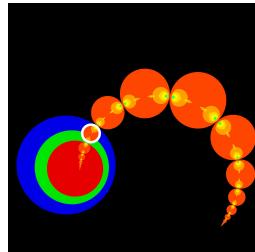
5. Implementation examples

In this section, we show a variety of uses of the IIS.

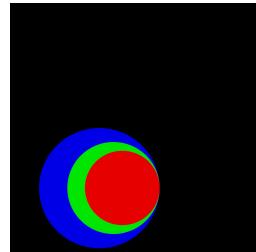
5.1. Geometrical representation of Möbius transformations



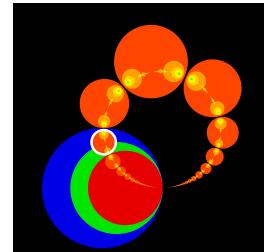
(a) Generator



(b) Orbit



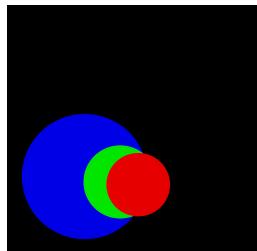
(a) Generator



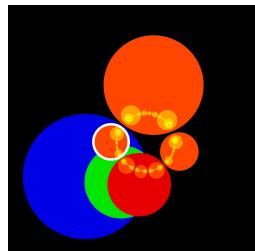
(b) Orbit

Figure 13.: Hyperbolic

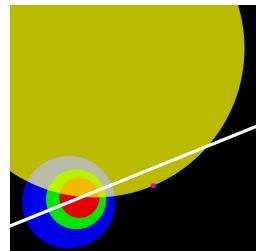
Figure 14.: Parabolic



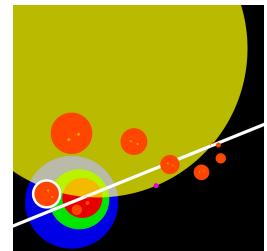
(a) Generator



(b) Orbit



(a) Generator



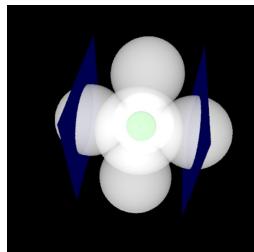
(b) Orbit

Figure 15.: Elliptic

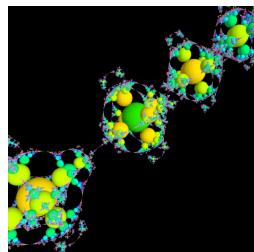
Figure 16.: Loxodromic

Thus far, we have only used a simple circle inversion or sphere inversion. Varieties of images can be rendered using other types of Möbius transformations. It is known that all of the two-dimensional and three-dimensional Möbius

⁸<https://cindyjs.org/gallery/main/Kleinian/>

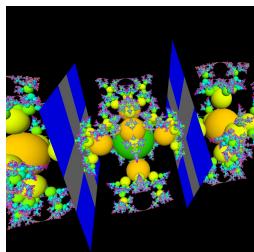


(a) Generator

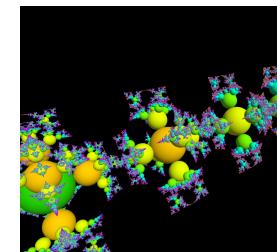


(b) Orbit

Figure 17.: Parabolic (parallel translation)

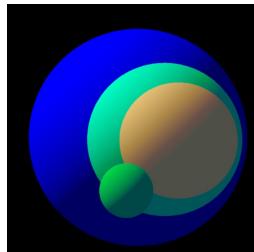


(a) Generator

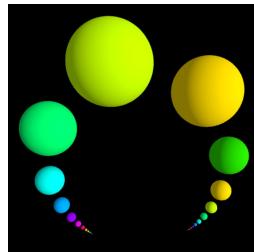


(b) Orbit

Figure 18.: Compound parabolic

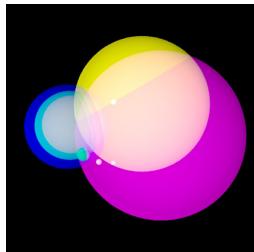


(a) Generator

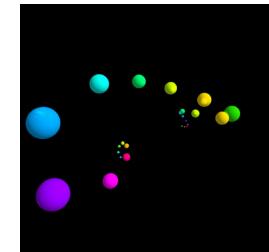


(b) Orbit

Figure 19.: Loxodromic

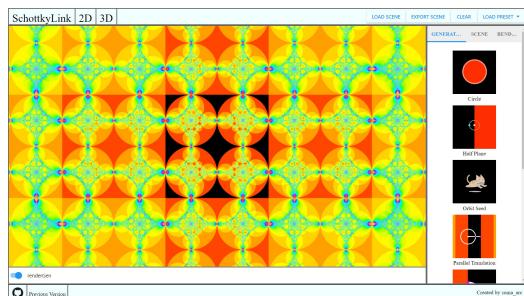


(a) Generator

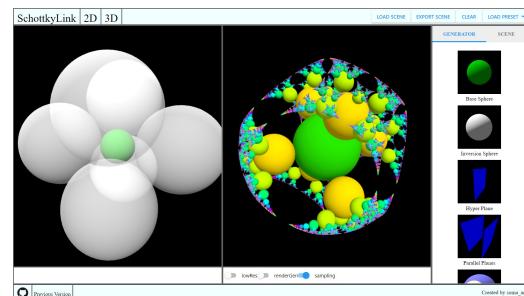


(b) Orbit

Figure 20.: Compound loxodromic



(a) Two-dimensional mode



(b) Three-dimensional mode

Figure 21.: Schottky Link

transformations are composed of a finite number of circle or sphere inversions, respectively. Extended Kleinian groups are composed of any numbers of Möbius transformations.

As noted in chapter 2.4, Möbius transformations are classified into three types. One of the solutions for visualizing extended Kleinian groups is to render the orbit of the disks transformed by Möbius transformations.

All of the Möbius transformations are composed of inversions in circles or spheres, that is, geometrical objects. Therefore, we can change the parameters of Möbius transformations by operating on the positions or radii of circles or spheres. This enables us to create intuitive geometrical constructions. In what follows, we show some examples of Möbius transformations. We also attach the animated example made with the GLSL in the footnote.

5.1.1. Two-dimensional IIS

As described above, the IIS needs a fundamental initial tile and transformations that move a given point to the fundamental tile. When we use circle inversions, we simply apply inversions because a circle inversion is an involution function. In this paper, we call the transformation that moves a point to the fundamental initial tile G .

First, see Figure 13(a), which shows hyperbolic transformations. The three overlapping disks represent the transformation. Let $C1$ be a red disk, let $C2$ be a green disk, and let $C1'$ be a blue disk. $C1$ and $C2$ are the parameters of the transformation. $C1'$ is $C1$ inverted by $C2$.

The fundamental area is changed to a green and blue area. Then, we have to put disks in the area; see Figure 13(b). The white-edged disk is transformed by inversions in $C1$ and $C2$ ⁹. The transformation G is described below. The prefix I represents an inversion; for example, I_{C1} represents an inversion in $C1$.

$$G = \begin{cases} I_{C2} \circ I_{C1} & (\text{The point is inside of } C1) \\ I_{C1} \circ I_{C2} & (\text{The point is outside of } C1') \end{cases}$$

When the disks $C1$ and $C2$ are kissing at one point, as in Figure 14(a), the transformation becomes parabolic. The orbit of the parabolic transformation converges to the kissing point, as shown in Figure 14(b)¹⁰.

When $C1$ and $C2$ are crossing, as in Figure 15(a), the transformations become elliptic. The orbit of the disks is rotated around the line composed of the crossing points $C1$ and $C2$; see Figure 15(b). Note that the angles crossing $C1$ and $C2$ should be rational angles; otherwise, the orbits of the disks will overlap each other.

Figure 16 shows the loxodromic transformation. It is composed of four disks, a line, and a pink control point. The orbit is a spiral, as in Figure 16(b). The transformation is composed of inversions in $C1$ and $C2$, the circumference of the yellow disk, and the reflection through the white line. The torsion is oper-

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

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

ated by the pink point¹¹. The loxodromic transformations are composed of four inversions. G is as follows: Let $C3$ be a yellow circle and let L be a white line.

$$G = \begin{cases} (I_{C2} \circ I_{C1}) \circ (I_{C3} \circ I_L) & (\text{The point is inside of } C1) \\ (I_L \circ I_{C3}) \circ (I_{C1} \circ I_{C2}) & (\text{The point is outside of } C1') \end{cases}$$

5.1.2. Three-dimensional IIS

Möbius transformations act in three dimensions as follows: A parabolic transformation is conjugate to a parallel translation, as in Figure 17. The blue plates are a part of the hyperplane. The plane applies plane symmetry. The two facing hyperplanes generate a parallel translation. A compound parabolic transformation adds a rotation around the translation axis, as in Figure 18¹². The torsion of two hyperplanes represents the amount of rotation.

There are loxodromic and parabolic transformations similar to two-dimensional transformations. They are also composed of three spheres¹³. When the spheres kiss at one point, they become a parabolic transformation. Let $S1$ be a red sphere, let $S2$ be a green sphere, and let $S1'$ be a blue sphere. $S1$ and $S2$ are the parameters of the transformation. $S1'$ is $S1$ inverted by $S2$.

$$G = \begin{cases} I_{S2} \circ I_{S1} & (\text{The point is inside of } S1) \\ I_{S1} \circ I_{S2} & (\text{The point is outside of } S1') \end{cases}$$

In a three-dimensional Möbius transformation, the simple generators are similar to those of two-dimensional Möbius transformations. However, some compound Möbius transformations have torsion. A compound loxodromic transformation has three-dimensional torsion, as in Figure 20¹⁴, which shows a compound loxodromic transformation. Let $S3$ be a purple sphere, and let $S4$ be a yellow sphere. There are three control points: P , $Q1$, and $Q2$. $S3$ and $S4$ are determined by four points. Let P' and P'' be inversions of P in $S1$ and $S2$, respectively; G is as follows:

$$S3 = \text{Sphere}(P, P', P'', Q1) \quad S4 = \text{Sphere}(P, P', P'', Q2)$$

$$G = \begin{cases} (I_{S4} \circ I_{S3}) \circ (I_{S1} \circ I_{S2}) & (\text{The point is inside of } S1) \\ (I_{S2} \circ I_{S1}) \circ (I_{S3} \circ I_{S4}) & (\text{The point is outside of } S1') \end{cases}$$

It is composed of four spheres. The transformation has three control points, and we can tweak the torsion with the control points. This cannot be done with two-dimensional Möbius transformations.

The IIS enables us to render real-time and interactive geometric constructions. For more details of the implementation and other generators, see [Nakamura & Ahara, 2017].

¹¹<https://www.shadertoy.com/view/lsSyDW>

¹²<https://www.shadertoy.com/view/lsjyzK>

¹³<https://www.shadertoy.com/view/ldByDW>

¹⁴<https://www.shadertoy.com/view/MdjyRV>

5.1.3. Schottky Link

The author is developing software called *Schottky Link*¹⁵, which visualizes two-dimensional circle inversion fractals and three-dimensional sphere inversion fractals quickly and intuitively. The software has a two-dimensional mode, as shown in Figure 21(a), and a three-dimensional mode, as shown in Figure 21(b). In two-dimensional mode, we simply add a disk and generators selected from the bar on the right side. In three-dimensional mode, the software has two main panels. The left panel shows the generators of the group and the green seed of the orbit. The right panel displays the orbit of the green sphere. When we operate the generators in the left panel, the orbits of the spheres are deformed according to the generators.

Thanks to the IIS, we can manipulate many generators and render the images in real time. It is easy to explore inversion fractals with Schottky Link.

However, Schottky Link cannot visualize all of the Kleinian groups. For example, the IIS cannot visualize a group that includes two or more loxodromic or parabolic generators because we cannot know that the transformations will move the point into the fundamental tile. Additionally, the IIS cannot render only the limit set.

5.2. Simple visualization techniques

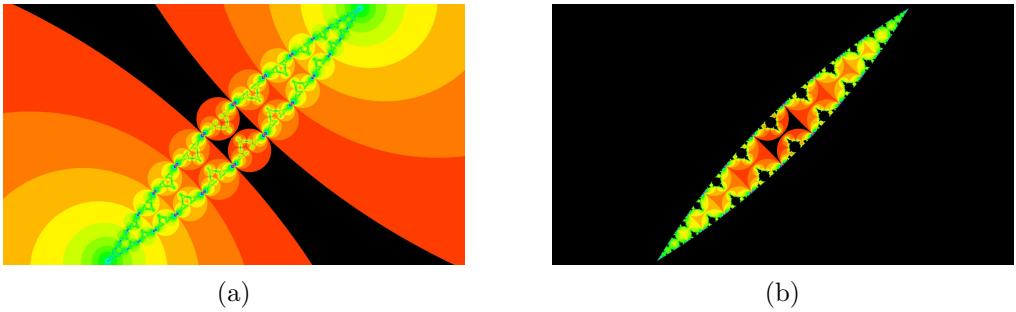


Figure 22.: Edge of the limit set of the circle inversion fractal

In this subsection, we show some simple techniques to render artistic images.

5.2.1. Render internal area

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 22(a). The image is generated by four inversions of the circles.

Circle inversion preserves the interior or exterior part of the limit set. Thus, after applying the IIS, we fill a pixel when the transformed point is moved into an inner part of the fundamental tile, that is, the inner black area. Then, we obtain only the inner part of the circle inversion fractals. This is shown in Figure 22(b).

¹⁵<https://schottky.jp>

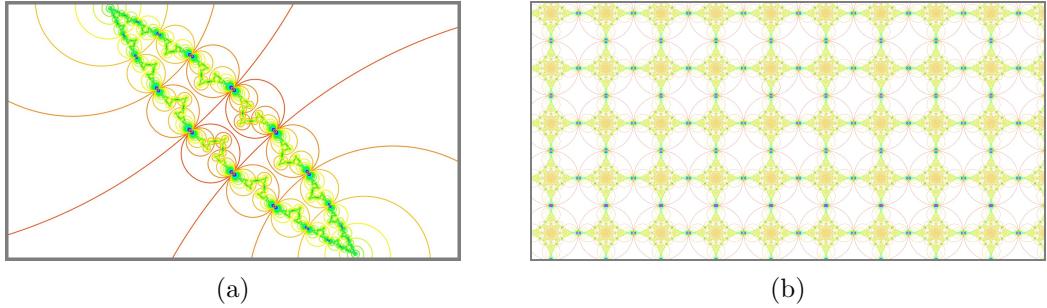


Figure 23.: Circumferences of the circle inversion fractal

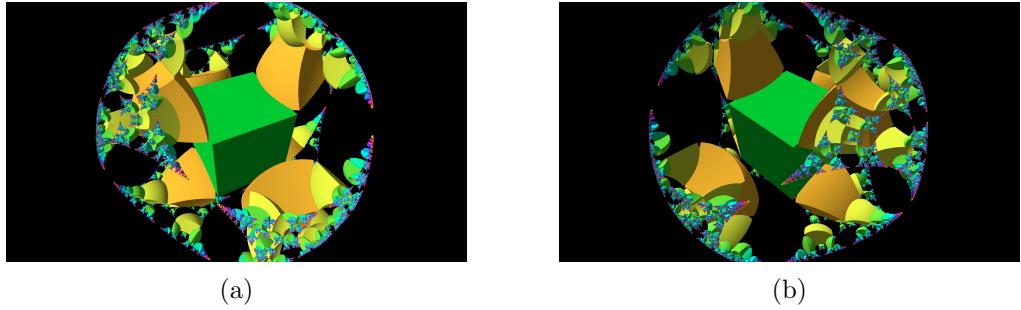


Figure 24.: The orbit of the cube

5.2.2. Rendering circles

In two dimensions, we can render, not the orbit of disks, but rather the orbit of circles using a Jacobian as shown in Figure 23¹⁶. We can estimate the distances between the circumferences of the disks and the point on the initial disks. When we apply circle inversions, we multiply and accumulate the Jacobian of the inversions. When the transformed point is moved to the initial disks, we divide the computed distance between the circumference of the initial disk and the transformed point by the accumulated Jacobian. We then obtain the distance between the circumference of the initial circle and the final point.

5.2.3. Other seed of an orbit

For three-dimensional sphere inversion fractals, we cannot use a seed sphere, but we can use another object expressed by distance functions. For example, we can visualize the orbit of the cube; see Figure 24. The transformed cubes are distorted by sphere inversions.

5.3. Sphairahedra and three-dimensional fractals

The sphairahedron is a geometrical concept invented by Kazushi Ahara and Yoshiaki Araki in 2003 [Ahara & Araki, 2003]. Sphairahedra are polyhedra that

¹⁶<https://www.shadertoy.com/view/4sSfzD>



Figure 25.: Cube-type sphairahedron

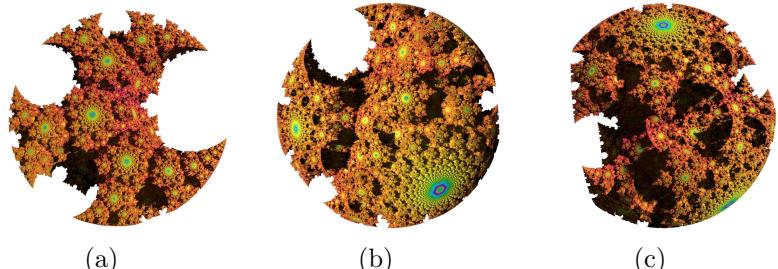
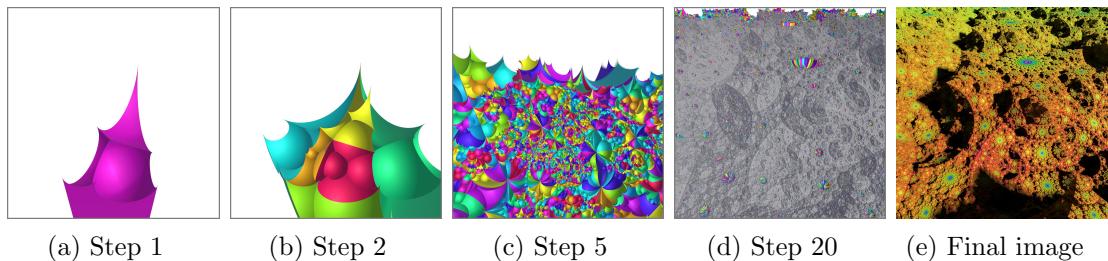


Figure 26.: Images of a quasi-sphere rendered in different viewpoints



(a) Step 1 (b) Step 2 (c) Step 5 (d) Step 20 (e) Final image

Figure 27.: Tiling of the finite sphairahedron



(a) Step 1 (b) Step 2 (c) Step 5 (d) Step 20 (e) Final image

Figure 28.: Tiling of the infinite sphairahedron

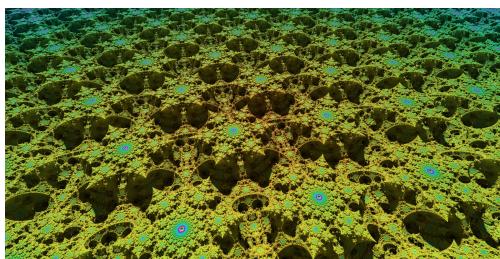


Figure 29.: Fractal terrain



Figure 30.: Sphairahedron-based Fractal Renderer

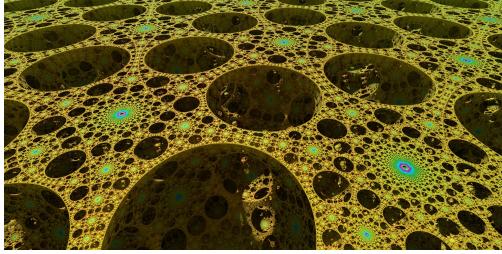


Figure 31.: Bridges

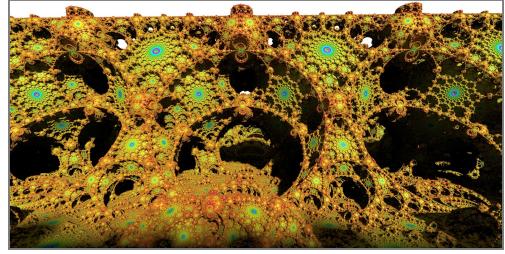


Figure 32.: Tunnels

have spherical faces instead of planar faces. We can visualize three-dimensional tilings using inversions in the spheres that form the faces of the sphairahedron. The inversions compose a Kleinian group. Figure 25 shows a cube-type sphairahedron, and Figure 26 shows the result of a three-dimensional tiling of a cube-type sphairahedron. We also call the sphairahedron a *finite sphairahedron* because all of the vertexes are finite. The union of all the tiles is homeomorphic to a ball. Thus, this boundary is called a *quasi-sphere*. The boundary of the tiling is the limit set of the Kleinian group. Figure 26 is drawn with the IIS. Figure 27 shows the tiling process for a finite sphairahedron.

Figure 28(a) shows a sphairahedron one of whose vertexes is at infinity; this is called an *infinite sphairahedron*. The tiling process is shown in Figure 28. It generates the fractal terrain shown in Figure 28(e). The fractal spreads out quite far, as shown in Figure 29.

For more details about the fractals generated by the sphairahedron, read [Nakamura & Ahara, 2018]. We can obtain various fractals according to the shape of the sphairahedron. Pictures of the fractals are summarized at the author's webpage¹⁷. There are many figures and gif animation files. This page also includes the renderer developed by the author to visualize the sphairahedron and its limit set.

The renderer is shown in Figure 30. There are two large panels at the top of the screen and two small panels at the bottom. The small panel at the bottom right shows the parameter space. The red point represents the current parameter of the sphairahedra. We can see the deformation of the quasi-sphere and sphairahedron at the same time. There are two sphairahedra, that is, a finite sphairahedron and an infinite sphairahedron. A finite sphairahedron is generated by applying a sphere inversion to an infinite sphairahedron. The shape of the finite sphairahedron changes according to the position and radius of the inversion sphere.

In the software, we can operate on the inversion sphere, the maximum number of iterations, the lighting and so on with the side panel. The parameters of a sphairahedron are determined by the number of faces and by combinations of the dihedral angles.

A sphairahedron has two properties, called *ideality* and *rationality*. An ideal rational sphairahedron generates the Kleinian group. Therefore, we compute the parameter space of the ideal rational sphairahedron. The parameter spaces of

¹⁷<https://sphairahedron.net>

various sphairahedra are shown in [Nakamura et al., 2020]. Mathematically, we are not interested in the outside of the parameter space. However, from an artistic perspective, the quasi-sphere outside of the parameter space is also interesting. It can make fractal bridges or tunnels in the resulting fractals, as shown in Figures 31 and 32.

In [Nakamura et al., 2020], we discussed sphairahedra and the fractals generated by them from a mathematical viewpoint. We experimented with the sphairahedron-based fractal renderer, and we found mathematical problems using the software.

5.3.1. Colouring of quasi-spheres

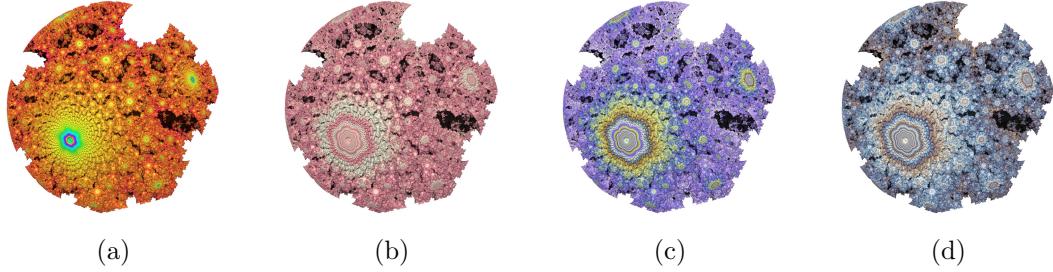


Figure 33.: Quasi-spheres coloured with different colour palettes

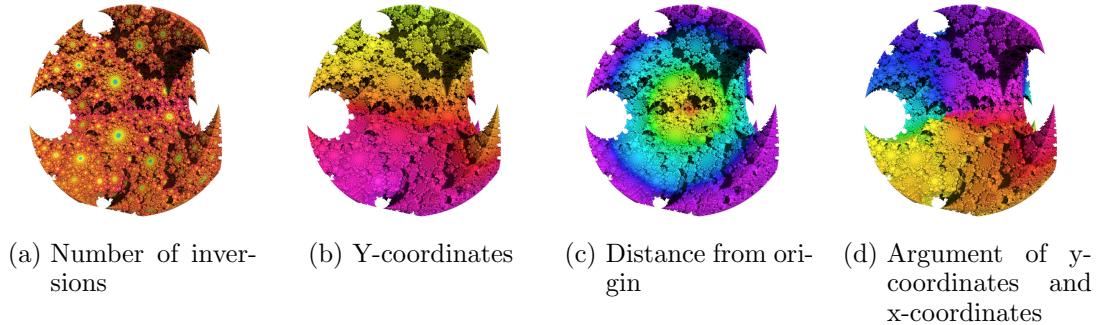


Figure 34.: Quasi-spheres coloured by different parameters

In this section, we discuss the colouring of the fractals. Colouring affects the appearances of fractals. Three-dimensional fractals have infinitely small surfaces. Thus, we cannot determine the colour of a pixel as one colour. Therefore, we take samples within a pixel, and we fill the pixel with the average colour.

Quasi-spheres have fractal surfaces. In this case, we refer to the colour wheel according to the number of inversions of the IIS. We use large steps on the colour wheel so that each tile will be coloured clearly, as shown in Figure 28(b). When we have many iterations, the tiles seem to be grey. In one pixel, there are many sampling points and tiles with various colours. Thus, the many colours are averaged, and the result is grey, as shown in Figure 28(d).

When we use the colour wheel with small steps, the sampled colours are similar, and they are averaged in the pixels. Therefore, the surfaces of the quasi-sphere are also similar colours, as shown in Figure 28(e).

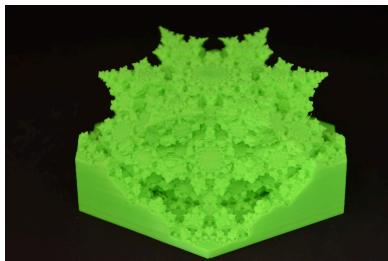
When Ahara and Araki rendered quasi-spheres, they coloured them according to the coordinates of the surfaces of the quasi-sphere¹⁸. In contrast, we colour the quasi-sphere according to the number of inversions applied to the first tile. This method emphasizes the surfaces that gather infinite numbers of tiles, but it has slow convergence. This phenomenon is often seen around parabolic fixed points.

We used a simple HSV colour wheel, but there is an algorithm to generate colour palettes. For more details, read the blog post by Inigo Quilez¹⁹. In Figure 33, we show four different colour palettes.

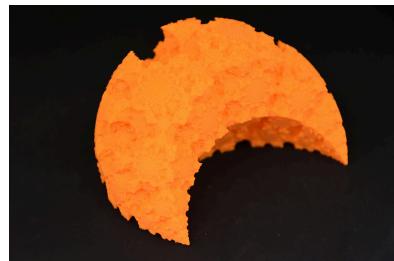
Colouring also depends on the types of parameters, for example, the number of iterations of inversions, the coordinates of a point, the distance between the origin and the point, and the argument of the point. These are shown in Figure 34.

In addition to these colouring methods, there are many ways to use the parameters originating from the IIS.

5.3.2. Three-dimensional printing

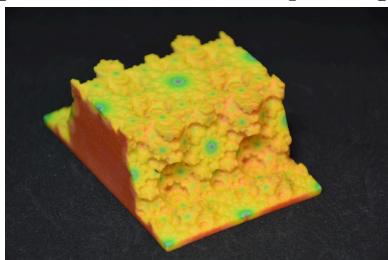


(a)



(b)

Figure 35.: Monochrome printing by MakerBot Replicator Z18 with PLA resin



(a) Plaster



(b) Plastic

Figure 36.: Full-colour printing by a DMM.make 3D PRINT

Three-dimensional printing is a useful way to materialize mathematical objects. Yoshiaki Araki tried to materialize quasi-spheres in 2006[Araki, 2006]. Re-

¹⁸<https://www.youtube.com/watch?v=31c09zRCv-4>

¹⁹<https://www.iquipilezles.org/www/articles/palettes/palettes.htm>

cently, many people have worked on materializing mathematical objects using three-dimensional printers. For example, Jeremie Brunet has made many beautiful three-dimensional printed fractals²⁰. Henry Segerman published a book about three-dimensional printed mathematical objects[Segerman, 2016].

We can also materialize three-dimensional fractals rendered by the IIS. The IIS generates voxel data; therefore, we have to convert voxel data into mesh data. For more details about generating data for three-dimensional printing using the IIS, see chapter 4.4.8 of [Nakamura, 2018].

Figure 35 is made with a monochrome three-dimensional printer. We used ‘MakerBot Replicator Z18’ and PLA resin. We can print the model with a home-use three-dimensional printer; however, full-coloured three-dimensional printing is not popular yet. The author places orders with a three-dimensional printing service. Figure 36 shows full-coloured plaster and plastic models.

The author has published three-dimensional models at Sketchfab²¹. Note that soma_arc is the screen name of the author. These kinds of mathematical objects are interesting models for applications such as art and studying objects in doing mathematics. They allow us to observe the objects from another point of view, and we may find underlying symmetrical properties by physically touching them.

5.4. Shader artwork

The author has published many artistic works with GLSL. They are published at *Shadertoy* with the source code²². Shadertoy is a website used to build and share shaders.

As mentioned in the background section, currently, the author has submitted artwork to the GLSL Graphics Compo at TDF three times. This category is a competition in real-time graphics rendered by the GLSL. All of the author’s artworks are related to Kleinian groups and are rendered using the IIS. All of them are looped animations. They are shown below. Additionally, the author has submitted PC 4K graphics to the Combined Graphics Compo once. This is a competition for 4 KB executable files that generate one image. This is also rendered by the GLSL.

5.4.1. Indra’s Bubbles

‘Indra’s Bubbles’²³ received the GLSL Graphics Compo 2nd place at TDF 2016. Figure 37 shows a scene from the work.

The themes of the work are three-dimensional fractals and reflections. A specular object at the centre is based on the orbit of the Kleinian groups shown in Figure 11(b). The number of the orbit of the spheres increases, and finally, they become one object. The fractal at the centre is a reflective material and reflects the surrounding objects. Mirror reflections by spheres are similar to sphere inversions. The reflections of spheres make beautiful patterns.

²⁰<https://www.shapeways.com/shops/3dfractals>

²¹https://sketchfab.com/soma_arc

²²https://www.shadertoy.com/user/soma_arc

²³<https://www.shadertoy.com/view/XsGGWG>

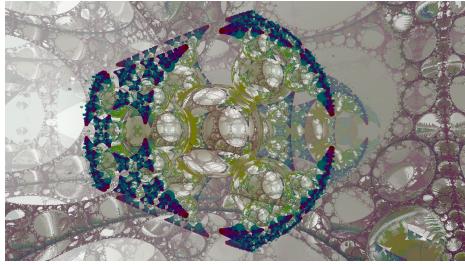


Figure 37.: Indra’s Bubbles

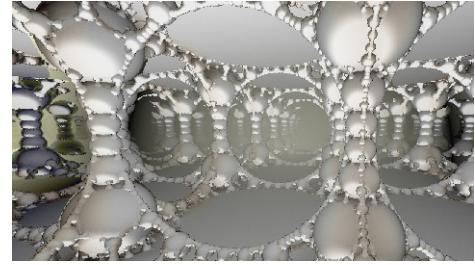


Figure 38.: Pseudo-Kleinian rendered with Fragmentarium

The Kleinian group-like fractal surrounding the spheres is called a *pseudo-Kleinian*. It is a famous kind of fractal similar to the limit set of the Kleinian groups shown in Figure 38. The image is rendered with a renderer called Fragmentarium²⁴. The source code is contained in Fragmentarium.

A pseudo-Kleinian uses a sphere fold operation similar to sphere inversion. Therefore, sphere folds cause Kleinian group-like shapes. For more details about sphere folds, see the blog post by Christensen²⁵.

The rendering algorithm enables us to render the fractal quickly, like the IIS, but the pseudo-Kleinian is not a Kleinian group that mathematicians study. We apply a mirror material to the ceiling, floor, and balls of the pseudo-Kleinian so that all of the space seems to expand infinitely.

Usually, a mirror reflection takes much time to render, but the IIS and sphere tracing enable us to render the work in real time.

5.4.2. Schottky Waltz

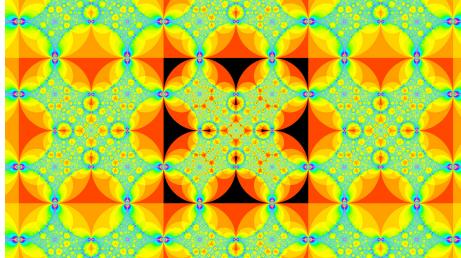


Figure 39.: Schottky Waltz

‘Schottky Waltz’²⁶ received the GLSL Graphics Compo 1st place at TDF 2017. The main theme is circular motion. The movie uses disks and their inversions on the circumference. The fractals are based on extended Kleinian groups.

The rotated and enlarged images make a large change to the graphics. Additionally, the circle inversion fractals are deformed drastically according to the inversions of circles. When the circles overlap each other, we obtain complicated

²⁴<http://syntopia.github.io/Fragmentarium/>

²⁵<http://blog.hvidtfeldts.net/index.php/2011/11/distance-estimated-3d-fractals-vi-the-mandelbox/>

²⁶<https://www.shadertoy.com/view/XslyzH>

and beautiful fractals; note, however, that they are not actual Kleinian groups.

Figure 39 shows a pattern generated by inversions in five circles and reflections by four half-planes. Thanks to the IIS, we can render enlarged images in real time.

When the author made this artwork, we conducted an experiment using Schottky Link to find the parameters that generate nice-looking fractals.

5.4.3. Morph

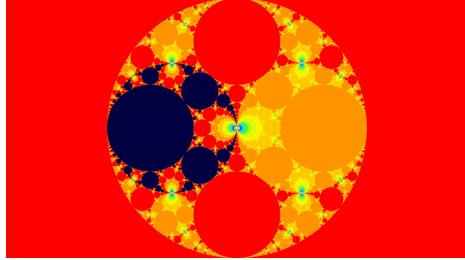


Figure 40.: Morph

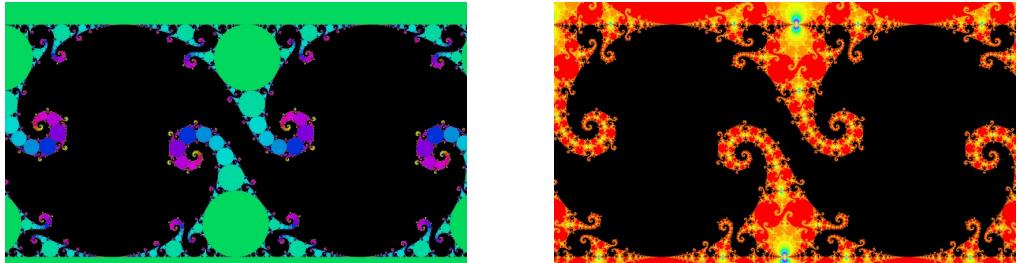


Figure 41.: The limit set of the Kleinian groups with Maskit parametrization Figure 42.: The limit set after applying inversions of the circles

‘Morph’²⁷ was published at TDF 2018; see Figure 40. The themes of the work are the continuous deformation of fractals. We render the disks using Jos Leys’s rendering algorithm for Maskit groups, as shown in Figure 41, and we apply inversions to the disks using the IIS. Then, we obtain Figure 42.

The Maskit group is a subgroup of the Kleinian group. The union of the circumferences of the disks shown in Figure 41 is the limit set of the Maskit group. The shapes of the fractal also change dramatically according to the parameters.

Although the IIS uses circle and sphere inversions, Jos Leys used Möbius transformations algebraically. He observed the orbits of the Maskit parametrization group and discovered an algorithm to visualize Maskit groups. For more details of the algorithm, see his article²⁸.

Jos’s method for rendering Maskit groups enables images of fractals to undergo

²⁷<https://www.shadertoy.com/view/MLGfDG>

²⁸http://www.josleys.com/article_show.php?id=221

continuous deformation. Therefore, the shape of the fractals does not collapse. Additionally, we apply the IIS and obtain flame-like patterns.

5.4.4. TokyoDemoFesTessellation

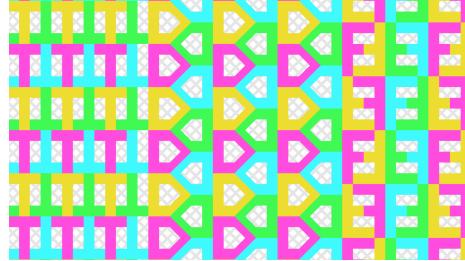


Figure 43.: TokyoDemoFesTessellation

‘TokyoDemoFesTessellation’²⁹ received the Combined Graphics Compo 2nd place at TDF 2017. It is shown in Figure 43 and is rendered with the GLSL and IIS.

This work is a simple tessellation pattern using T, D, and F. It is not related to Kleinian groups but uses the basic IIS idea. In other words, we prepare the original tile and use transformations to move tiles to the original tile.

5.5. Flower: A flow-based programming environment for kaleidoscope patterns

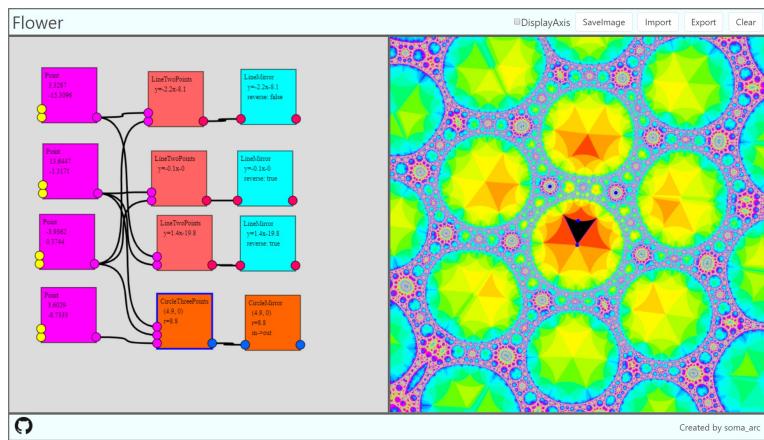


Figure 44.: Flower

Kaleidoscope patterns are generated by mirror reflections. Therefore, the IIS enables us to render kaleidoscope patterns in real time. The author has developed a kaleidoscope pattern generator based on a flow-based program called Flower. Additionally, we can make geometrical constructions interactively, as in GeoGebra, with Flower.

²⁹<https://www.shadertoy.com/view/MsscR4>

Flow-based programming is a programming paradigm in which all of the process and data are represented by nodes. We connect the nodes with edges, and the data are sent from the output socket to the input socket. For more details, see [Nakamura & Ahara, 2020]. Additionally, there is an introductory video on YouTube³⁰.

There are three advantages of flow-based programming. First, it is easy to understand the construction procedure. Second, flow-based programming is flexible. We can insert processes wherever we like easily. Third, flow-based programming has high extensibility. We can create original nodes or scripts easily.

The construction is rendered with a shader and the IIS. We can render most of the kaleidoscope patterns in real time. Figure 44 shows the screen of the software. The left panel shows a node graph. The right panel shows the rendered constructions generated by the node graph, which are rendered by the IIS. It also enables us to render enlarged images of the fractals in real time. This software can be used artistically and educationally.

6. Summary

As noted above, the IIS has various applications and can render in real time. The algorithm is valuable in mathematics and art. Additionally, we seek not only a computational experiment but also beautiful artistic images. Kleinian groups show rich fractal shapes. However, there are many Kleinian groups that we cannot visualize using the IIS; for example, we cannot render the *degenerate group*. Our ultimate goal is to find methods to visualize all of the Kleinian groups in real time.

In particular, four-dimensional Kleinian groups have many unsolved problems and few visualized images. For instance, Sakugawa's families of Kleinian groups, the sphairahedron, and quasi-spheres have unsolved problems.

Moreover, there are still fractals among the Kleinian groups that have not been seen. The IIS will play an important role in their visualization and in finding clues to unravel the unsolved problems of four-dimensional Kleinian groups.

References

- [Ahara & Araki, 2003] Kazushi Ahara and Yoshiaki Araki, *Sphairahedral approach to parameterize visible three dimensional quasi-Fuchsian fractals*, Proceedings of Computer Graphics International, 2003, pp. 226–229.
- [Araki, 2006] Yoshiaki Araki, *Materializing 3D quasi-Fuchsian fractals*, Forma, Volume 21, Number 1, 2006, pp.19–27.
- [Araki & Ito, 2008] Yoshiaki Araki and Kentaro Ito, *An extension of the Maskit slice for 4-dimensional Kleinian groups*, Conformal Geometry and Dynamics of the American Mathematical Society, Volume 12, Number 14, 2008, pp.199–226.

³⁰<https://youtu.be/FWp-eF5gz5o>

- [Gagern & Richter-Gebert, 2009] Martin von Gagern and Jürgen Richter-Gebert, *Hyperbolization of Euclidean Ornaments*, Electronic Journal of Combinatorics, Volume 16, Number 2, 2009.
- [Hart, 1996] John C Hart, *Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces*, The Visual Computer, Volume 12, Number 10, 1996, pp. 527–545.
- [Marden, 2016] Albert Marden, *Hyperbolic Manifolds: An Introduction in 2 and 3 Dimensions*, Cambridge University Press, 2016.
- [Montag, 2014] Aaron Montag, *Interactive Image Sequences Converging to Fractals*, Bachelor Thesis at Technical University of Munich, 2014.
- [Mumford et al., 2002] David Mumford, Caroline Series, and David Wright, *Indra's Pearls: The Vision of Felix Klein*, Cambridge University Press, 2002.
- [Nakamura & Ahara, 2016] Kento Nakamura and Kazushi Ahara, *A New Algorithm for Rendering Kissing Schottky Groups*, Proceedings of Bridges 2016: Mathematics, Music, Art, Architecture, Education, Culture, Tessellations Publishing, Phoenix, Arizona, 2016, pp. 367–370.
- [Nakamura & Ahara, 2017] Kento Nakamura and Kazushi Ahara, *A Geometrical Representation and Visualization of Möbius Transformation Groups*, Proceedings of Bridges 2017: Mathematics, Music, Art, Architecture, Education, Culture, Tessellations Publishing, Phoenix, Arizona, 2017, pp. 159–166.
- [Nakamura & Ahara, 2018] Kento Nakamura and Kazushi Ahara, *Sphairahedra and Three-Dimensional Fractals*, Proceedings of Bridges 2018: Mathematics, Music, Art, Architecture, Education, Culture, Tessellations Publishing, Phoenix, Arizona, 2018, pp. 171–178.
- [Nakamura, 2018] Kento Nakamura, *Iterated Inversion System: An Efficient Algorithm to Visualize Kleinian Groups Based on Inversions*, Master thesis at Meiji University, 2018, Available online at <https://soma-arc.net/paper/masterThesis.pdf> as of May 23, 2020.
- [Nakamura & Ahara, 2020] Kento Nakamura and Kazushi Ahara, *A Flow-based Programming Environment for Geometrical Construction*, Mathematical software – ICMS 2020, Lecture Notes in Computer Science, Volume 12097, Springer, 2020, pp. 426–431.
- [Nakamura et al., 2020] Kento Nakamura, Yoshiaki Araki, and Kazushi Ahara, *Polyhedra with Spherical Faces and Four-Dimensional Kleinian Groups*, preprint, 2020, Available online at <https://soma-arc.net/paper/expMath2020preprint.pdf>.
- [Sakugawa, 2010] Keita Sakugawa, *On Limit Sets of 4-dimensional Kleinian Groups with 3 Generators*, Tokyo Journal of Mathematics, Volume 33, Number 1, 2010, pp. 165–182.
- [Segerman, 2016] Henry Segerman, *Visualizing Mathematics with 3D Printing*, JHU Press, 2016.