Abstract

Hyperbolic geometry is much less well known than Euclidean geometry or spherical geometry. One important reason for this is that there is no smooth, distance-preserving embedding of hyperbolic geometry in Euclidean 3-space, as there is for the Euclidean and spherical geometry. Thus several distance-distorting models of hyperbolic geometry have been used in order to understand it. These models are isomorphic, but complex formulas and projections are required to convert from one model to another.

The goal of this thesis is to demonstrate that standard viewing projections of the Weierstrass model can be used to obtain several other models. That is, simply by viewing the Weierstrass model from special points, one can recover the other models. But, the visualization software is completely general, allowing the user to navigate around the Weierstrass model to any viewpoint. Thus the user can gain more insight as to what hyperbolic geometry "really looks like" than with static models.

Acknowledgements

I am grateful to my advisor Dr. Douglas Dunham for providing me an opportunity to work with him and the valuable guidance that he provided. I also thank Dr. Gary Shute and Dr. Robert McFarland for their co-operation.

I am indebted to my family and friends, without whose support nothing would have been possible.

Contents

List of Figures

Chapter 1

Introduction

Hyperbolic geometry is much less well known than Euclidean geometry or spherical geometry. One important reason for this is that there is no smooth, distance-preserving embedding of hyperbolic geometry in Euclidean 3-space, as there is for Euclidean and spherical geometry. Thus several distance-distorting models of hyperbolic geometry have been used to represent it. These models are isomorphic, but complex formulas and projections are required to convert from one model to another.

The goal of this thesis is to demonstrate that standard viewing projections of the Weierstrass model can be used to obtain several other models. That is, simply by viewing the Weierstrass model from special points, one can see the other models. But, the visualization software is completely general, allowing the user to navigate around the Weierstrass model to any viewpoint. Thus the user can gain more insight into what hyperbolic geometry "really looks like" than with static models.

In the next sections, we will review hyperbolic geometry and its models, 3D viewing, and Java 3D. Then we will show how common models of hyperbolic geometry can be recovered by viewing the Weierstrass model from special points. Next we show samples of hyperbolic art obtained by viewing from these special points. Finally, we indicate directions for future work.

Chapter 2

Hyperbolic Geometry and its Models

2.1 Euclidean Geometry

The geometry that we are all familiar with is called Euclidean geometry. Euclid in his series of books called *The Elements* defined certain postulates which form the pillars of modern day geometry. In these books, he listed five *postulates* [10] , known as axioms in modern mathematical terminology. They are:

- **Postulate 1:** It is possible to draw one and *only* one straight line connecting two distinct points.
- **Postulate 2:** From each end of a finite straight line it is possible, to produce it continuously in a straight line by an amount greater than any assigned length.
- **Postulate 3:** It is possible to describe one and only one circle with any center

and radius.

- **Postulate 4:** All right angles are congruent to each other.

- **Postulate 5:** If a straight line falling on two straight lines make the interior angles on the same side less than two right angles, the two straight lines, if produced indefinitely, meet on that side on which the angles are less than two right angles.

This is equivalent to John Playfair's Postulate:

For every line ℓ and for every point P that does not like on ℓ, there exists a unique line m through P that is parallel to ℓ.

Hilbert proved the consistency of these postulates.

Some other theorems that we are familiar with, regarding Euclidean geometry are:

The sum of all the angles in a triangle is *always* equal to 180°.

The sum of all the interior angles of a quadrilateral is equal to 360°.

2.2 Non-Euclidean Geometry

Euclid's Fifth Postulate had always been thought of by mathematicians as special. Mathematicians never doubted its truth, but always thought of it as a theorem, which could be proved from the other four.

By the latter half of the 18th Century, this problem of proving the Fifth postulate had become really famous and many mathematicians had attempted to prove it. It was only a matter of time before the difficulty of the problem would cause some to conclude that this problem was unsolvable. It did not in any way mean that

Postulate 5 was unprovable. But, this unproven idea made the discovery of non-Euclidean geometry inevitable. The logical transition behind it was that since neutral geometry, the goemetry of Euclid's first four axioms, itself did not imply Postulate 5, there must be a new geometry different from Euclid's.

It has been observed many times in the history of science and mathematics, that when many people are working on the same problem, and when the communication between them is infrequent, it leads to multiple independent discoveries. In this sense, it seems that non-Euclidean geometry was discovered about four times in a span of twenty years. Apparently, Carl Friedrich Gauss was the first to discover non-Euclidean geometry. But, he did not publish any papers related to his discoveries. He worked on this new geometry for many years and discovered many theorems.

In the meanwhile, he received a letter from Ferdinand Schweikart which indicated that Schweikart himself had discovered non-Euclidean geometry and had basically reached similar conclusions and results as Gauss.

None of them published any papers though, and hence, when Janós Bolyai published his work on non-Euclidean geometry in the *Appendix* of his father's book, it established the field of non-Euclidean geometry.

Bolyai though, was not the first person to have published a paper on non-Euclidean geometry. A Russian mathematics professor, Nocolai Lobachevsky had already published a paper on the topic. But, since the paper was published in Russian, it was not well known in the European mathematical circles. Its translations into French and German later on reaffirmed Lobachevsky's work and discoveries in the field of non-Euclidean geometry.

Non-Euclidean geometry is technically any geometry which is not Euclidean. One of the most useful non-Euclidean geometries is spherical or elliptical geometry, which describe the geometry of the sphere. Hyperbolic geometry, the geometry of hyperbolic space, is another non-Euclidean geometry. This is the geometry discovered by Bolyai, Gauss, Lobachevsky and Schweikart [10].

2.3 Hyperbolic Geometry

Hyperbolic geometry is defined as

The geometry you obtain by assuming all the axioms for neutral geometry (geometry without a parallel postulate) and replacing Hilbert's parallel postulate by its negation, which we shall call the "hyperbolic axiom." [7].

Basically, in hyperbolic geometry all of Euclid's postulates hold, other than Postulate 5. Instead of the fifth postulate, an axiom called the *Hyperbolic Axiom* is used.

According to the Hyperbolic Axiom

If there exists a line ℓ and a point P not on line ℓ, there are atleast two distinct lines parallel to ℓ passing through P.

This postulate in turn, is used to prove and obtain many other useful results including:

- There exists a triangle whose angle sum is less than $180°$.
- All quadrilaterals have angle sum less than $360°$.
- If two triangles are similar, they are congruent, that is, if the angles of the triangle are equal, so are the sides.

To prove that any geometry is consistent, we have to have a model of it. A model is an interpretation of the primitive terms under which the axioms become true statements. Here interpretation does not allude to the "understanding of meaning", but in a more basic sense of "giving of meaning".

Unlike Euclidean geometry models that are infinite, some models of hyperbolic space can represent hyperbolic objects in a finite portion of Euclidean 2-space. The Beltrami-Klein model and the Poincaré circle model are examples of such finite models, while the Poincaré's Upper Half Plane model, the Weierstrass model and the Minkowski model are infinite structures embedded in Euclidean space. The finite models have boundaries, which play an important role in the definition of parallel lines in hyperbolic space.

2.4 The Beltrami-Klein Model

The Beltrami-Klein Model is one of the most widely known and used models. It is used to visualize hyperbolic objects such as points and lines [7]. We shall first obtain an interpretation of terms like "point", "lines", "lies on" and "between". The Beltrami-Klein model is represented by the interior of a circle, a point in which represents a point in the hyperbolic plane. We know that a chord of a circle is a line joining any two points of the circle. The concept of an *open chord* is introduced in this model. An open chord is the segment between the two points, omitting its endpoints. In the Klein model, an open chord represents a hyperbolic line. The relation "lies on" has the regular interpretation when we say that P lies on the open chord AB, it means that it lies on the the Euclidean Line AB. The hyperbolic interpretation of "between" is the same as its usual Euclidean interpretation.

Consider the following figure from [7], where we have a line ℓ and a point P outside it. From the Hyperbolic axiom, we have lines m and n parallel to the open chord ℓ.

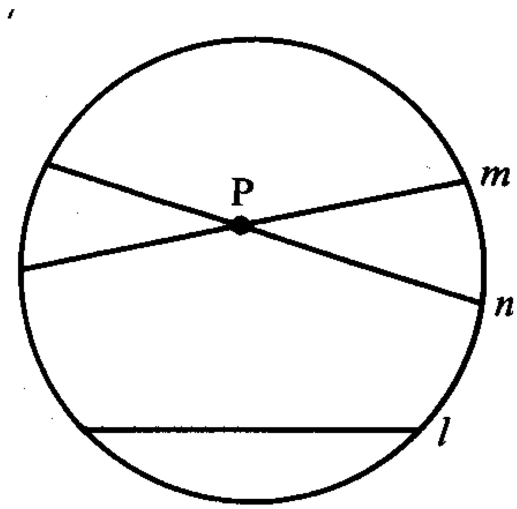

Figure 2.1: Lines m and n parallel to a given line ℓ

We can say that lines m and n are parallel to ℓ, from the definition of parallel lines. The definition states that the two lines are parallel if they have no points in common. So in Klein's representation two open chords are parallel when they have no point in common. The fact that they will intersect outside the circle is not important since the points outside the circle will lie outside the hyperbolic plane.

2.5 The Poincaré Model

The Poincaré circle model was developed by the French Mathematician Henri Poincaré [7]. This model is also represented by the interior of a circle, a point of which represents a point in the hyperbolic plane. Hyperbolic lines are represented differently in this model. Firstly, all the open chords that form diameters (i.e. pass through the center) represent lines. The other lines are represented by open arcs of circles orthogonal to this circle. Two circles are said to be orthogonal if, at the point of intersection their tangents are perpendicular.

Figure 2.2 shows both the possible representations of lines in the Poincaré model. The diameter as well as the arcs in the interior of the circle can be seen.

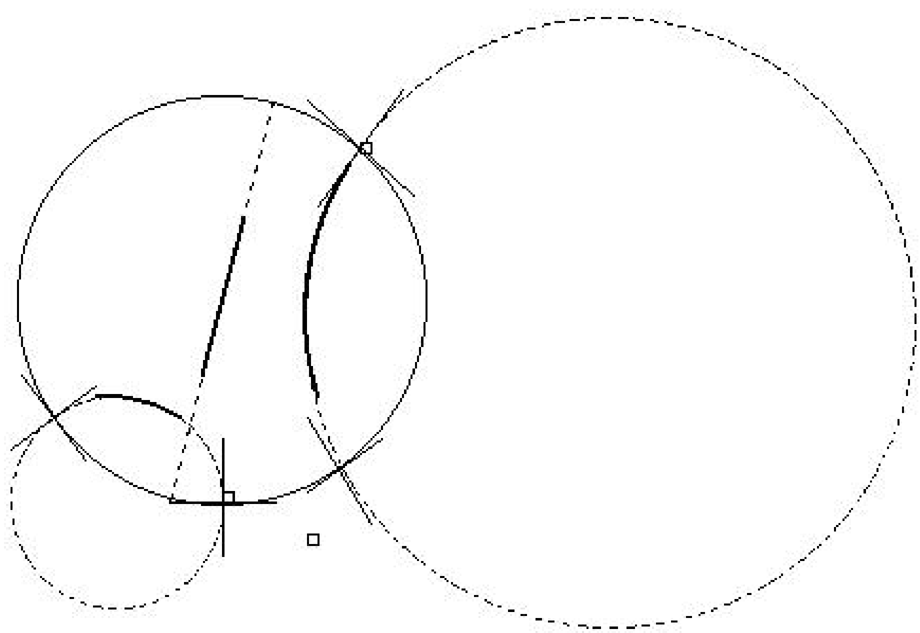

Figure 2.2: Line representations in the Poincaré circle model

So, we will call either an open diameter or an open circular arc a *Poincaré line.* The interpretation of "lies on" and "between", in this model, is same as that of their usual Euclidean interpretation.

Congruence for angles in the Poincaré model has the usual Euclidean meaning. This is an advantage over the Klein model, since angles are represented accurately unlike in the Klein model. If two arcs intersect at a point, the number of degrees measured between the tangents to the two arcs at the point of intersection, is the angle between the two arcs.

Having interpreted all the undefined terms, we can now define the meaning of two lines being parallel. In the Poincaré model, if two lines do not have any point in common, they are parallel to each other.

Figure 2.3 from [7] contains lines ℓ,m and n. Line ℓ is a diameter and hence is a straight line, whereas the lines m and n are circular arcs perpendicular to the bounding circle D.

Lines m and n are not parallel since they share a common point. Line ℓ and line n are divergently parallel, and line ℓ and line m are asymptotically parallel.

2.6 The Isomorphism between the Beltrami-Klein and the Poincaré model

There exists an isomorphism between the Klein and the Poincaré model. This means that a one-to-one correspondence can be set up between the *points* and *lines* in one model and those in the other, which preserves incidence, congruence, and betweenness.

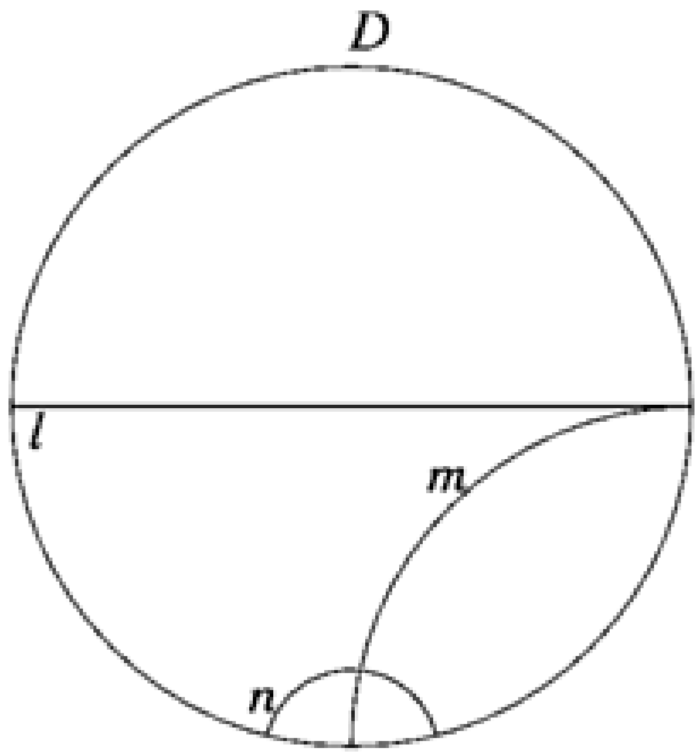

Figure 2.3: Some lines in the Poincaré model and their relationship

The formulae for converting from the Klein model to the Poincaré model are:

$$u = \frac{x}{1 + \sqrt{1 - x^2 - y^2}} \tag{2.1}$$

$$v = \frac{y}{1 + \sqrt{1 - x^2 - y^2}} \tag{2.2}$$

where the Klein coordinates (x,y) are converted into the Poincaré coordinates (u,v).

The formulae for converting from the Poincaré model to the Klein model are:

$$x = \frac{2u}{1 + u^2 + v^2} \tag{2.3}$$

$$y = \frac{2v}{1 + u^2 + v^2} \tag{2.4}$$

where Poincaré coordinates (u,v) are converted into the Klein coordinates (x,y).

The mapping from the Klein model to the Poincaré model can be described as a succession of two projections. Another mapping between the two models can be found by the use of Stereographic projection.

Consider a sphere of exactly the same radius placed on the Klein's model. First, orthogonally project all the lines and points upwards onto the lower hemisphere of the sphere. By this projection, the lines on the Klein model become circular arcs on the lower hemisphere of the sphere perpendicular to the equator. Secondly, project all the arcs stereographically from the north pole of the sphere back onto the original plane. The equator of the sphere will project onto a circle twice as large as that of the Klein model and the arcs on the lower hemisphere will form arcs in this larger circle. By these projections, the chords of the Klein model are mapped one-to-one onto the diameters and arcs of the Poincaré model. This is another way of establishing the isomorphism between these two models. Figure 2.4 below, from [7], shows a diagrammatic representation of the isomorphism and the process of stereographic projection.

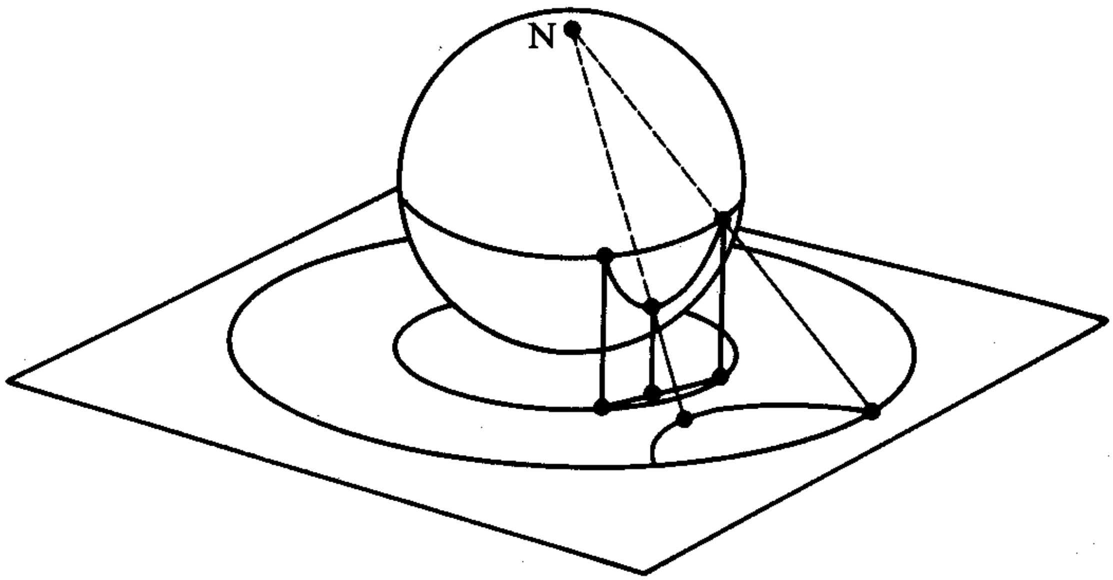

Figure 2.4: An Isomorphism between the Klein and the Poincaré model

2.7 The Upper Half-Plane Model

The upper half-plane model is another model of hyperbolic geometry developed by Henri Poincaré [7]. This model consists of all those points in the upper hall of the xy-plane, excluding those point on the x axis, which is the boundary for this model. The "lines" are represented in two ways: a) by semicircles in the upper half-plane whose centers lie on the x axis. b) as upward rays perpendicular to the x axis, as can be seen in Figure 2.5.

Here, lines m, n and k are divergently parallel, whereas lines ℓ and n are not parallel, since they have a point in common. Lines ℓ and m are also divergently parallel, whereas line ℓ and k are asymptotically parallel.

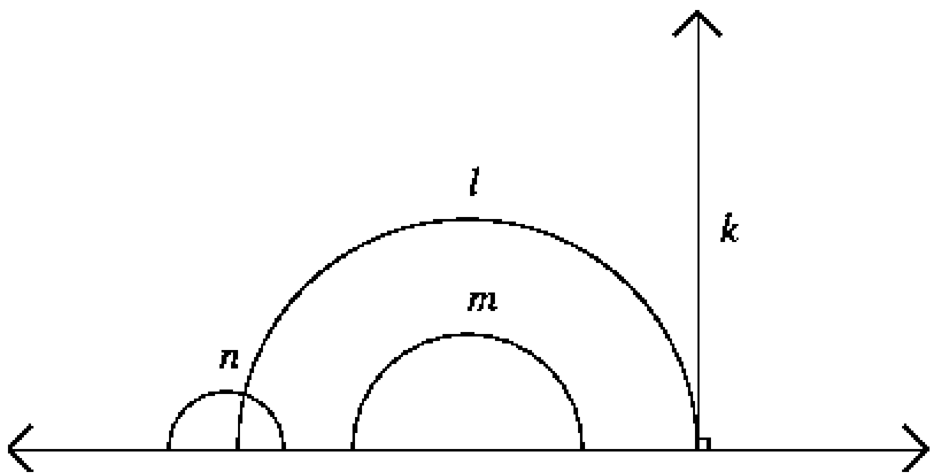

Figure 2.5: The Upper Half-Plane model

The properties of incidence and betweenness have their usual Euclidean meaning and degrees of angles are measured in the Euclidean way.

2.8 The Gans' Model

This model was proposed by David Gans in his paper *A New Model of the Hyperbolic Plane* [6]. We shall refer to this model as the *The Gans' Model.*

This model, unlike the other more famous models, utilizes the entire Euclidean plane rather than some part of it. In this model, the points are points of the Euclidean plane and so the interpretation of points is not different. A "line" in this model is a branch of a hyperbola centered at the origin or a line through the origin (a special case).

The two hyperbolic lines are lines in the Gans' model. The dotted lines are also lines, but are a special case of a hyperbola in which the eccentricity of the hyperbola becomes infinite and the branch approaches the straight line containing the diameter.

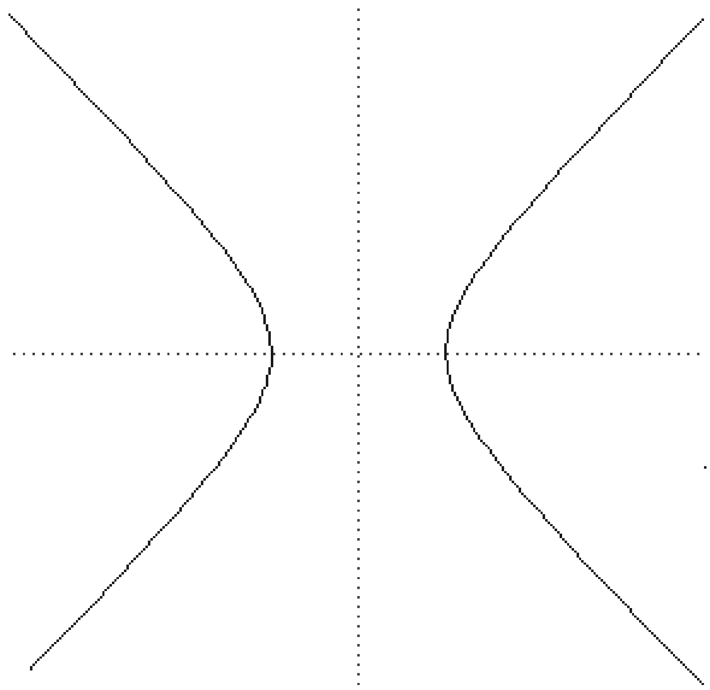

Figure 2.6: The Gans' Model

2.9 The Weierstrass Model

The Weierstrass model is a model which demonstrates the consistency of hyperbolic geometry on the surface of a hyperboloid Euclidean 3-space [4]. It is a lesser known model than the Klein and Poincaré models. It has many properties similar to elliptical or spherical geometry. This model readily generalizes to n dimensions and it introduces an analogue of the Lorentz metric, which is central to the theory of relativity.

Before we define the interpretation for "point", "line", "lies on" and "between", we shall go over some preliminaries.

"Points" in the real 3-dimensional coordinate system are defined as triples of real numbers, such as $X = (x, y, z)$. X can be thought of as not only a point, but also as a *vector*, visualized as an arrow drawn from the origin, O(0,0,0), to X.

Following Faber, we define *hyperbolic product* of two vectors X_1 and X_2 is defined by the following formula

$$< X_1, X_2 >= x_1x_2 + y_1y_2 - z_1z_2$$

We can now understand the Weierstrass model and prove the consistency of hyperbolic geometry. We have to define the interpretation of terms like "point", "line", "lies on" and "between". The surface (with constant curvature k) can be represented by

$$< X, X >= x^2 + y^2 - z^2 = -k^2$$

forms a hyperboloid of two sheets in the Euclidean 3-space. We do not take the lower sheet into account, so that we do not have to deal with antipodal point pairs.

A point X in the Weierstrass model is represented as a point (or vector) such that $< X, X >= -k^2$ and $z > 0$. Any such point will form the upper hyperboloid surface, which we shall denote by H^2.

A hyperbolic "line" is defined as the intersection of the surface H^2 with a plane passing through the origin. We know from Euclidean geometry that this defines the two branches of a hyperbola. The upper branch is shown in Figure 2.7, whereas the lower branch is on the discarded lower sheet of the hyperboloid.

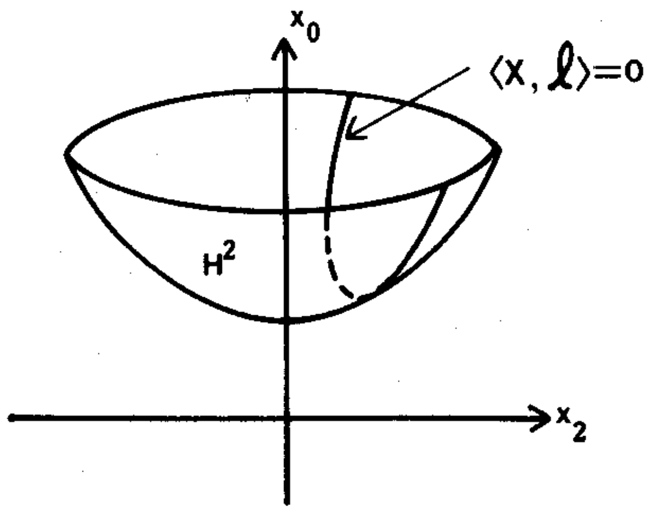

Figure 2.7: A "Line" in the Weierstrass model

Now, a plane passing through the origin would have the equation $< X, \ell >= 0$ where ℓ is a hyperbolic-normal of the plane.

In the Figure 2.9, ℓ is outside H^2 and hence $< \ell, \ell > \; > 0$. But, since any scalar multiple of ℓ determines the plane $< X, \ell >= 0$, we will always choose ℓ to be a point such that $< \ell, \ell >= k^2$, so that ℓ will always be a point on the hyperboloid of one sheet $< X, X >= k^2$ as shown in the Figure 2.8.

Hence, a "line" is a section of the surface H^2 by a plane passing through the origin. It is determined by the equation $< X, \ell >= 0$ or simply by naming its vector ℓ.

A point X *lies on* a line *l*, only if $< X, \ell >= 0$. Two points X and Y, lie on a

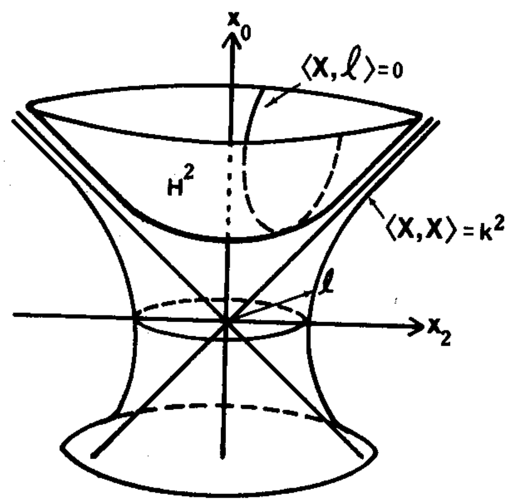

Figure 2.8: $< X, \ell >= 0$ in the Weierstrass model

unique line, since they form a plane with the origin, which intersects the H^2 surface. This intersection is one branch of a hyperbola. If there are three points A, B and C on the line, we will say that C is *between* A and B if, when the branch is traversed in either direction, the points are encountered in the order ACB or BCA.

2.10 Isomorphism between the Weierstrass Model and other models

The Weierstrass model is isomorphic to the Beltrami-Klein model, the Poincaré model and the Gans' model.

The Weierstrass model is related to the Poincaré model, embedded as the unit disk in the xy-plane. This can be observed by stereographically projecting the Weierstrass model onto the xy-plane toward the point (0,0,-1). The projection is given by:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \longrightarrow \frac{1}{(1+z)} \begin{pmatrix} x \\ y \\ 0 \end{pmatrix}$$

The inverse projection from the Poincaré model to the Weierstrass model is given by:

$$\begin{pmatrix} x \\ y \\ 0 \end{pmatrix} \longrightarrow \frac{1}{(1-x^2-y^2)} \begin{pmatrix} 2x \\ 2y \\ 1+x^2+y^2 \end{pmatrix}$$

Similarly, the relation between the Klein model and the Weierstrass model is obtained by projecting the Weierstrass model onto the plane $z = 1$ towards the origin (0,0,0). The Klein model is obtained as a unit disk on the plane $z = 1$. The projection is given by :

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \longrightarrow \begin{pmatrix} x/z \\ y/z \\ 1 \end{pmatrix}$$

The inverse from the Klein model to the Weierstrass model can be given by:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \longrightarrow \begin{pmatrix} \frac{x}{\sqrt{1-x^2-y^2}} \\ \frac{y}{\sqrt{1-x^2-y^2}} \\ \frac{1}{\sqrt{1-x^2-y^2}} \end{pmatrix}$$

The relation between the Gans' model and the Weierstrass model is intriguing. The Weierstrass model when projected orthographically onto the xy-plane results in the Gans' model. The formulae for this projection are:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \longrightarrow \begin{pmatrix} x \\ y \\ 0 \end{pmatrix}$$

The inverse from the Gans' model to the Weierstrass model can be obtained by:

$$\begin{pmatrix} x \\ y \\ 0 \end{pmatrix} \longrightarrow \begin{pmatrix} x \\ y \\ \sqrt{1+x^2+y^2} \end{pmatrix}$$

There is no simple projection that allows us to transform from the Weierstrass model to the upper half-plane model. However, it can be done by a succession of three projections: first project onto the Poincaré model as above, then project verticall(orthogonally) upward onto the unit hemisphere centered at the origin (whose equator is the bounding circle of the Poincaré circle model) and finally project the hemisphere stereographically from the point (-1,0,0) onto the plane $x = 1$ (which is tangent to the hemisphere at (1,0,0).

Chapter 3

3D Viewing

In comparision to two-dimensional viewing, three-dimensional viewing is much more complex. This can be attributed to the fact that there is an additional dimension which needs to be taken into account. In 2D viewing, the object to be viewed has to be clipped into the current view and then transformed to the viewport for display. In 3D, the complexity is further increased because the objects are 3-dimensional and the display is 2-dimensional.

The solution is to project the 3D object onto a 2D plane and then transform it to the 2D display. A *view volume* is specified in 3D. The 3D object is clipped to the view volume and is then projected onto a 2D projection plane.

Projections transform points in a coordinate system of dimension n into points of a coordinate system of dimension less than n [5]. In this case specifically, we need to project 3D objects onto the 2D display after clipping.

There are two types of projections commonly used in computer graphics:

- Perspective projection
- Parallel projection

The distinction is in the relation of the *center of projection* to the Projection Plane. For *parallel projection*, the distance between the center of projection and the projection plane is infinite. This essentially means that the projectors are parallel and the center of projection is at infinity.

For a *perspective projection*, the center of projection is at a finite distance from the projection plane. The projectors meet at the center of projection. A perspective projection whose center moves to a point at infinity becomes a parallel projection. The visual effect of a perspective projection is similar to that of photographics systems and also the human visual system. The surface or edge closest to the viewer seems the biggest, whereas an object surface or edge length decreases as its distance from the viewer increases. Thus, the size of the object varies inversely with its distance from the user. Perspective projection gives a realistic view, but it distorts the actual distances and angles of objects. Most parallel lines in the actual object will not remain parallel and will seem to converge.

Parallel Projection gives a less realistic view because there is no relation between the distance to the user and the size of the object. Parallel lines will always remain parallel and all measurements will be exact. Both the projections preserve the angles only on faces of the object parallel to the projection plane [5].

Chapter 4

Java 3D

4.1 Java 3D: An Introduction

Java 3D is an *application programming interface* (API) developed by Sun. It is used for creating 3D graphical applications and applets. It adds to the core Java language and it gives the user the capability to develop 3-dimensional models using the Java Language. It allows the user to develop 3D scenes faster than other available options for creating 3D scenes, mainly because it simplifies the tasks of building scenes, allowing the developer to disregard non-essential or obscure details if they choose to do so.

The API provides a set of object-oriented interfaces that support a simple, high-level programming model. This enables developers to build, render, and control the behavior of 3D objects and visual environments.

Hardware acceleration is obtained in Java 3D either using the lower level API of DirectX, QuickDraw3D or OpenGL. It uses OpenGL on Unix (and its flavors), QuickDraw3D on the Macintosh Operating system, and DirectX on all Windows versions, thus providing the platform-independence that is expected from a Java API. These API's in turn have optimized implementations that provide support for various graphics hardware subsystems [9].

4.2 Java 3D and VRML

The Virtual Reality Modeling Language (VRML) is a modeling language that allows the developer to create 3D content. It is the International Standard (ISO/IEC 14772) file format for describing interactive 3D multimedia on the Internet. It follows the World file format (.wrl). The source file is dynamically interpreted and the final scene is then generated. It requires a VRML browser which will browse its source files and interpret them. Hence, VRML is bound to its file format, and it is also a higher level modeling language, and hence hardware acceleration cannot be leveraged. Platform-independence is not possible, unless a VRML browser for that platform is available. Browsers for IRIX and Windows are freely available, but for other platforms the browsers are not freely available.

On the other hand, Java3D does have a lower level interface since it is built on DirectX, OpenGL and QuickDraw3D and can leverage lower level control, and so is more scalable and demonstrates a higher level of performance. It does not require any specific file format and is basically an extension of the Java core language, and hence is a part of the Java's "write once, run anywhere" commitment. It provides support for applications such as mechanical CAD, real-time simulation, data visualization, and scientific modelling.

For the problem at hand, the need was for superior rendering, texturing capabilities, viewing, interaction and 3D geometry. Compared to VRML, Java3D provides extensive support for all of the above. It also excels in transforms, lighting, fog, sound playback and reverberation, and gives the programmer total control over the input devices and their use.

Java3D is a programming API, and the model it uses is that people will write programs that cause geometry to appear and do things. It is more program-centric than VRML and allows the user more control to create the 3D objects in the scene. VRML is scene based, and its model is that scripts and the External Authoring Interface (EAI) manipulate objects in the scene [9].

Hence Java 3D was the natural choice for the implementation of the visualization of the 3D Weierstrass model. The problem required a more program-centric option, and hence Java 3D was chosen.

Chapter 5

The Visualization Problem and its Solution

As we have seen in Section 2.9, the conversion from one hyperbolic geometric model into another model can be difficult and often requires complex mathematical formulae. In addition, if we want to visualize a hyperbolic object in two different models, it has to be converted independently from one model to another, and there is no easy way to switch between all these models.

By using standard viewing projections of 3D graphics and using the less well-known Weierstrass model, we can view the well known Poincaré circle model, the Klein model and the not-so-familiar Gans' model. Gans' model is the only model in which the hyperbolic lines are represented by Euclidean hyperbolas. Visualization of the 3-Dimensional Weierstrass model from different viewpoints displays each of these models.

A viewer at the origin looking upward would see the projection of the Weierstrass model onto the Klein model. Similarly, a viewer looking at the Weierstrass model from (0,0,-1) would see the projection onto the Poincaré model. Both of the projections are perspective projections. As regards, Gans' model, it can be observed as the viewer moves to negative infinity along the Z-axis, thus obtaining a parallel projection of the Weierstrass model.

A program was developed which displays the 3-dimensional Weierstrass model. The viewer can make smooth transitions between the views, dynamically showing the transformation of one model into another. Thus, the viewer can interactively change the viewpoint and observe hyperbolic geometry as it would be represented using different models.

This should allow the user to get a better feel of what hyperbolic geometry really "looks like".

Chapter 6

Hyperbolic Geometry and Art

M.C.Escher was the first artist to use hyperbolic geometry to create hyperbolic patterns. There is a distinctly mathematical theme to his artwork. He used the idea of tessellations in the hyperbolic plane.

Escher used the Poincaré model as the basis for his art. His famous Circle Limit patterns use hyperbolic geometry and the Poincaré model in particular. They are: *Circle Limit I* (Black and White fishes), *Circle Limit II* (Red and Black crosses), *Circle Limit III* (Colored Fishes) and *Circle Limit IV* (Heaven and Hell, with bats and angels). Circle Limit I, II, III and IV are shown below [2][3].

The hyperbolic points of this model are the interior points of the fixed circle, which is the boundary of the Poincaré model. The Poincaré model is conformal which means that its representation of angles is the same as Euclidean angles, so that a transformed object has roughly the same shape as the original.

The Poincaré model is better than the Klein model, since it conforms to the Euclidean sense of angles, contrary to the Klein model. It is also observed that in the Klein model, there is a less uniform distribution of the hyperbolic pattern. It is more dense towards its circular boundary and more rarified as we move in towards the center.

The Weierstrass model is the ideal model to be used for generating hyperbolic patterns using computers. Programs to generate hyperbolic patterns are more easily to develop using the Weierstrass model. The main reason for this is that the Weierstrass model represents its points as 3D vectors. They can then be transformed by using 3x3 matrix transformations, which are preferred when we need to carry out transformations in computer graphics.

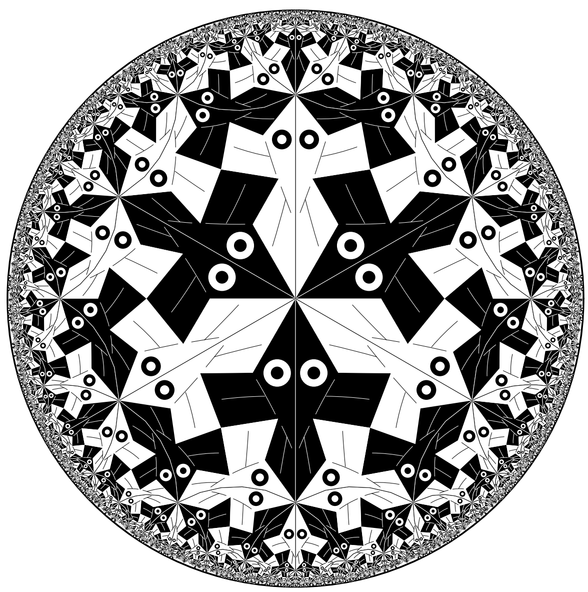

Figure 6.1: Escher's Circle Limit I

Figure 6.2: Escher's Circle Limit II

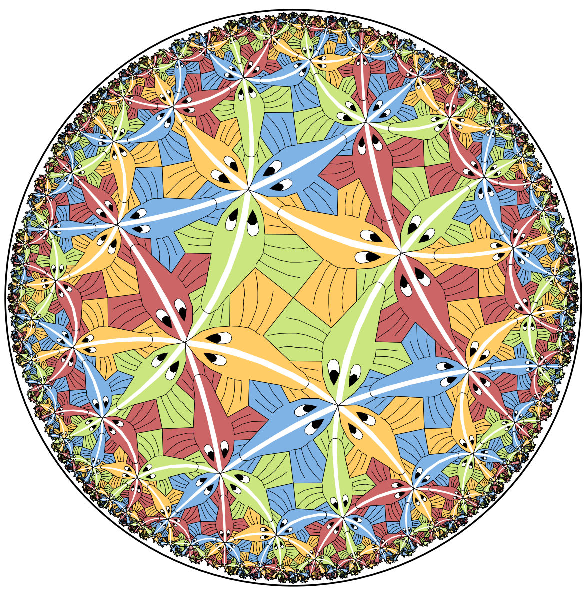

Figure 6.3: Escher's Circle Limit III

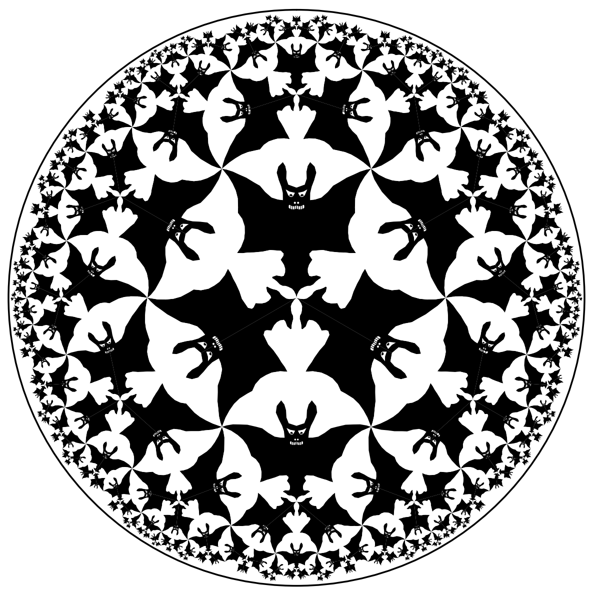

Figure 6.4: Escher's Circle Limit IV

Chapter 7

Results

In this chapter, we shall look at the results obtained by viewing the Weierstrass model from different viewpoints. Escher's Circle Limit I was used to test the standard viewing projections of the 3D Weierstrass model to show views of other models of hyperbolic geometry.

Figure 7.1 shows a side view of the Weierstrass model with the Circle Limit I pattern on it.

Figure 7.2 shows the expected Poincaré model obtained by observing the Weierstrass model from the (0,0,-1) point in the 3D space. Figure 7.3 shows the observed Poincaré model.

Figure 7.4 shows the expected Beltrami-Klein model obtained by observing the Weierstrass model from the origin (0,0,0) in the 3D space. Figure 7.5 shows the observed Klein model.

Figure 7.6 shows the expected Gans' model obtained by observing the Weierstrass from a large distance (theoretically infinity). Figure 7.7 shows the observed Gan's model. The lines in this case are hyperbolas and hence, this observed model is the Gans' model.

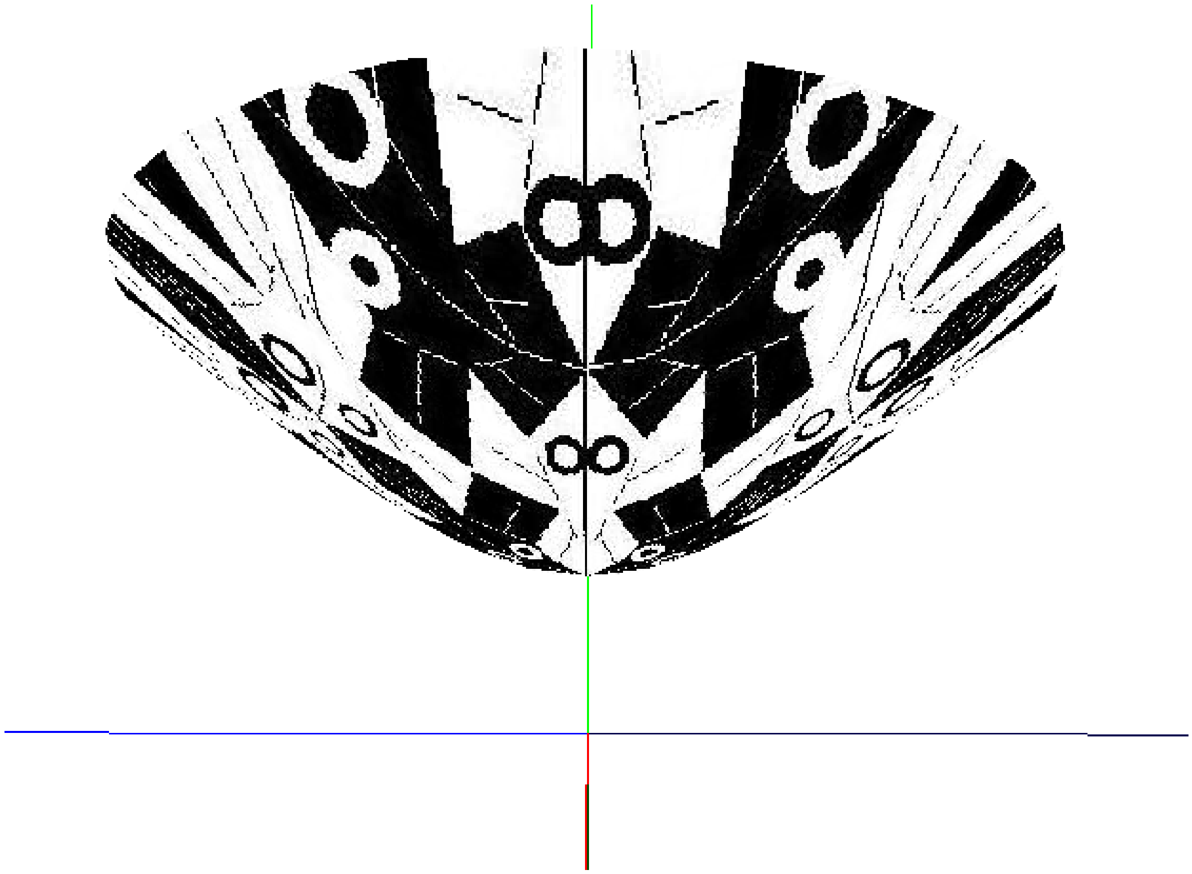

Figure 7.1: Escher's Circle Limit I: A side view of the Weierstrass Model from (0,5,0)

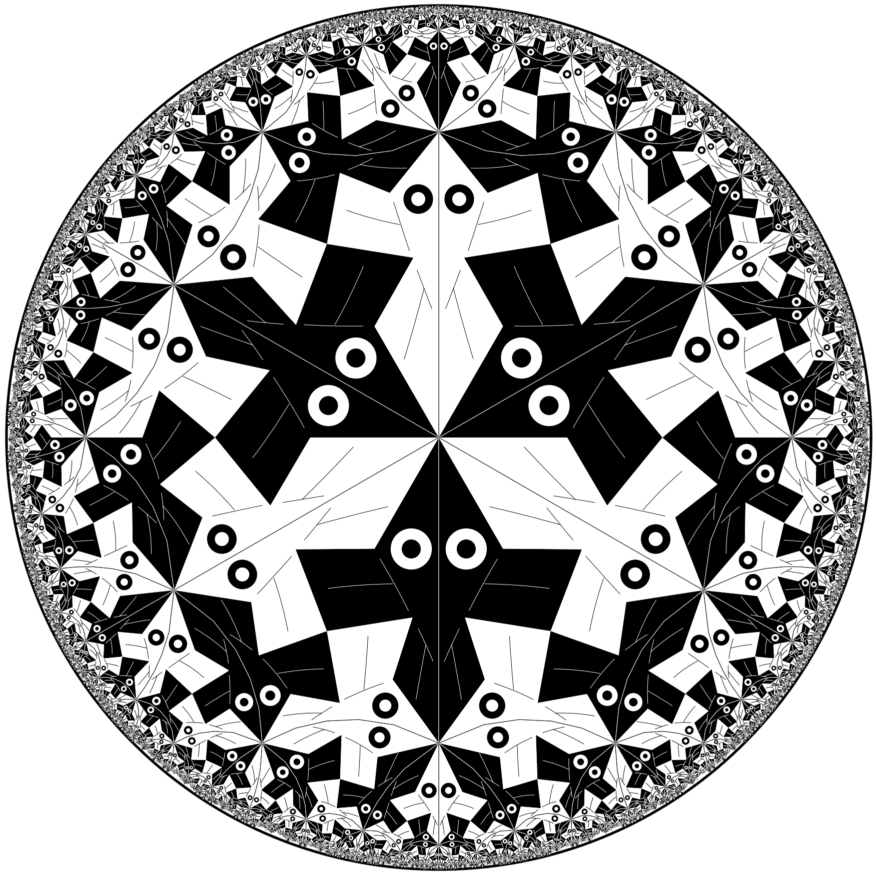

Figure 7.2: Escher's Circle Limit I: Expected Poincaré Model observed at (0,0,-1)

Hence, the Poincaré model, Klein model and the Gans' model can be seen from different viewpoints around the 3D Weierstrass model.

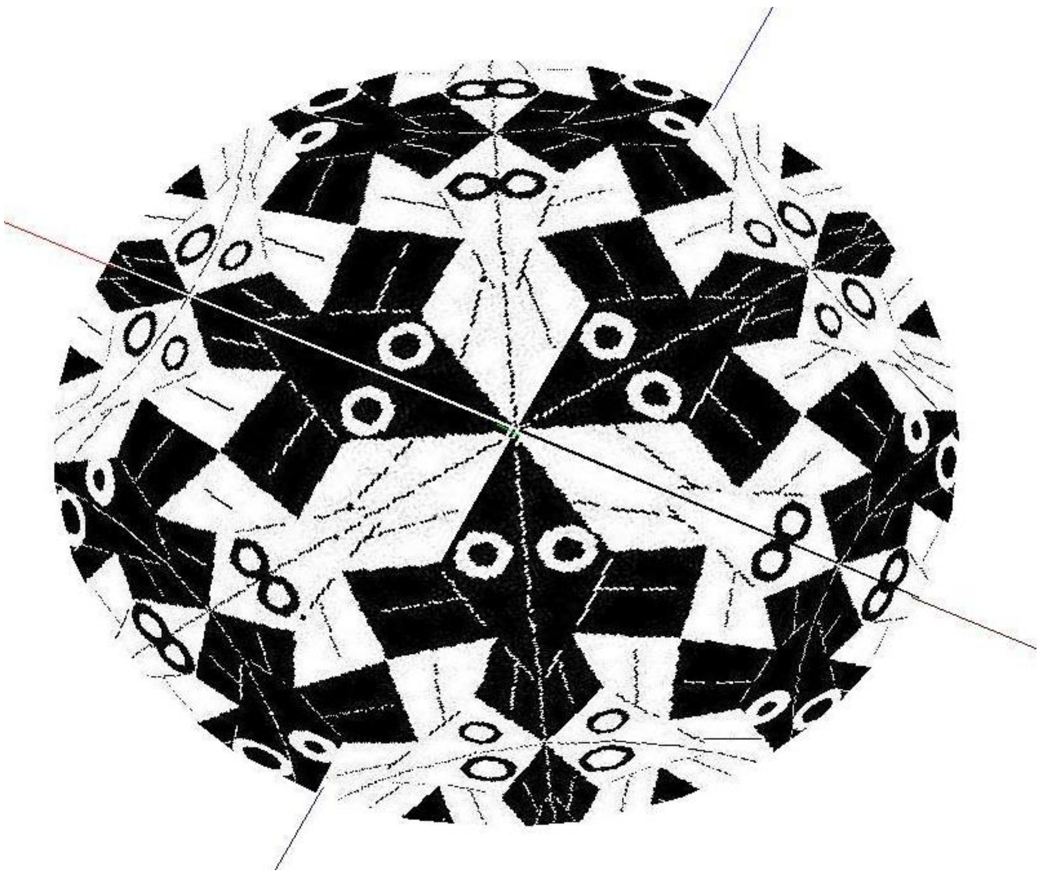

Figure 7.3: Escher's Circle Limit I: Observed Poincaré Model

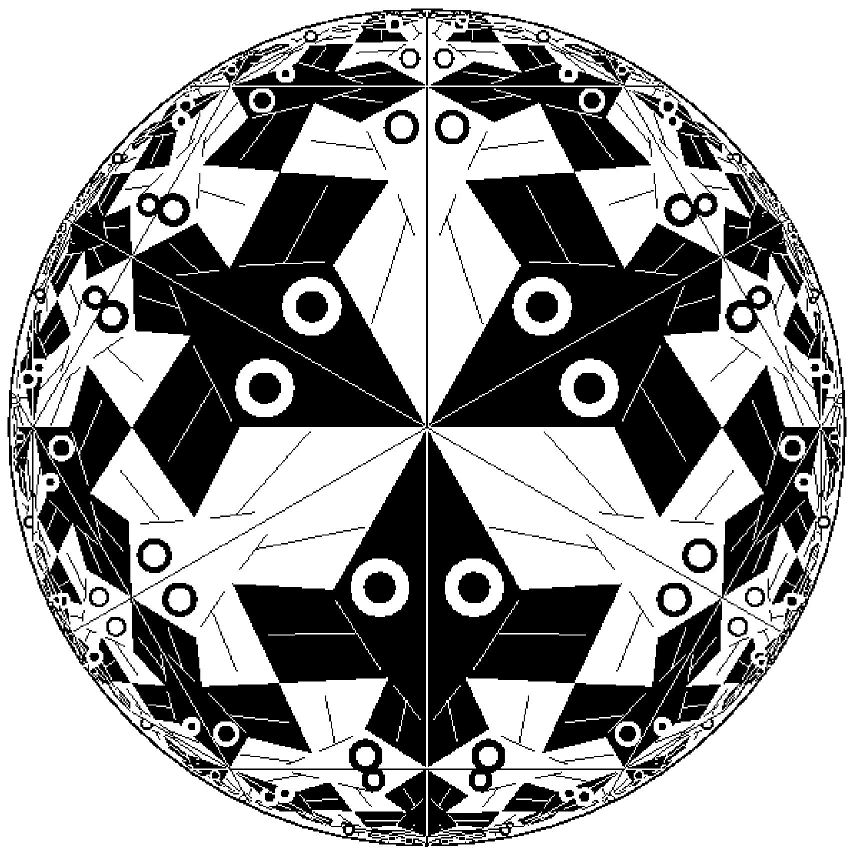

Figure 7.4: Escher Limit I: Expected Beltrami-Klein Model observed at the origin (0,0,0)

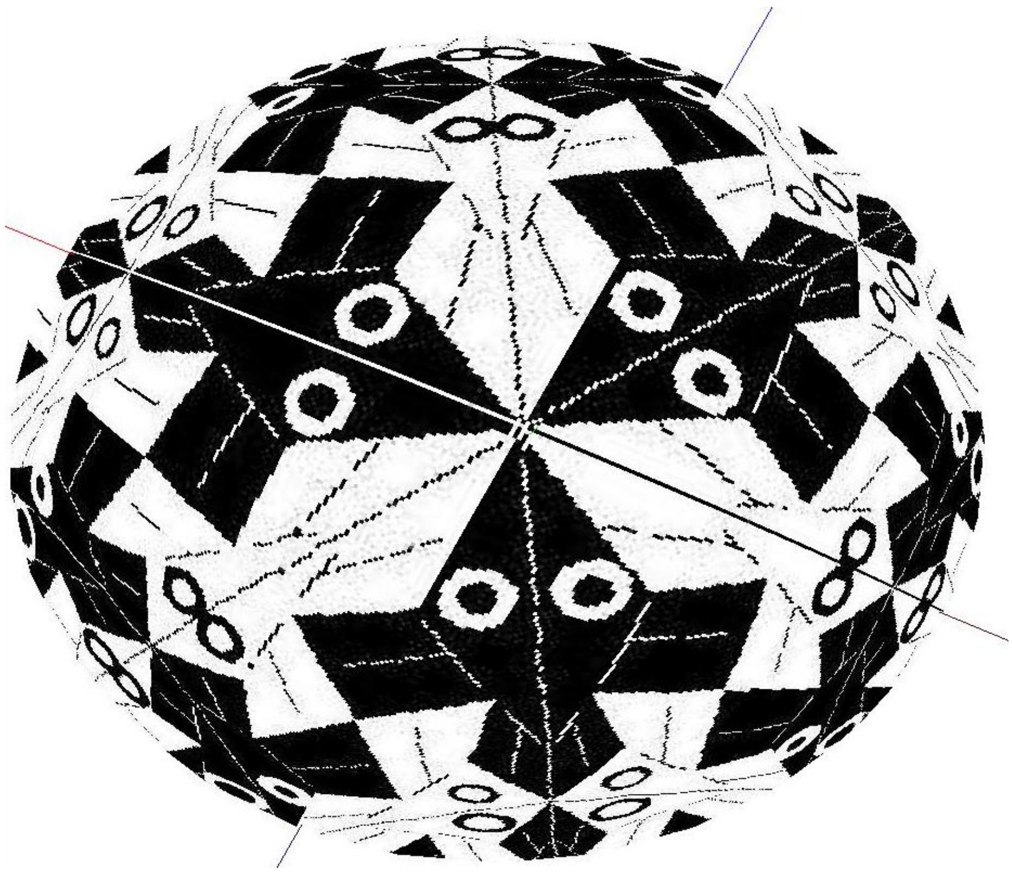

Figure 7.5: Escher Limit I: Observed Beltrami-Klein Model

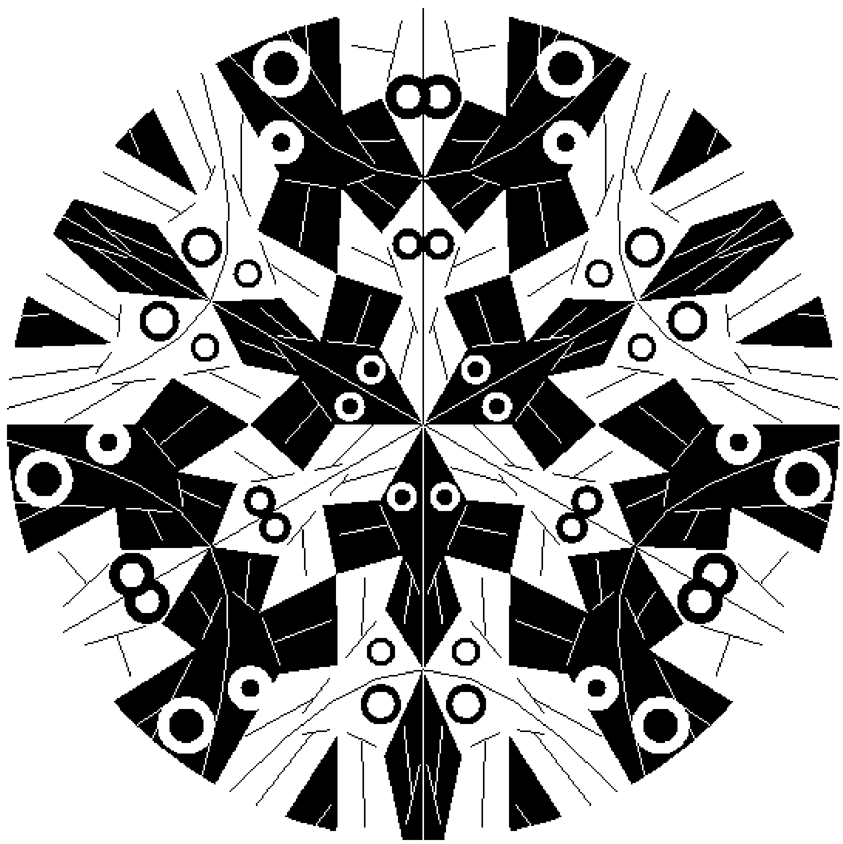

Figure 7.6: Escher Limit I: Expected Gans' Model

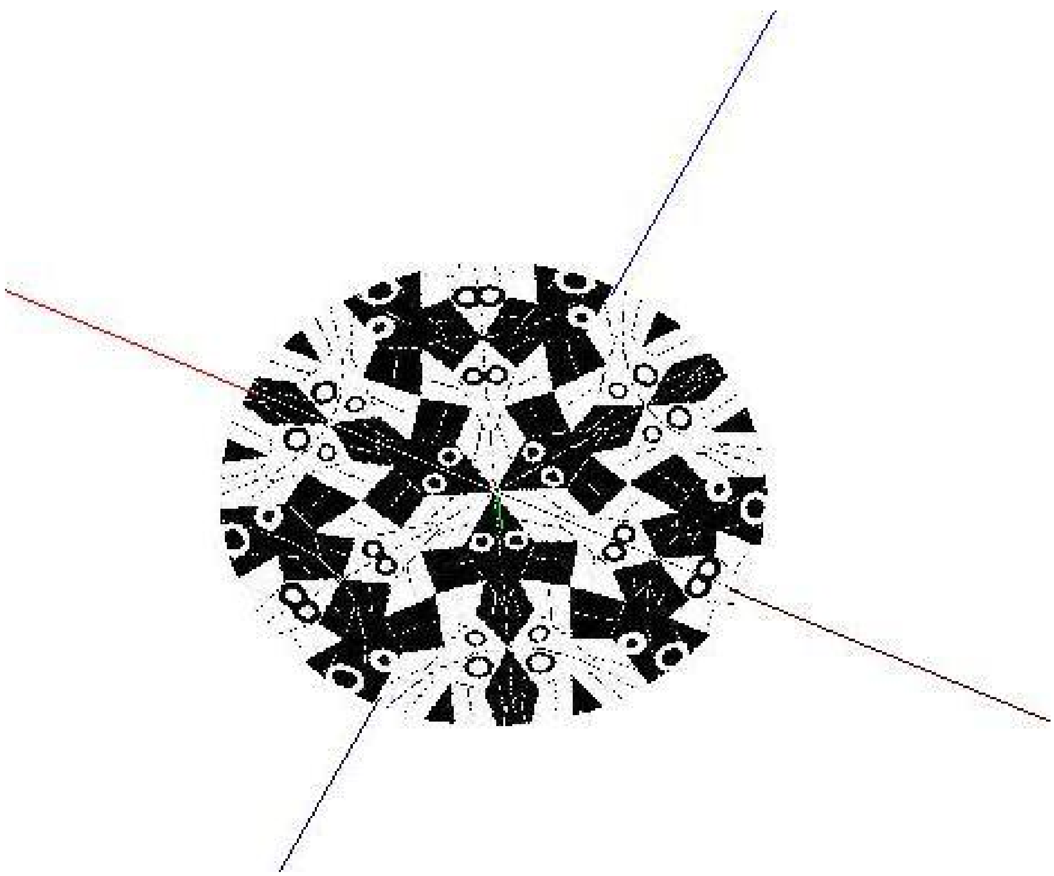

Figure 7.7: Escher Limit I: Observed Gans' Model

Chapter 8

Future Work

In this thesis, we have demonstrated that standard viewing projections of the 3D Weierstrass model can show views of other models of hyperbolic geometry. Thus, it facilitates the visualization of hyperbolic geometry through its different models. This has given us a better understanding of the various models and hyperbolic geometry in general.

However, there are a number of improvements to the graphical interface that could be made to the program to make it more user-friendly. For example, we could provide more controls to the user and allow the user to view his position in the 3-space.

We could allow the user to select a motif file and then generate an entire repeating hyperbolic pattern.

We could set several viewpoints and enable the user to jump to one of them by clicking, and thus instantly observe the model from that view. The user could also be allowed to specify viewpoints by using sliders or typing in coordinates, which would be more precise than trying to control the viewpoint with mouse movements.

This would allow the user to observe from any specified viewpoint. Seeing multiple view and viewpoints at the same time would be of great use to the viewer. Having different views at the same time would also allow the user to visualize the same hyperbolic object from different viewpoints, and also to directly compare the models. This would facilitate comparison and understanding of the Weierstrass and other isomorphic models.

This would make the program more intuitive and a better understanding can be obtained from observing different models.

Chapter 9

Appendix: Source Code

The source code for the Visualization of the Weierstrass Model using Java3D is provided in the following pages. The source code which performs the customized Texture Loading is also attached. This is provided by Sun Microsystems in their Tutorial on Java3D [1].

References

1. Bouvier Denis J. (1999). Getting Started with the Java 3D API. Sun Microsystems, California.

2. Dunham D. (1986). Hyperbolic Symmetry. Computer & Mathematics with Applications, Vol.12B, Nos. 1/2, pp. 139-153.

3. Dunham D. (1986) Creating Hyperbolic Escher Patterns. In M.C.Escher, Art and Science, edited by H.S.M. Coxeter, et al. New York: North-Holland, pp. 241-248.

4. Faber Richard L. (1983). Foundations of Euclidean and Non-Euclidean Geometry. Marcel Dekker Inc, New York.

5. Foley James, van Dam Andries, Feiner Steven, Hughes John, (1997), Computer Graphics: Principles and Practice, 2nd edition, Addison-Wesley Publishing Company, Inc, pp. 229-236.

6. Gans David (1966). A New Model of the Hyperbolic Plane. American Mathematical Monthly, Vol 73, Issue 3, March 1966, pp 291-295.

7. Greenberg M.J. (1994). Euclidean and Non-Euclidean Geometries. W.H.Freeman and Company, New York.

8. Rosenfeld B.A. (1988). A History of Non-Euclidean Geometry: Evolution of the Concept of a Geometric Space. Spring-Verlag, New York.

9. Sowizral Henry, Rushforth Kevin, Deering Michael (2000), The Java 3D API Specification, 2nd Edition, Addison-Wesley Publishing Company, Inc.

10. Trudeau Richard J. (1987). The Non-Euclidean Revolution. Birkhäuser, Boston.