A Geometrical Representation and Visualization of Möbius Transformation Groups

Kento Nakamura, and Kazushi Ahara
Meiji University

Abstract

In this paper, we propose a system which enables us to create computer graphics arts originating from Kleinian groups easily and intuitively. Using this system, we can realize various types of Möbius transformations by arranging some circles and lines on the plane. The system uses the efficient rendering algorithm called IIS, introduced in the previous paper by the authors. In the 3D space, making a change circles and lines into spheres and planes, we extend the system to the 3D graphics of Kleinian groups.

Introduction

A Möbius transformation on the complex plane is defined by a linear fractional transformation. Kleinian group theory is one of the fields of mathematics studying Möbius transformations. In the past decade, graphics of the limit set and the orbits of Kleinian groups have emerged as objects of interest because they frequently have a fractal structure. In addition, these beautiful, artistic objects occupy a space between mathematics and art. *Indra's Pearls* [1] is a book describing Kleinian groups from this viewpoint.

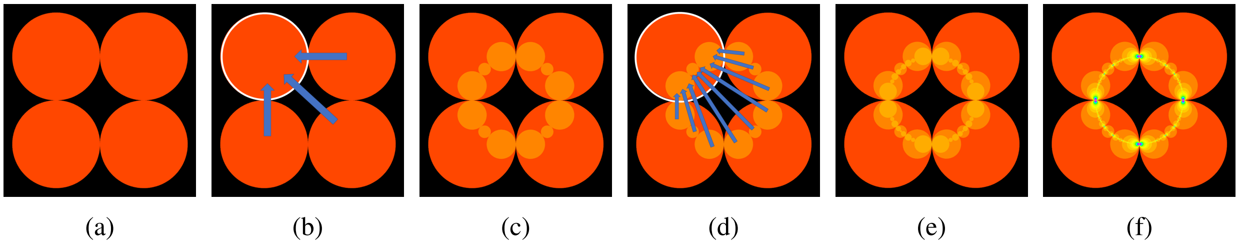


Figure 1: *The process of rendering the orbit of Schottky disks*

As a simple example, we start with *circle inversion fractals* as shown in Figure 1f. Nesting disks are transformed images of disks by iterations of the circle inversions. We call them *the orbit of Schottky disks* and the limits of disks *the limit set*. The process of the transformation is as follows.

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

Generally, to render such figures, we have to calculate coordinates and radii of all of the circles. Here we consider the group generated by the inversions in the Schottky circles. We enumerate elements of the group and transform original circles by these elements. To do so, we have to traverse the Cayley graph of the group with a breadth first search. For more details, see the chapter 4 of *Indra's Pearls* [1]. However, such a method has a fault that if we increase Schottky circles, computational complexity increases exponentially. To solve the problem, we introduced the algorithm called *Iterated Inversion System (IIS)* in the previous paper [2]. The algorithm can be applied to each point on the plane and allows us to get the nesting depth of the disk which contains the point. It is easy to implement IIS in parallel, which enables us to render circle inversion fractals in real time. Also, we can extend the fractals to 3D space using sphere inversions.

Our first example used only circle 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 [3]. Thus, we can apply IIS to visualize fractals combining circle inversion fractals and Möbius transformation groups. Moreover, we can tweak parameters of Möbius transformations easily by arranging some circles and lines on the plane. In this paper, we introduce how to apply IIS to Möbius transformations using inversions. The first author is developing a web application called *Schottky Link* [1]. It enables us to render fractals shown in this paper intuitively. See also the *Shadertoy* page [2] for some sample codes.

Möbius Transformations and Inversions

Möbius transformations are defined in the extended complex plane, $\hat{\mathbb{C}} = \mathbb{C} \cup \{\infty\}$ and expressed as linear fractional transformation $f(z) = \frac{az+b}{cz+d}$, where $a,\ b,\ c,\ d,\ z \in \hat{\mathbb{C}}$. However, it is also known that we can construct them out of a finite composition of inversions. For more details, see the introduction of [3]. Möbius transformations are classified into three types as loxodromic, parabolic, or elliptic. Loxodromic transformations have two fixed points and are conjugate to scaling by complex numbers except for scaling by unit complex numbers. Those whose multiplier is a positive real number are also called hyperbolic transformations. Parabolic transformations have one fixed point and are conjugate to parallel translations. Elliptic transformations have two fixed points and are conjugate to rotations.

An inversion in a circle is defined as $f(z) = \frac{r^2}{\overline{z-c}} + c$, where c and r are center and radius of the circle. Note that an inversion in a circle with infinite radius is the same as a reflection over a line. Also, sphere inversion can be derived from a similar equation, and an inversion in a sphere with infinite radius is the same as a reflection through a plane.

Iterated Inversion System

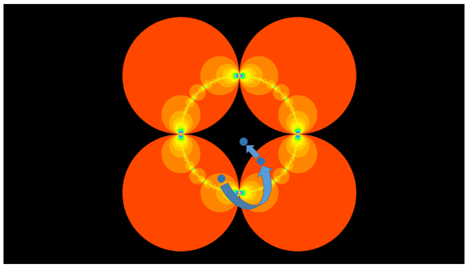
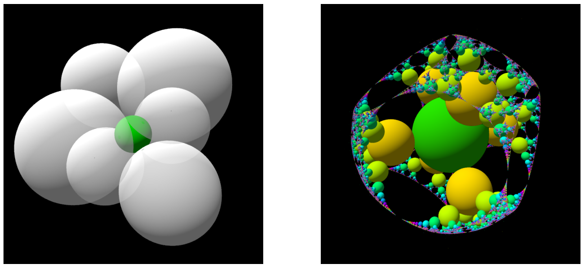


Figure 2: *The orbit of Schottky disks and the orbit of the point generated by IIS*



(a) *Generator* (b) *Orbit*

Figure 3: *The orbit of the base sphere*

[1] Schottky Link: `https://schottky.jp` [2] Shadertoy: `https://www.shadertoy.com/user/soma_arc`

In this section, we will briefly explain IIS. It is applied to each point on the plane and computes the nesting depth of the disk which contains the point. The algorithm is as follows. If the point is contained in one of the Schottky circles, we invert the point in that circle. We continue iterating this process until the transformed point is in *the fundamental domain*. The fundamental domain is the exterior area of all of the Schottky circles. Finally, we color the pixel according to the number of iterations of inversions. Now we consider the circle inversion fractal as shown in Figure 2. In Figure 2, the fundamental domain is the black area. Figure 2 also shows the orbit of the blue point generated by IIS. The point has two iterations of inversions to reach the fundamental domain. Thus, we find that the point is in the second level depth of the nesting disks. Remember, the points closer to the limit set require the larger number of inversions to get into the fundamental domain. Furthermore, a point actually at the limit set never reaches the fundamental domain. So, we have to determine the maximum number of iterations in advance to prevent the algorithm from running indefinitely.

Later, we will introduce generators other than simple inversions. 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 generalized pseudo-code is in Algorithm 1.

Algorithm 1 Iterated Inversion System (IIS)

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

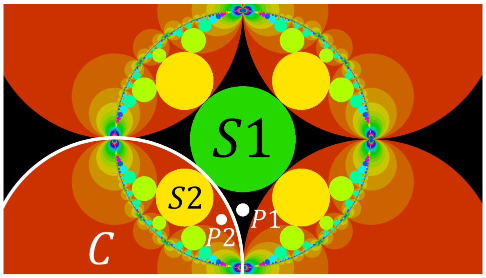


Figure 4: *XY-slice image of Figure 3*

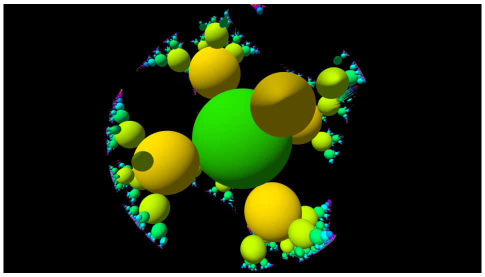


Figure 5: *Figure 3b with artifacts*

We can use the same code to draw sphere inversion fractals. However, it is difficult to render nesting spheres efficiently, and the rendered images do not interest us. Thus, we render the orbit of another sphere. See Figure 3a. We call gray spheres *Schottky spheres* and a green sphere *a base sphere*. We focus on rendering the orbit of the base sphere transformed by compositions of inversions in the Schottky spheres. Figure 3b shows the orbit. We use *ray marching* and *distance estimation* to render such figures efficiently. Ray marching is a kind of ray tracing technique to calculate an intersection of a ray and objects approximately. A ray is something like a vector. First of all, 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 intersection algebraically. On the other hand, in ray marching, 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*. However, in regard to fractal rendering, it is difficult to get an actual distance to its shape. So, we use lower estimated distance as a return value of the distance function. The technique to approximate distance is called distance estimation. For more details about ray marching and fractal rendering, see the blog post [3] by Christensen.

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
        $dr \leftarrow dr*$ (Jacobian of $G($coordinates$)$)
        coordinates $\leftarrow G($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($d$, scalingFactor $*$ (distance(coordinates, $S$.center) $-$ $S$.radius) / (absolute value of $dr$))
  end for
  return $d$
```

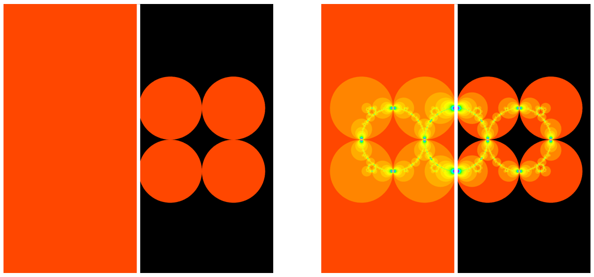
In order to prepare a distance function to render the orbit of base spheres, for the sake of simplicity, we consider a slice image of Figure 3. See Figure 4. This image shows the XY-slice of the orbit of spheres. Orange disks are slices of the orbit of Schottky spheres. Slices of the orbit of the base sphere are colored in the same color as the orbit shown in Figure 3b. C is the white circle, the boundary of the Schottky sphere. $S1$ is the base sphere and the inversion of $S2$ in the circle C. The white point $P1$ is the inversion of $P2$ in the circle C. Now we assume that the tip of the ray is at $P2$. Let's calculate the minimum distance between $P2$ and the orbit of base spheres. The nearest sphere to $P2$ is $S2$. So, we have to calculate the distance between $P2$ and $S2$. We call the distance d. However, we do not know the center and radius of $S2$. So, we calculate d from a distance between $S1$ and $P1$. Inversions in spheres and Möbius transformations do not preserve Euclidean distance. Thus we use the Jacobian (sometimes referred to as the Jacobian determinant) to estimate the distance. The equation to calculate the Jacobian of an inversion is as follows. Let S be the center of the Schottky sphere, let R be its radius, and let P be a point before applying the inversion in S. $Jacobian = R^2/distance(P,\ S)^2$. Note that the Jacobian of the inversions in a sphere with infinite radius is 1. We accumulate the Jacobian of inversions by multiplying the Jacobian for every inversion. Finally, we

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

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 $d \geq distance(P1,\ S1)/Jacobian$. The formula gives a lower bound for spheres. For more details on the derivation of this estimation formula, see the blog post [4] by Inigo Quilez.

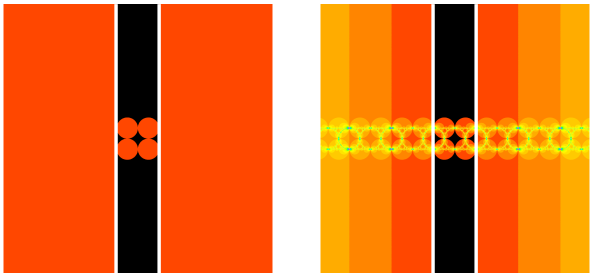
We have one more thing to consider because the above calculation is a rough estimate. If a given point is in the outer area of the limit set, the ray can pass through the objects unintentionally. This is because the point is inverted to the outer fundamental domain, and the distance function returns an unintentionally large distance. This causes artifacts as shown in Figure 5. The fore part of the object is not rendered. In order to avoid this problem, we shrink the length of estimated distance. It increases the number of steps of ray marching, but we can eventually obtain the intersection of the ray and the spheres. The scaling factor is determined experimentally according to the size of the spheres. The generalized pseudo-code for a distance function is in Algorithm 2.

2D Generators



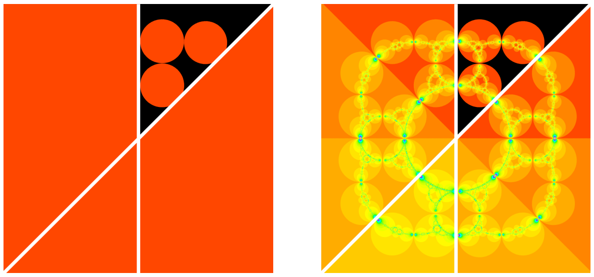
(a) *Generator* (b) *Orbit*

Figure 6: *Inversion in the circle with infinite radius and four Schottky disks*



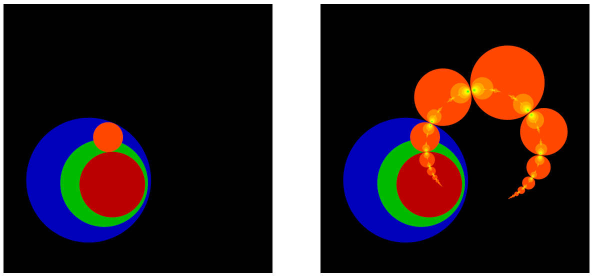
(a) *Generator* (b) *Orbit*

Figure 7: *Parallel translation generator and four Schottky disks*



(a) *Generator* (b) *Orbit*

Figure 8: *Rotation generator and three Schottky disks*

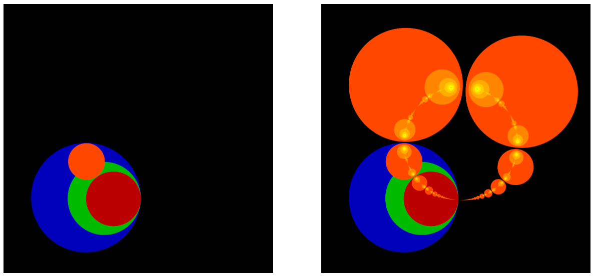


(a) *Generator* (b) *Orbit*

Figure 9: *Hyperbolic generator and a Schottky disk*

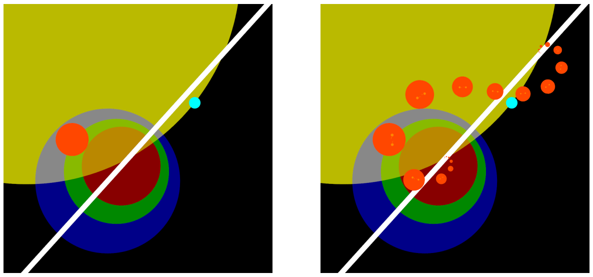
Inversion in a Circle with Infinite Radius. An inversion in a circle with infinite radius is treated as a reflection over a line. See Figure 6a. The four Schottky disks are lying on the right side, and there is the orange region on the left side. The region is a disk with infinite radius, that is, a half plane. Its boundary is colored with a white line. The orbit of the group is shown in Figure 6b. As we can see, it is generated by the reflection over the white line.

[4] Inigo Quilez, distance estimation to implicits: `http://www.iquilezles.org/www/articles/distance/distance.htm`



(a) *Generator* (b) *Orbit*

Figure 10: *The orbit of Parabolic generator and a Schottky disk*



(a) *Generator* (b) *Orbit*

Figure 11: *Loxodromic generator and a Schottky disk*

Parallel Translation. A composition of reflections over two parallel lines facing each other generates a parallel translation. See Figure 7a. There are two orange half planes on the right and left sides. They are the disks with infinite radius. The orbit is shown in Figure 7b. Parallel translation is a parabolic transformation.

Rotation. A composition of reflections over two crossing lines generates a rotation. The generator is shown in Figure 8a. The disks with infinite radius are crossing. The boundary of the disks are colored with white, and the rotation axis is crossing point of two white lines. The orbit is shown in Figure 8b. It has a rotation symmetry. Rotation is an elliptic transformation.

Composition of Two Circles. Next, we use a composition of inversions in two circles. See Figure 9a. There are one Schottky disk and three regions colored with red, green, and blue. We call the boundary of red disk $C1$, the outer circle of green region $C2$, and the outer circle of blue region $C1'$, and let $C1'$ be the inversion of $C1$ in $C2$. The generator is composed of $C1$, $C2$, and $C1'$. While $C1$ and $C2$ have no intersection, the composition of inversions in $C1$ and $C2$ represents hyperbolic transformations [5]. The orbit is shown in Figure 9b. 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_{C1} represents an inversion in $C1$.

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

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. Then, we displace $C1$. When $C1$ and $C2$ are kissing as shown in Figure 10a, this generator becomes a parabolic transformation [6]. The fixed points overlap each other, and the orbit converges to the point as shown in Figure 10b.

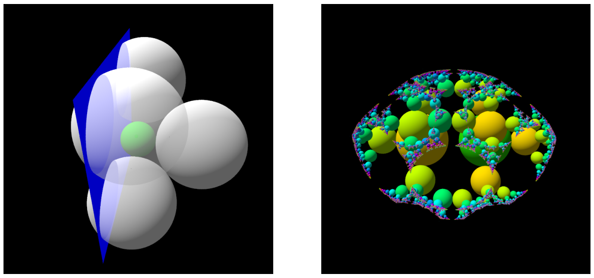
Loxodromic. We can twist the orbit by adding another two inversions. See Figure 11a. The yellow disk and the white line are added to the hyperbolic generator. The white line is a line with two centers of $C1$ and $C2$. We call the line L, the boundary of the yellow disk $C3$, and light blue point P. P is a user-defined control point, and the circle $C3$ is determined by three points, one is the point P, and the others are inversions of P in $C1$ and $C2$. L and $C3$ are perpendicular to $C1$ and $C2$. A composition of the reflection over L and the inversion in $C3$ represents a rotation. Thus, the orbit of the group is twisted as shown in Figure 11b, and this is a loxodromic transformation [7]. The map G is as follows.

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

[5] https://www.shadertoy.com/view/MsScWW

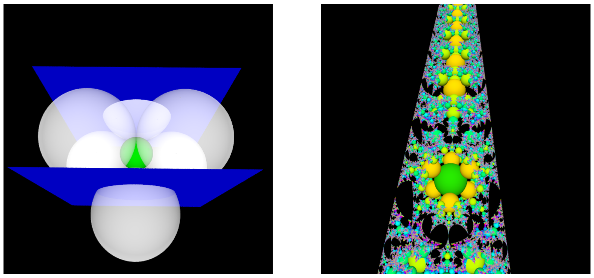
[6] https://www.shadertoy.com/view/XsBcDD

[7] https://www.shadertoy.com/view/lsSyDW



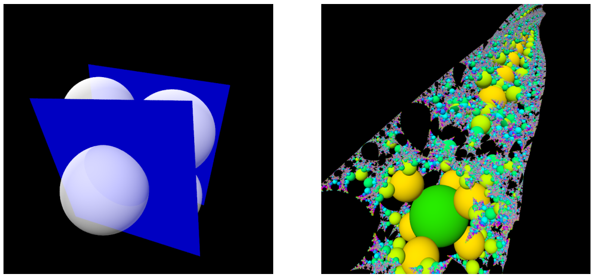
(a) *Generator* (b) *Orbit*

Figure 12: *Inversion in the sphere with infinite radius, four Schottky spheres, and a base sphere*



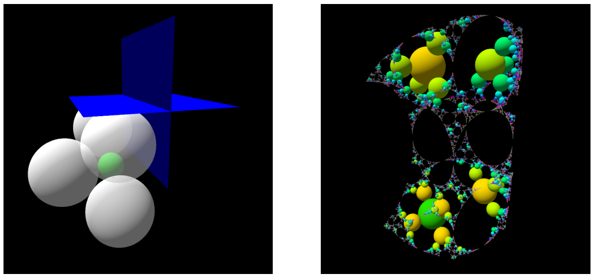
(a) *Generator* (b) *Orbit*

Figure 13: *Parallel translation generator, six Schottky spheres and a base sphere*



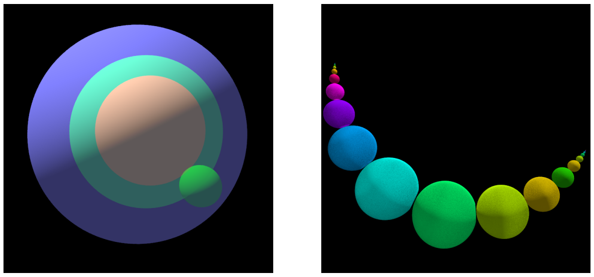
(a) *Generator* (b) *Orbit*

Figure 14: *Compound parabolic generator, six Schottky spheres and a base sphere*



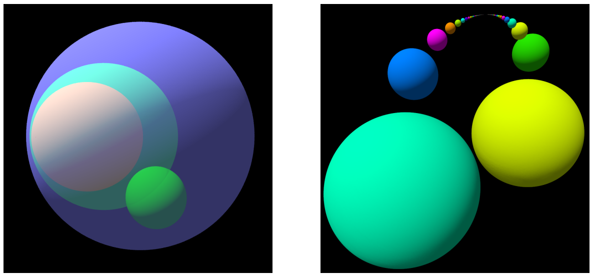
(a) *Generator* (b) *Orbit*

Figure 15: *Rotation generator, four Schottky spheres, and a base sphere*



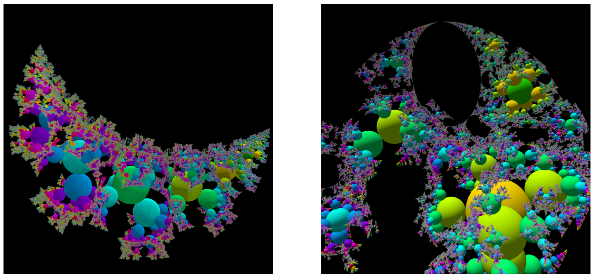
(a) *Generator* (b) *Orbit*

Figure 16: *Hyperbolic generator and a base sphere*



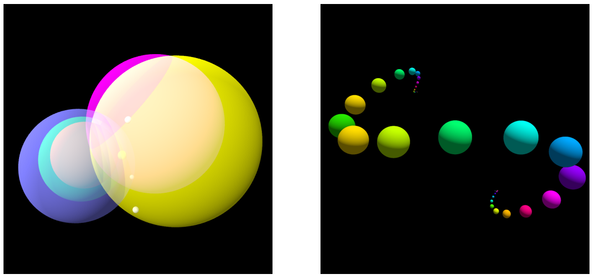
(a) *Generator* (b) *Orbit*

Figure 17: *Parabolic generator and a base sphere*



(a) *Hyperbolic* (b) *Parabolic*

Figure 18: *The orbit generated by a composition of two spheres and six Schottky spheres*



(a) *Generator* (b) *Orbit*

Figure 19: *Compound loxodromic generator and a base sphere*

3D 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 12a. There are four Schottky spheres, one 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 12b. The resulting orbit has a reflection symmetry.

Parallel Translation [8]**.** A composition of inversions in two parallel planes represents a parallel translation along a normal vector of the planes. See Figure 13. This is a parabolic transformation. Moreover, in 3D, we can add a twist to the orbit. See Figure 14. 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 15.

Composition of Two Spheres [9]**.** We compose generators using two spheres. See Figure 16a. We call the light red sphere $S1$, the light green sphere $S2$, and the blue sphere $S1'$. The map G is as follows.

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

While $S1$ and $S2$ have no intersection, the generator is hyperbolic transformation. The orbit of the base sphere is shown in Figure 16b. When $S1$ and $S2$ come in contact with each other at one point as shown in Figure 17a, it becomes a parabolic transformation. The orbit of spheres touches at the fixed point. The orbit is shown in Figure 17b. Also, Figure 18 shows the example of the more complicated orbit of spheres generated by adding six Schottky spheres to the group shown in Figure 16 and Figure 17.

Compound Loxodromic [10]**.** Finally, we add two spheres perpendicular to $S1$ and $S2$. See Figure 19a. We call pink sphere $S3$ and yellow sphere $S4$, and there are three user-defined 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 $S2$. The spheres and the map G are as follows.

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

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

The composition of inversions in $S3$ and $S4$ represents rotation. The twisted orbit shown in Figure 19b is analogous to the loxodromic transformations in 2D. Therefore, we call this generator *compound loxodromic*.

References

[1] David Mumford, Caroline Series, David Wright. *Indra's Pearls*. Cambridge University Press, 2002.

[2] Kento Nakamura and Kazushi Ahara. A New Algorithm for Rendering Kissing Schottky Groups. In *Proceedings of Bridges 2016: Mathematics, Music, Art, Architecture, Education, Culture*, Pages 367-370. Tessellations Publishing, 2016. `http://archive.bridgesmathart.org/2016/bridges2016-367.html`

[3] Iwaniec, Tadeusz and Martin, Gaven. The Liouville theorem. *Analysis and topology*, Pages 339-361. World Scientific Publishing, 1998.

[8] `https://www.shadertoy.com/view/lsjyzK`

[9] `https://www.shadertoy.com/view/ldByDW`

[10] `https://www.shadertoy.com/view/MdjyRV`