

A New Algorithm for Rendering Kissing Schottky Groups

Kento Nakamura, and Kazushi Ahara
Meiji University

Abstract

In this paper, we present an algorithm for drawing images of Schottky groups. This algorithm is called an Iterated Inversion System (IIS). It is easy to parallelize and renders images fast. It can be used to render 2-dimensional kissing Schottky groups and also 3-dimensional kissing Schottky groups.

Introduction

A Schottky group is a kind of Kleinian group. They are Mobius transformation groups which are generated by Schottky transformations. Schottky transformations are composed of a pair of circles or spheres. (See Indra's Pearls [1].) Conventionally, to visualize Schottky groups we traversed Cayley graph composed of its generators. However, it takes too much time if the number of generators is large or we require high quality images. We present a new algorithm which allows us to draw images of kissing Schottky groups by pixel. It is easy to parallelize and also to extend it to draw 3-dimensional kissing Schottky groups. There already have been similar work using a spherahedron approach [2] by Knighty [3]. Knighty has succeeded in developing a fast algorithm to draw the limit sets of Kleinian groups using inversions about spheres. We haven't succeeded in drawing more complex figures than those of Knighty. However, we present an original algorithm using the mathematical character of Schottky groups.

Traditional Ways

In this paper, we call circles (spheres) which compose the groups *Schottky circles (spheres)*. Generally, when we visualize Schottky groups, we calculate the limit set or all of orbits of Schottky circles transformed by elements of the group. The limit set of the group consists of the limit points of the orbit of the circles. The limit set corresponds to the limit ends of the Cayley graph. In order to traverse Cayley graph we usually use Breadth First Search for the orbit of the circles and Depth First Search for the limit set. For more details refer Indra's Pearls [1]. We can use it for 3-dimensional Schottky groups too. However, these search methods have some faults. First, if we increase Schottky circles, computational complexity increases exponentially. Secondly, any algorithm for traversing the Cayley graph is difficult to parallelize, so it is hard to make it quick. Thirdly, it is difficult to draw a partial image of the limit set, especially the neighborhood of a fix point because the speed of convergence is slow.

The Algorithm

We assume we have a kissing Schottky group. We transform the Schottky circles by all elements in the kissing Schottky group. The transformed figures are circles and we try to draw an image of all of such circles. The image in Figure 1 is an example of such figure. In Figure 1, four circles surrounding the black areas are the Schottky circles. Because of a mathematical reason, this figure is the same as a figure of circles we get by transforming the Schottky circles by all of the compositions of the inversions of the Schottky circles. We call this figure *the orbit of Schottky circles*. Suppose that there are an even number of disjoint (or

externally tangent) circles in the plane. These are Schottky circles. Our new algorithm is simple. For each point in the plane, if the point is contained in any Schottky circle, we invert the point about the circle. We continue iterating this process until the transformed point is in the exterior area of the Schottky circle. We call this area *the fundamental domain*. We call a point in this domain *an exterior point*. At the end we color the pixel according to the number of iteration of inversion. In this way, the orbit of the Schottky circles are displayed in the time of multi-linear order of the number of iteration, and the number of the Schottky circles. A pseudocode of this algorithm is in Algorithm 1.

Algorithm 1 Iterated Inversion System (IIS)

Require: countInversion = 0 and coordinates = position determined by pixel

```

for  $i = 0$  to MAX_INVERSION do
  loopEnd  $\leftarrow$  true
  for each circle C in Schottky circles do
    if coordinates are inside C then
      coordinates  $\leftarrow$  inversion image of coordinates by C
      INCREMENT countInversion
      loopEnd  $\leftarrow$  false
      BREAK inner for
    end if
  end for
  if loopEnd then
    BREAK for
  end if
end for
if countInversion > 0 then
  COLOR pixel by countInversion
end if

```

We can also draw only the inner part of the limit set. The limit set of a kissing Schottky group is homeomorphic to a circle, and the limit set divides the plane into two parts, a finite inner area and an infinite outer area. It is known that the inversion of a Schottky circle preserves the interior/exterior part of the limit set. We can divide the fundamental domain into the inner and outer two parts. Thus if we put a color only for a pixel whose exterior point is contained in the inner fundamental domain, we can draw the inner part of the limit set. See Figure 2. Each calculation is atomic and we can parallelize this algorithm. To draw these images, we use a Fragment Shader in the OpenGL Shading Language (GLSL).

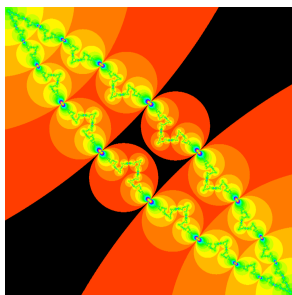


Figure 1 : *The kissing Schottky group composed by four circles.*

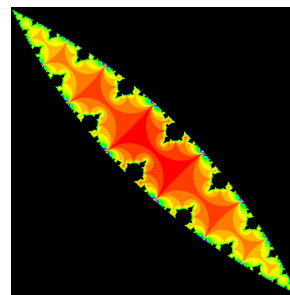


Figure 2 : *Inner part of the limit set of the kissing Schottky group*

3D Extension

We can extend the algorithm to draw images of 3-dimensional kissing Schottky groups. In order to render 3D objects we use ray marching, a kind of ray tracing. Ray marching requires the distance function, which returns a distance between a tip point of a ray and the 3D object. For more details about ray marching, refer the article [4] by Christensen. In our case, the distance function is given in Algorithm 2. This algorithm is written in a similar way to Algorithm 1. We consider some fixed Schottky spheres. If a tip point of ray is inside of any of the Schottky spheres, we invert the point about the sphere. We continue iterating this process until the transformed point is in the exterior area of the Schottky spheres. We accumulate the Jacobian of inversions to estimate the local reduced scale. At the end of Algorithm 2, we fix a sphere in the center of the outer area of all Schottky spheres and, we return a scaled distance between the point and the sphere. In Figure 3, we set 6 Schottky spheres at the position of the vertexes of an octahedron and draw an image of the orbit of a sphere. In Algorithm 2, we need to fix a constant *factor* to draw a proper image. In this case we find the constant is less than 0.08. In the same situation if we get the radius of the center sphere larger, we have the image of the limit set on a sphere. See Figure 4. Figure 5 shows a 3D kissing Schottky group generated by 8 spheres. In Figure 6[5], we can see the demonstration of this algorithm by the first author.

Algorithm 2 Distance function

Require: countInversion = 0 , $dr = 1.0$ and coordinates = position of the ray

```

for  $i = 0$  to MAX_INVERSION do
  loopEnd  $\leftarrow$  true
  for each sphere S in Schottky spheres do
    if coordinates are inside S then
      coordinates  $\leftarrow$  inversion image of coordinates by S
      INCREMENT countInversion
       $dr \leftarrow dr * (\text{Jacobian of inversion of the sphere})$ 
      loopEnd  $\leftarrow$  false
      BREAK inner for
    end if
  end for
  if loopEnd then
    BREAK for
  end if
end for
return factor * (LENGTH of coordinates – rad) / (absolute value of  $dr$ )

```

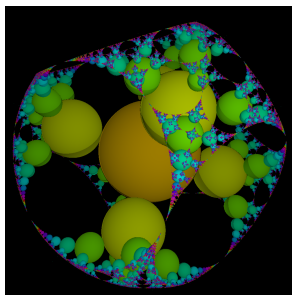


Figure 3: Orbit of spheres of the 3D kissing Schottky group

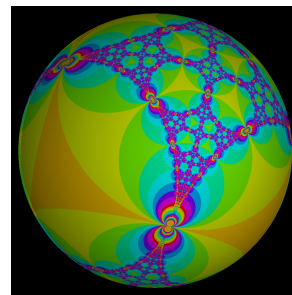


Figure 4: The Limit set of the 3D kissing Schottky group

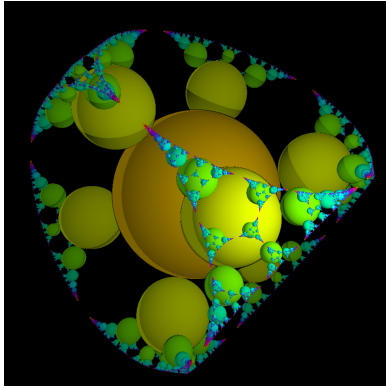


Figure 5: *Orbit of spheres of the 3D kissing Schottky group by 8 spheres*

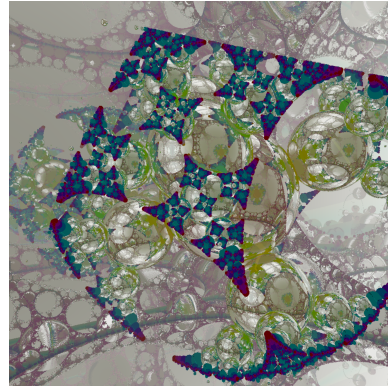


Figure 6: *Indra's Bubbles*

Summary and Future work

We presented an algorithm for rendering Schottky groups. For complete implementation of the algorithm by GLSL, see the repository [6]. This runs in the time of multi-linear order of the number of iteration, and the number of the Schottky circles/spheres. We present the possibility of a new kind of fractal art using Kleinian groups. This algorithm may be extended to other Kleinian groups. Our goal is to develop a mathematical theory and an algorithm for fast rendering.

References

- [1] David Mumford, Caroline Series, David Wright, *Indra's Pearls* (2002), Cambridge University Press.
- [2] Kazushi Ahara, Yoshiaki Araki, *Sphairahedral Approach to Parameterize Visible Three Dimensional Quasi-Fuchsian Fractals* (2003), http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?tp=&arnumber=1214471&isnumber=27295, IEEE Computer Graphics International
- [3] Knighty, Another 3D Kleinian, <http://www.fractalforums.com/ifs-iterated-function-systems/another-3d-kleinian/> (as of Jan. 10 2016)
- [4] Mikael H Christensen, Distance Estimated 3D Fractals (Part I), <http://blog.hvidtfeldts.net/index.php/2011/06/distance-estimated-3d-fractals-part-i/> (as of Jan. 10 2016)
- [5] Kento Nakamura as soma_arc, *Indra's Bubbles* <https://www.shadertoy.com/view/XsGGWG> (as of March 4 2016)
- [6] Kento Nakamura as soma-arc, Iterated Inversion System in GitHub, <https://github.com/soma-arc/IteratedInversionSystem> (as of March 4 2016)