T.C.
MARMARA UNIVERSITY
INSTITUTE FOR GRADUATE STUDIES IN
PURE AND APPLIED SCIENCES

POLYHEDRA METAMORPHOSIS USING CUBIC SYMMETRY IN 3D INTERACTIVE ENVIRONMENT

Onur KARADELİ
Computer Engineer

COMPUTER ENGINEERING

MASTER OF SCIENCE THESIS

SUPERVISOR
Prof.Dr. M.Akif EYLER

ISTANBUL 2006

T.C.
MARMARA UNIVERSITY
INSTITUTE FOR GRADUATE STUDIES IN
PURE AND APPLIED SCIENCES

POLYHEDRA METAMORPHOSIS USING CUBIC SYMMETRY IN 3D INTERACTIVE ENVIRONMENT

Onur KARADELİ
Computer Engineer
(141100320020018)

COMPUTER ENGINEERING
MASTER OF SCIENCE THESIS

SUPERVISOR
Prof.Dr. M.Akif EYLER

ISTANBUL 2006

ACKNOWLEDGEMENTS

Thanks to Prof. Dr. Akif Eyler to help me complete this thesis. Thanks to my mom and grandpa to fill me with wish to complete this thesis.

Onur KARADELİ

TABLE OF CONTENTS

ÖZET

Çok Yüzlüler: yüzeyleri çokgenlerden oluşan, 3 boyutlu geometride üzerinde çok çalışılmış çekici bir konudur. Bunun en önemli nedeni çok yüzlünün taşıdığı yüksek seviyedeki simetridir. Kübik Simetri bu simetri tiplerinden birisi olup, tek bir şekili, belli bir yaklaşımla, birçok şekile dönüştürebilir.

Kübik simetri, bir küre ve bir çekirdek noktası kullanılarak, sınırsız çok yüzlü üretilebilir. Bazı özel durumlar eşsiz çok yüzlülerin çıkarılmasını sağlar. Arşimet ve Platonik cisimleri gibi. Bu tezde özel bir metot kullanılarak 3 boyutlu uzayda çok yüzlülerin oluşturulması sağlanmaktadır.

Kullanıcı etkileşimli bir ortam , bu çok yüzlülerin gösterimi ve doğalarının incelenmesi için son derece önemlidir. Bu yüzden bu çeşit bir 3 boyutlu ortam hazırlanmıştır.

Kullanıcının etkilieşimini de sağlayan 3 boyutlu zahiri bir ortam öte yandan karmaşık bir iştir. Bu yüzden 3 boyutlu uzayda geometri işlemleri ve 3 boyutlu grafik motorlarının kullanımı tez de önemli bir yer teşkil eder.

3 boyutlu geometri ile ilgili olarak; 3 boyutlu noktalar , düzlemler, doğrular, noktaları düzlemler etrafında döndürmek gibi konular kullanılmıştır.

3 boyutlu Grafik Motoru ile ilgili olarak; umut vadeci bir motor olan Java3D kullanılmıştır. Bu araçla 3 boyutlu uzayda nesne işleme işlemleri olan : nesne yaratımı, ekrana çizmesi, nesnenin bir kısmını gizlemek, boyutunu küçültüp büyütmek, çevirmek, gözlemcinin bakış noktasını değiştirmek gibi işlemler kolayca yapılmaktadır.

Ocak 2006 **Onur KARADELİ**

ABSTRACT

Polyhedron (plural Polyhedra) is a three dimensional (3D) solid composed of many polygons. It is a long-studied, charming subject in 3D geometry. The most important reason for this is the high degree of symmetry the Polyhedron holds. Cubic Symmetry is a one of these kinds of symmetries that leads a single shape to transform to many totally different yet familiar shapes with a certain approach.

By using cubic symmetry, a sphere and a seed point unlimited number of Polyhedra can be generated. Some special cases leads to the unique polyhedra with their regularity including Platonic Solids, Archimedean Solids and so on. In this thesis a special approach is used to create Polyhedra in 3D Space.

An interactive environment for visualization of Polyhedra would be very beneficial for the watchers to visualize and understand the 3D nature of the shapes, similarities between shapes, symmetries that a shape and so on. In this thesis such an interactive virtual environment is modeled and created.

3D Virtual Environment that also allows interaction with user is a challenging task however. A great deal of 3D Geometry Knowledge and effective use of 3D Graphics Engine is essential to create such an environment.

Related with 3D Geometry; 3D Points, Planes, Lines, Revolving points around lines/planes, getting symmetries are extensively used.

Related with 3D Graphics Engine; A promising graphics engine called Java3D is used. With this tool the fundamental functions of 3D Object manipulation including; Object creation, drawing, clipping, scaling, revolving, changing the viewpoint and so on was implemented rather easily and effectively.

January 2006 **Onur KARADELİ**

LIST OF SYMBOLS

O_h : O_h Symmetry Type

O : O Symmetry Type

T_h : T_h Symmetry Type

T_d : T_d Symmetry Type

T : T Symmetry Type

n-fold : A notation for symmetrical rotation operation that is 360/n degrees.

LIST OF FIGURES

LIST OF TABLES

PART I

INTRODUCTION

Polyhedron (plural Polyhedra) is a three dimensional (3D) solid composed of many polygons. It has attracted the attentions as it holds a high degree of symmetry. Some of the polyhedra can be seen easily in daily life like cube, tetrahedron… where as, most them are rare to see like rhombcuboctahedron and so on.

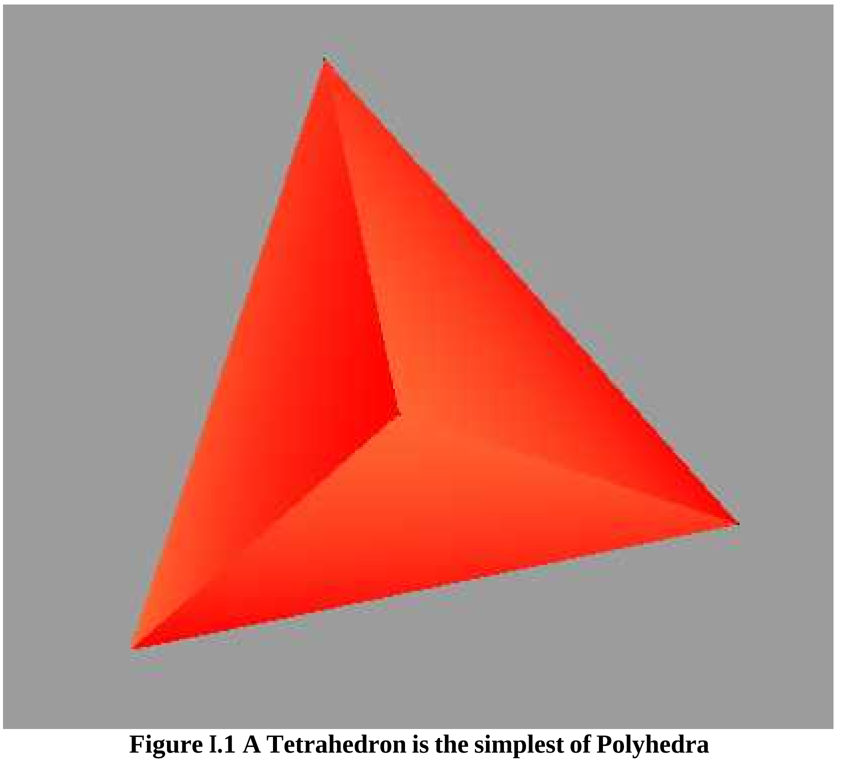


Figure I.1 A Tetrahedron is the simplest of Polyhedra

The aim of this thesis is to use a model to create polyhedra with cubic symmetry and create an interactive environment so that users may visualize and investigate the polyhedra.

I.1 Polyhedra in Life

Polyhedra are subjects of many scientific and social topics throughout the civilization.

I.1.1 Polyhedra in Architecture

In daily life it is customary to see beautiful, symmetrical constructions so that the visitors will memorize the place for very long.

Figure I.2 A One of the Wonders of the world: The Egypt Pyramids

A beautiful construction is in the front of my office so that it is hard for me to forget where should I go.

Figure I.3 The Truncated-Tetrahedron in Istanbul-Atasehir.

I.1.2 Polyhedra In Nature

In molecular scale, the atoms or molecules most of the time represent symmetrical and repeating structures. Some of these structures form Polyhedra when the edges are drawn through vertices (the molecules).

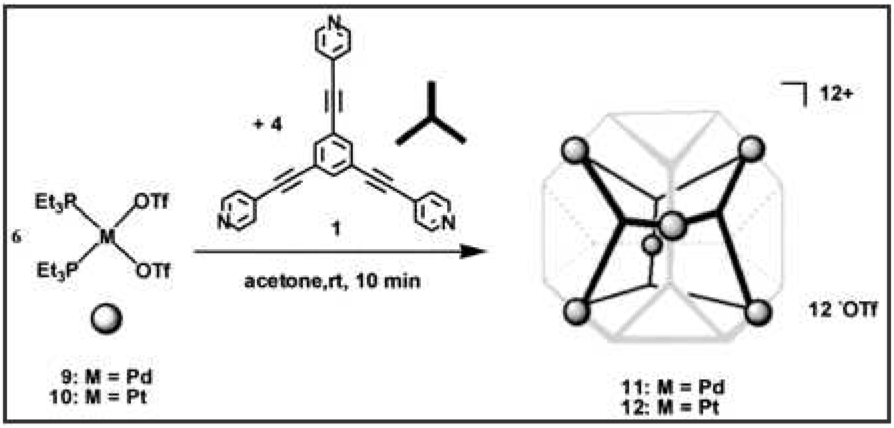


Figure I.4 A Molecular structure forming a Truncated-Tetrahedron.

In games a die is one of the tools that produces regular randomness that all outcomes have the same probability of occurrence. A 6-Faced die is common in many games. There are games where other multi-faced dice are used such as Fantasy Role Playing and so on.

Figure I.5 Multi-Faced dice, 4-6-8-12-20 faces in order. Tetrahedron-Cube-Octahedron-Dodecahedron-Icosahedron as Polyhedra names respectively.

I.1.3 Polyhedra In Geometric Origami

The Japanese art of folding paper to form shapes is also interested with Polyhedra. Many of the Polyhedra can be built with Origami with varying difficulties.

Figure I.6 Some polyhedra that are created with Origami.

I.2 Problem Analysis

Polyhedra metamorphosis is the continuous change of one polyhedron to another.

In this thesis a family of polyhedra, which owns Cubic Symmetry is studied, created and visualized. To model a polyhedron: 3D Geometry is used extensively used. In order to depict and to create an interactive environment a 3D Graphics Engine is used.

After creating a 3D Interactive environment then polyhedra metamorphosis is observed and similarities are identified between polyhedra.

PART II

POLYHEDRA AND SYMMETRY

II.1 Polygon

Polygon: the building block of Polyhedra is a closed plane figure bounded by three or more line segments. A polygon can be either convex or concave. A polygon is convex if it contains all the line segments connecting any pair of its points.

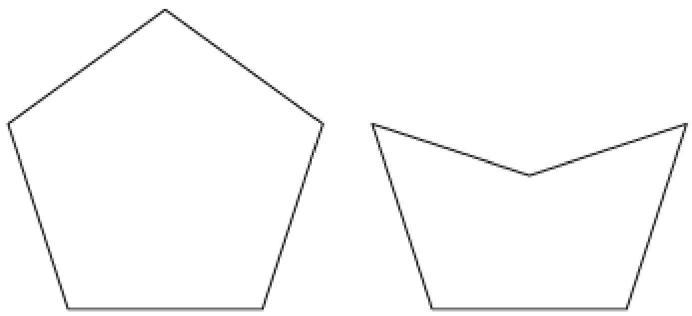


Figure II.1 A convex and a concave polygon.

Another property of the polygon is its regularity.

A regular polygon is a convex polygon with all its are sides equal and all its angles are equal

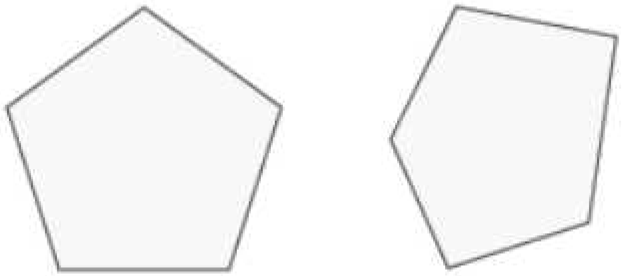


Figure II.2 A regular (pentagon) and an irregular polygon

II.2 Polyhedra

If we rephrase the definition of polyhedra, Polyhedra (plural Polyhedra) are 3 dimensional (3D) solids composed of many polygons. These polygons can be either convex or concave. In this thesis however polyhedra that contains concave polygons are not studied.

II.3 Convexity

Similar to the convexity of a polygon the convexity in polyhedra means with every pair of points that belong to the polyhedra, the polyhedra contains the whole straight line segment connecting the two points.

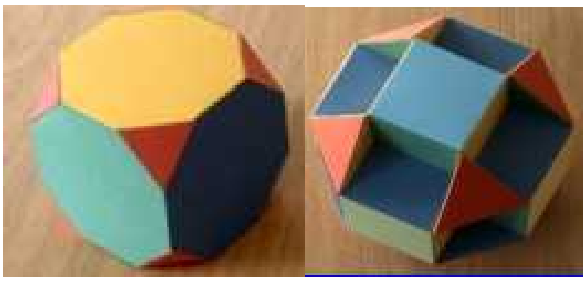


Figure II.3 A Convex and a nonconvex(concave) polyhedra

In this thesis convex polyhedra are studied.

II.4 Regularity

Regularity of a polyhedra is determined with the faces of the polyhedra and the properties of these faces.

There are three types of regularity that polyhedra can have; Regular, semi-regular and irregular polyhedral. In this thesis all three regularity types of polyhedra can be generated however regular and semi-regular polyhedra types are studied extensively.

II.4.1 Regular Polyhedra

A Polyhedron is said to be regular if

i) all its faces are the same
ii) and all the faces are regular polygons

There are 5 convex polyhedra that are regular. Those are also known as Platonic solids

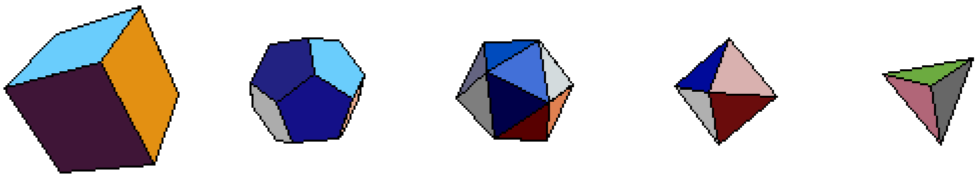


Figure II.4 The five convex regular polyhedra: Cube, dodecahedron, icosahedron, octahedron and tetrahedron

In this thesis four of them can be generated. Namely: tetrahedron, cube, octahedron and icosahedron.

Also there are 4 nonconvex polyhedra, which are regular as well. Those are known also as Kepler-Poinsot solids.

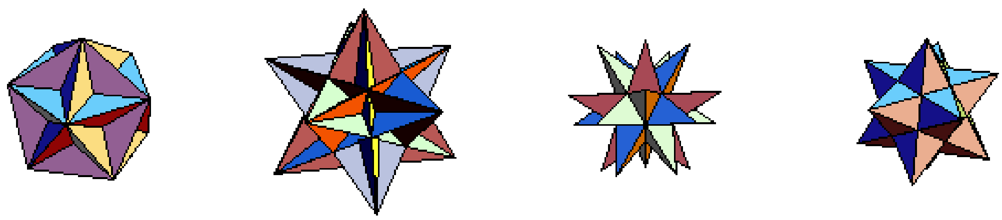


Figure II.5 The four nonconvex regular polyhedra: Great dodecahedron, great icosahedron, great stellated dodecahedron, small stellated dodecahedron.

II.4.2 Semi-regular Polyhedra

i) all its faces are NOT the same
ii) all the faces are regular polygons

There are two infinite sets of polyhedra, the set of prisms (Figure II.6), the set of antiprisms (Figure II.7), and the 13 Archimedean solids (Figure II.8) are semiregular polyhedra.

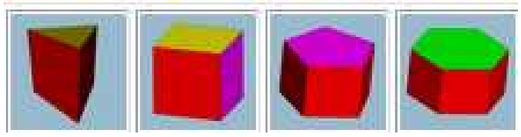


Figure II.6 Some prisms: Triangular prism, cube, pentagonal prism, and hexagonal prism.

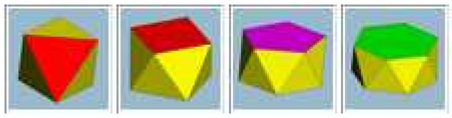


Figure II.7 Some anti-prisms: octahedron, square antiprism, pentagonal antiprism, hexagonal antiprism.

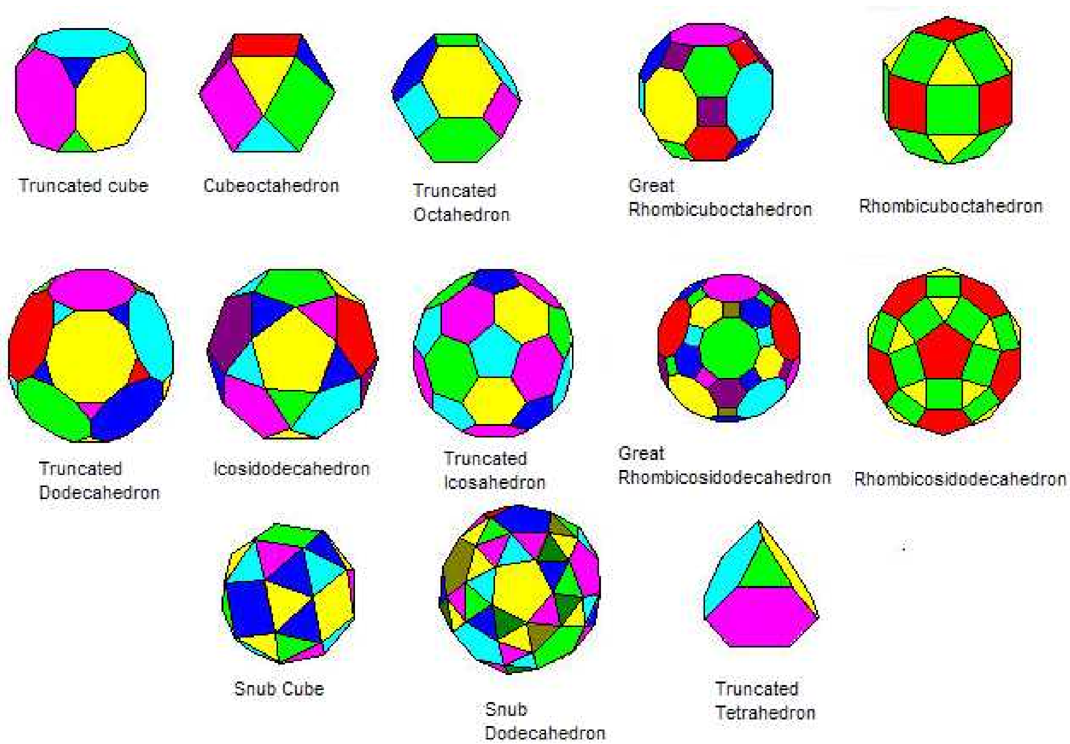


Figure II.8 The Archimedean solids.

II.4.3 Irregular Polyhedra

iii) all its faces are NOT the same
iv) and some of the faces are NOT regular polygons

There exists infinite amount of irregular polyhedra. Two of them are the following.

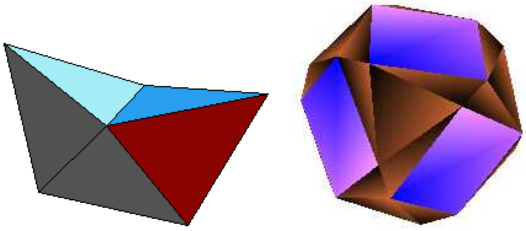


Figure II.9 A Boat and a deformed snub cube..

II.5 Symmetry

In the context of Polyhedra, symmetry implies a Polyhedron looks the same from another different viewpoint. Alternatively a polyhedron is symmetrical if it is possible to perform certain operations that change the positions in space of individual faces but which leave the polyhedron that is indistinguishable from its original position. There are 2 symmetry operations: rotation and reflection that forms the symmetry types.

II.5.1 Rotational Symmetry

If a Polyhedra looks the same after rotating the polyhedra on a specific axis then the polyhedra holds a rotational symmetry.

Rotational Symmetry is characterized by the **axis of rotation** and the **angle of rotation**. Sometimes there is more than one rotational symmetry axis.

The angle of rotation is the minimum angle of rotation on a given axis that leads to an indistinguishable (symmetrical) shape. As an example in figure II.11. the light blue axis has 120 degrees of rotation symmetry thus after rotating the tetrahedron 120 degrees along any light blue axis the new shape looks the same again.

Another notation for the angle of rotation is to indicate the number of rotations in a full (360 degrees) turn. 120 degrees of rotational symmetry thus can be called 360/120=3 fold symmetry. 3-fold symmetry indicates the shape looks the same 3 times in a full rotation.

In the figure below the tetrahedron is studied for its rotational symmetries.

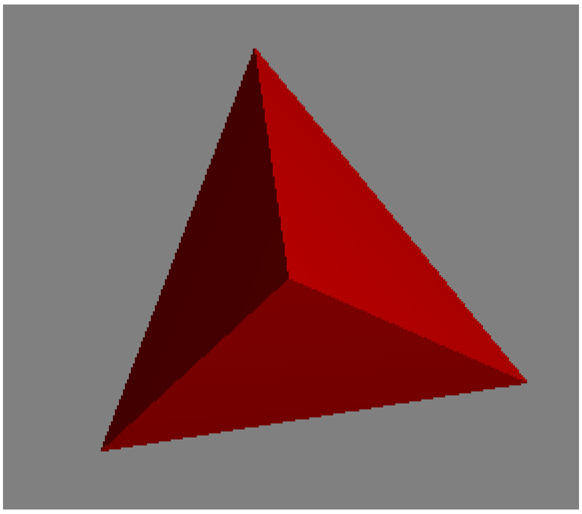


Figure II.10 The Tetrahedron

When we study the tetrahedron at figure II.10 we can we find the following rotation axis and consequently rotational symmetries.

- Four rotational axis passing through from one vertex (highlighted with blue) and middle point of the opposite face (highlighted with yellow)

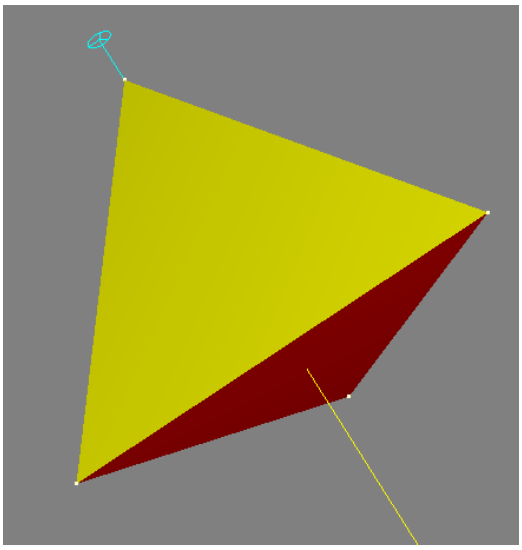


Figure II.11 The 3-fold rotation axis of tetrahedron

On the other hand this rotational axis creates 3 symmetries that is when each time a yellow face is exactly in front of our viewpoint.

Thus the tetrahedron has **four 3-fold rotational symmetry**.

- Tetrahedron has three rotation axis which passes from the middle of one edge (highlighted with blue) and the middle of the edge at the opposite side edge (highlighted with yellow)

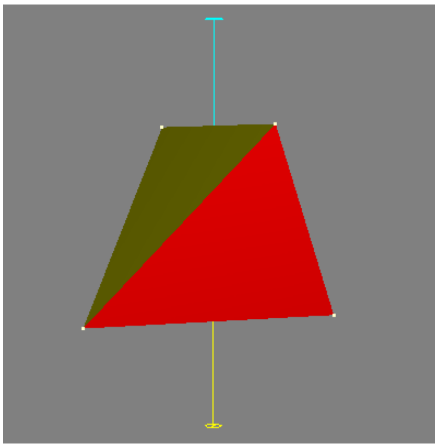


Figure II.12 The 2-fold rotation axis of tetrahedron

On the other hand this rotational axis creates 2 symmetries that is when each time a vertex is exactly in front of our viewpoint thus the tetrahedron has **three 2-fold rotational symmetry**.

II.5.2 Reflection Symmetry

Reflection symmetry involves defining a plane of reflection for the solid. Then slicing the solid into two parts using the plane of reflection. If the two parts are identical then the solid is said to have reflection symmetry for the given plane of reflection.

For example human body does not hold any rotational symmetry. However it holds a clear reflection symmetry. (i.e. after visually slicing the body from head: the left arm is identical with right arm , left eye with right eye…)

In tetrahedron there are 6 planes of reflection as depicted below

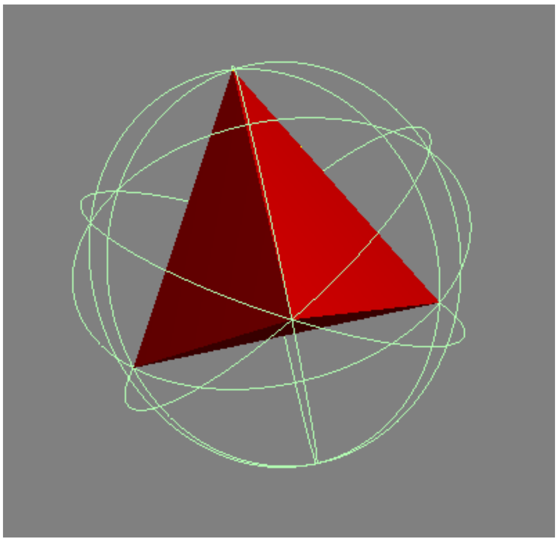


Figure II.13 Tetrahedron has 6 reflection planes.

II.6 Cubic Symmetry Types

With the symmetry operations (rotation, reflection) there are 17 types of symmetry in 3D geometry. Each of these symmetry types contains a set of Rotational Symmetries and a set of reflection symmetries. In this thesis **only Cubic Symmetry types are studied**. 5 of the symmetry types are cubic as the following.

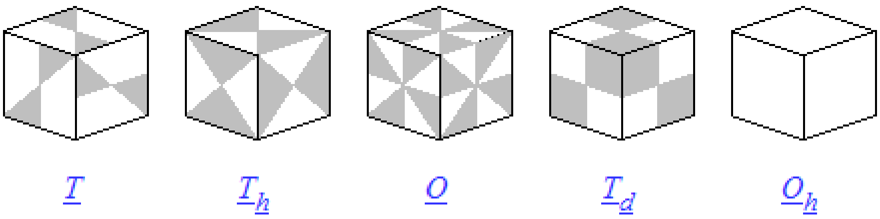


Figure II.14 Some polyhedra carrying a cubic symmetry type

In the following section what these symmetry types imply and how they are used are given.

In depth explanations and illustrations of the 17 Symmetry types are given in Cromwell's book on Polyhedra [PET].

II.6.1 Cubic Symmetry Type: O_h

The last (5th from the beginning) cube in figure II.14 is said to have Oh symmetry type. This solid has several rotational and reflection symmetries. Oh symmetry type implies the following symmetries exist in the solid.

This symmetry type is called ‘O’ because it has the identical rotational axes of a **Octahedron**. Also there is a ‘h’ in the naming which indicates there is (are) a horizontal plane of reflection as well

Rotational

- 3 Axes of 4-fold symmetry which joins the centers of opposite faces
- 4 Axes of 3-fold symmetry, which joins diagonally opposite vertices.
- 6 Axes of 2-fold symmetry which joins midpoints of opposite edges

Reflection

- 3 Mirror Planes passing through the middle points of 4-parallel edges
- 6 Mirror Planes passing through an edge and its opposite side edge.

II.6.2 Cubic Symmetry Type: O

The 3rd cube in figure II.14 is said to have O symmetry type. The O symmetry type has lost some of the Rotational symmetries and all Reflection symmetries of an Octahedron. Thus it only has the,

Rotational

- 4 Axes of 3-fold symmetry, which joins diagonally opposite vertices.

II.6.3 Cubic Symmetry Type: T

The 1st cube in figure II.14 is said to have T symmetry type. This symmetry type is called ‘T’ because it has the identical rotational axes of a **Tetrahedron**.

Rotational

- 4 Axes of 3-fold symmetry, which joins diagonally opposite vertices.
- 3 Axes of 2-fold symmetry which joins the centers of opposite faces

Reflection

- 3 Mirror Planes passing through the middle points of 4-parallel edges

II.6.4 Cubic Symmetry Type: T_h

The 2nd cube in figure II.14 is said to have Th symmetry type. This symmetry type is called 'T' because it has the identical rotational axes of a **Tetrahedron**. Also there is a 'h' in the naming which indicates there is at least one horizontal plane of reflection as well

Rotational

- 4 Axes of 3-fold symmetry, which joins diagonally opposite vertices.
- 3 Axes of 2-fold symmetry which joins the centers of opposite faces

Reflection

- 3 Mirror Planes passing through the middle points of 4-parallel edges

II.6.5 Cubic Symmetry Type: T_d

The 2nd cube in figure II.14 is said to have Th symmetry type. This symmetry type is called 'T' because it has the identical rotational axes of a **Tetrahedron**. Also there is a 'd' in the naming, which indicates there is (are) a vertical plane of reflection as well

Rotational

- 4 Axes of 3-fold symmetry, which joins diagonally opposite vertices.
- 3 Axes of 2-fold symmetry which joins the centers of opposite faces

Reflection

- 6 Mirror Planes passing through an edge and its opposite side edge.

PART III

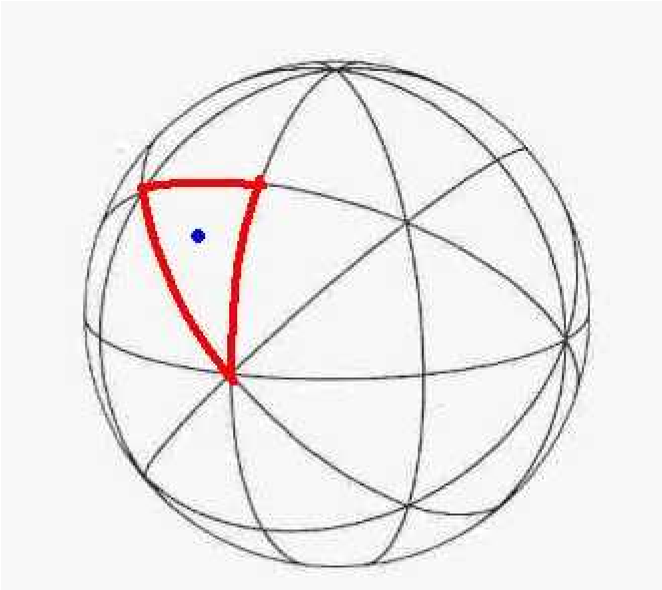
CREATING POLYHEDRA WITH CUBIC SYMMETRY

The idea of creating polyhedra using cubic symmetry that was first proposed by Cromwell [PET] requires 2 parameters and 2 operations.

- Parameter 1: A Seed Point
- Parameter 2: The symmetry Type (Oh, T …)
- Operation 1: Rotation Operations and Reflection Operations (depending on the symmetry type)
- Operation 2: Taking the convex hull of the final points

III.1 Model Sphere

To implement cubic symmetry the existence of a model sphere is very helpful.



®

Figure III.1 The Model sphere for Cubic Symmetry

This sphere is the skeleton to take the symmetry points and the blue point is the seed point. The axes slicing the sphere are rotation axes and reflection planes.

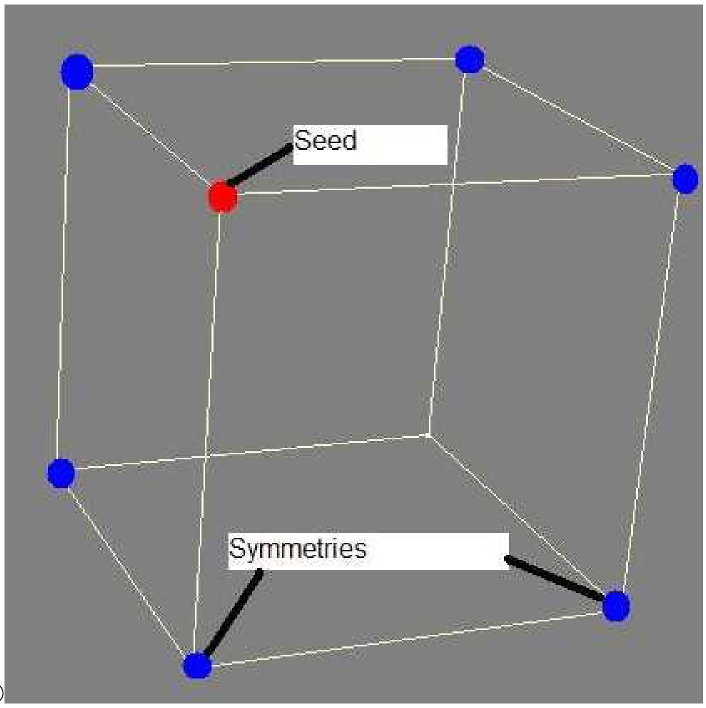
Every triangle (highlighted with red) in this sphere includes exactly 1 point (blue) inside the triangle, which is on the sphere.

After applying the symmetry operations, each region (red triangle) will contain at most 1 symmetry point, inside or on itself. There are 48 regions on the Model sphere, thus any polyhedron that is created in this thesis has at most 48 points.

For some special seed points however many symmetry points are on top of others. Thus the actual number of symmetry points decrease to a number smaller than 48. The intersection of lines in figure III.1 and points on these lines are some examples that will lead to special types polyhedra which will be examined in later sections.

III.2 Seed Point

Seed point (the red point in figure III.1 and figure III.2) is the original point, which is used to take symmetries and reflections. Those symmetrical and reflection points to the seed point are also taken as symmetries and reflections iteratively for each Rotation Axis and Reflection Planes.



®

Figure III.2 The seed point and derived symmetrical points after rotation and reflection operations

In figure III.2 the seed point is highlighted with red. After applying several rotation operations the symmetry points highlighted with blue are created.

III.3 Finding Rotation and Reflection Symmetries

The seed point is used to derive a of number symmetrical of points for the given rotational axes (if any) and reflection planes (if any). As declared in Chapter 2, some of the Cubic Symmetry Types have only rotational symmetries (O, T) and some of them have both rotational symmetries and reflection symmetries. (Oh, Td…) The symmetry type determines which operations to use: rotation operations, reflection operations or both.

We developed an algorithm to create the symmetry points using the seed point and symmetry type. The algorithm that is used to derive symmetrical points is as the following. The 3D geometry section gives the details of operations such as revolving a point around an axis and so on.

1. Initially create an operation set including only the **seed point**.
2. For each rotation axis
    a. For each point in the operation set
        i. Create a rotatedPoints set including all the points of the operation set
        ii. Rotate the current point around the current rotation axis and find a symmetry point.
        iii. Add this symmetry point to the set of rotatedPoints if not already exists.
    b. Reset the operation set to the rotatedPoints set.
3. For each reflection planes
    a. For each point in the operation set
        i. Create a mirroredPoints set including all the points of the operation set
        ii. Get mirror image of the current point against the current reflection plane.
        iii. Add this symmetry point to the set of mirroredPoints set if not already exists.
    b. Reset the operation set to the mirroredPoints set.
4. The set of points that the polyhedra will have is found.

Always starting with 1 point (the seed point) , the derived points may increase in number twice each time a reflection or rotation operation is applied.

Maximum 48 points may emerge at the end of this procedure, because many points will be the same where only one of them is taken.

III.4 Finding the convex hull to create Polyhedra

After the points of the polyhedra are found, a typical 3D Convex Hull finding algorithm and a java package [STO] is used to get the edges and faces of the Polyhedra.

The convex hull of a set Q of points is the smallest convex polygon (polyhedron) P for which each point in Q is either on the boundary of P or in its interior.

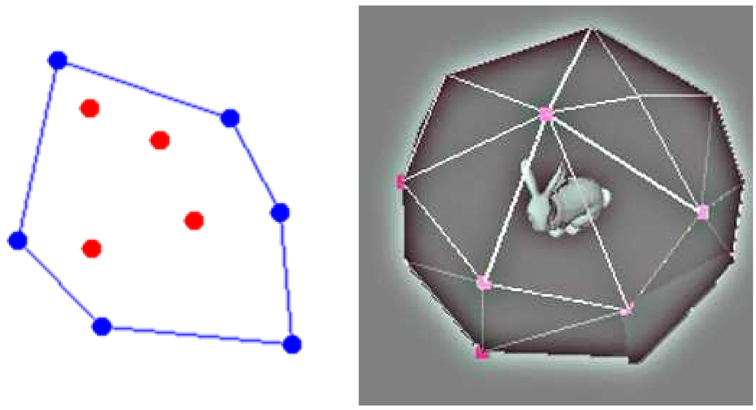


Figure III.3 Convex hull of a polygon(left) and polyhedra (right).

PART IV

TESELLATION 3D APPLICATION

Different seed points and different type of symmetries lead to infinite kinds of polyhedra. Among these polyhedra some of them are special.

To examine the study of Polyhedra Creation with cubic symmetry a 3D Application: Tessellation 3D is developed using the Java 3D Graphics Engine.

In this chapter the polyhedra will be created with different parameters (seed point, rotation, reflection operators) and the similarities between them are compared.

IV.1 Tessellation 3D Application

Tessellation 3D is an interactive 3D environment to create and manipulate Polyhedra using cubic symmetry. It will enable the users.

- To create polyhedra parametrically (seed point and symmetry types)
- To rotate and locate the polyhedra to see different views of polyhedra.

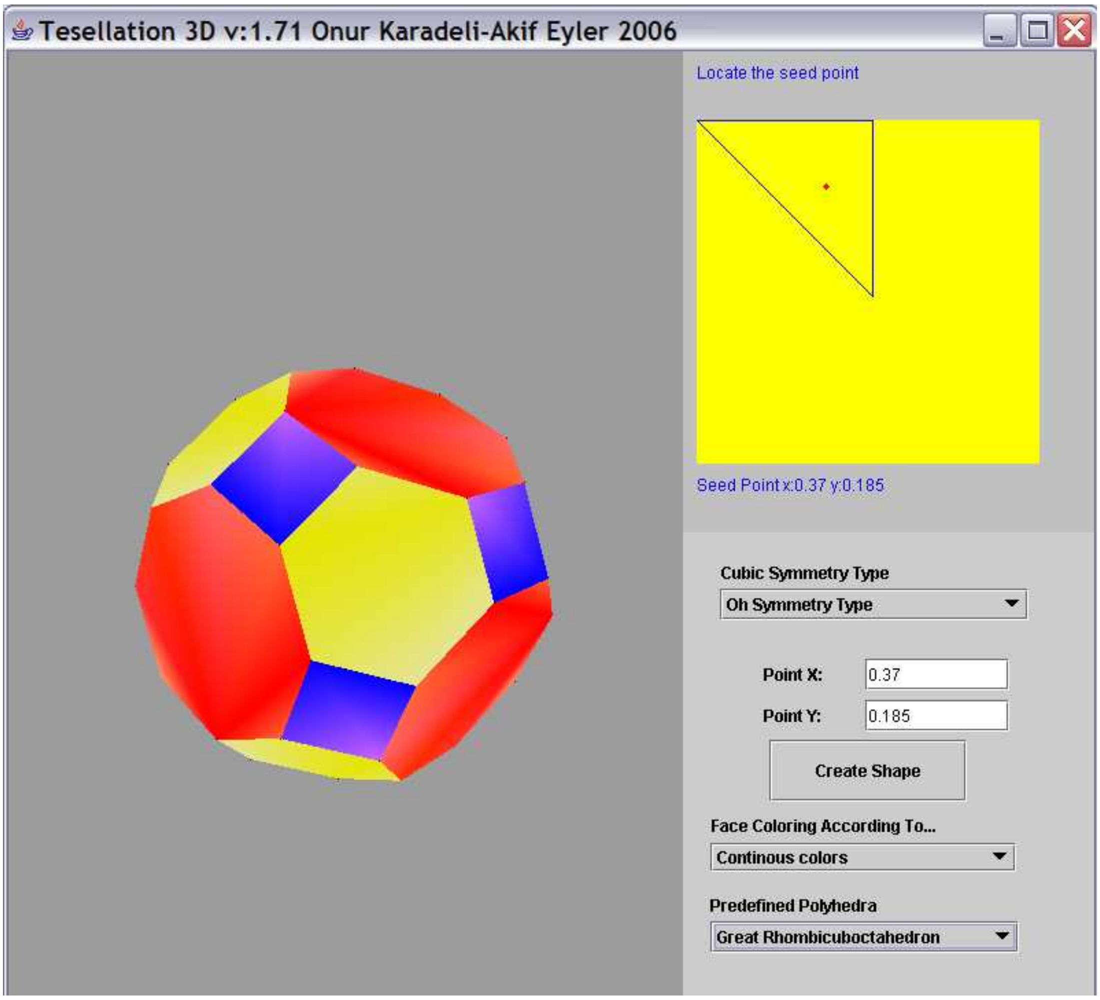


Figure IV.1 The Tessellation 3D Application

IV.1.1 Interactive Polyhedra Object

On gray canvas the 3D Polyhedra is drawn and is in interaction with user mouse actions.

- Rotate (left mouse dragging)
- Locating the viewpoint (right mouse dragging)
- Zooming in and Zooming out (Shift+left mouse dragging)

IV.1.2 Seed Point Selector (Mouse)

As described in section 3.1 a seed point is the origin of points that the symmetry operations are applied upon. A different seed point may lead to a completely different polyhedron. Instead of a spherical region in figure III.1 to select a seed point , a two dimensional approximation of the sphere region is used.

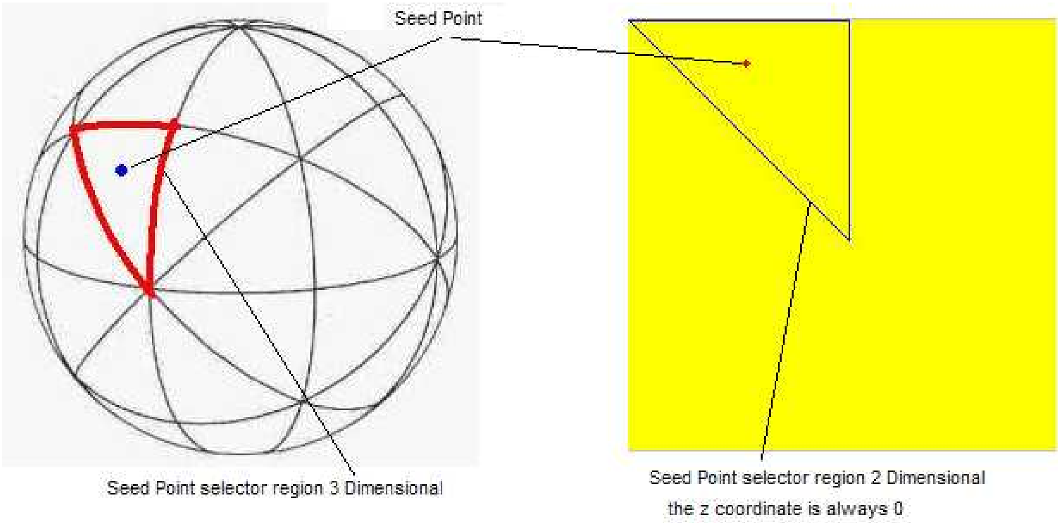


Figure IV.2 The ideal seed point selection region (left) and approximate 2d seed point selection region (right)

As shown in figure IV.2 in Tesellation3D application instead of an ideal sphere(3D) for seed point selection, a 2D triangle is used. . There are two approximations with this approach

1. The z coordinate is taken as 0. However for every seed point in a 3D selection (left model at figure IV.2) region the z coordinate actually is slightly greater than 0.
2. The triangle at the 2D selection region (right model at figure IV.2) is actually not a “right triangle”. The edge lying at x coordinate of the triangle is slightly shorter than the edge lying at the y-coordinate of the triangle.

For the sake of simplicity of this application however these 2 approximations are used and 2D selection region is used instead of 3D selection region.

In figure IV.3 the red point in the blue triangle is the seed point. The seed point can be selected with a mouse click, which immediately generates a new polyhedron.

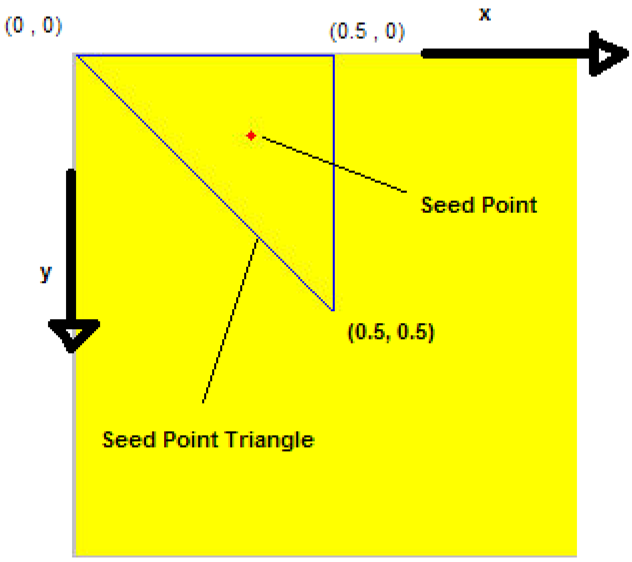


Figure IV.3 Seed Point Selector region

The seed point selector region (figure IV.3) determines only the X and Y coordinates of the seed point. The z coordinate of the seed point is accepted as 0. Thus seed point will be shown as (x, y) instead of (x, y, 0) unless stated otherwise.

IV.1.3 Seed Point Selector (Keyboard)

To enable the users to define more accurate seed points a keyboard controlled seed point coordinate selector is implemented (figure IV.4).

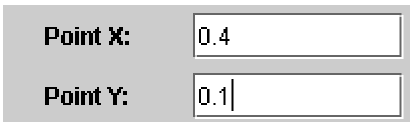


Figure IV.4 Accurate Seed Point Selector

The x and y coordinates of the seed point can be configured here. After pressing the “Create” button, polyhedron is constructed with the new seed point.

Again to note the z coordinate of the seed point is accepted as 0 and can’t be selected from any of the seed point selector regions.

IV.1.4 Symmetry Type Selector

Among 5 cubic symmetry types (in figure IV.5) user can select which symmetry type to operate.

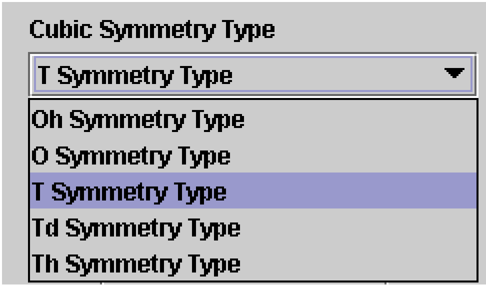


Figure IV.5 Symmetry type selector.

IV.1.5 Face Coloring Method Selector

There are two coloring methods to colorize the faces of polyhedra.

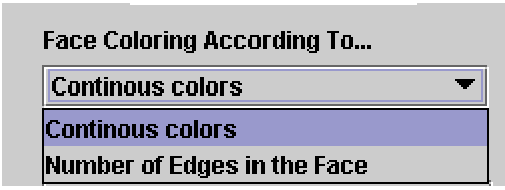


Figure IV.6 Symmetry type selector.

These two coloring methods are discussed at section V.3

IV.1.6 Predefined Polyhedra

Some special parameters lead to special polyhedra (will be discussed), those polyhedra (in Table IV.1) can be selected with this tool and immediately the initial parameters are configured.

Polyhedra Name	**Symmetry Type**	**Seed Point**
Tetrahedron	T	(0, 0)
Cube	Oh	(0, 0)
Octahedron	Oh	(0.5, 0.5)
Icosahedron	Th	(0.5, 0.2)
Truncated Tetrahedron	T	(0.34, 0.34)
Cubeoctahedron	Td	(0.5, 0)
Truncated Octahedron	Oh	(0.5, 0.26)
Truncated Cube	Th	(0.3, 0)

Small Rhombcuboctahedron	Oh	(0.3, 0.3)
Great Rhombicuboctahedron	Oh	(0.37, 0.18)
Snub Cube	O	(0.35, 0.23)

Table IV.1 Several special polyhedra that can be created with Tesellation3D application.

In the above table, it is shown that cube can be created with symmetry type Oh and seed point (0, 0). However it is not the only way to create this polyhedron the same cube can be created with the symmetry type O and seed point (0, 0). This is because for some special seed point locations, some of the reflection or rotation operations do not yield a different symmetry point than that is already found. (Refer to "Finding Reflection and Rotation Symmetries" section)

PART V

POLYHEDRA METAMORPHOSIS AND SIMILARITY

V.1 Generic Polyhedra

Varying parameters in the input leads to different yet similar polyhedra in this study. Depending on the two initial parameters: the symmetry type and the seed point.

The seed point has a dramatic effect on determining the polyhedra that is generated. The seed points' location leads to different polyhedra. However there are in fact 7 classes of locations of the seed point in seed point triangle that are completely different.

Seed Point Location Type	**Seed Point Location**
1	Seed point is exactly at the left-top corner.
2	Seed point is exactly at the right-top corner.
3	Seed point is exactly at the right bottom corner.
4	Seed point is on the diagonal line (between points left-top and right bottom)
5	Seed point is on the horizontal line
6	Seed point is on the vertical line
7	Seed point is inside the triangle.

Table V.2. Seven classes of seed point location.

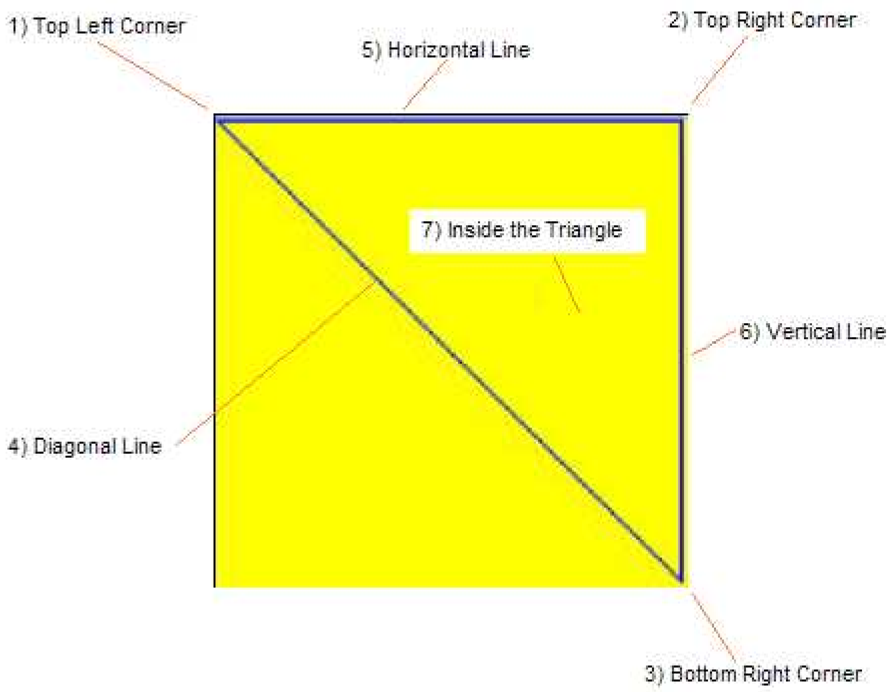


Figure V.1 Seven classes of seed point location

Below in figure V.2 seven polyhedra are generated with these seed point location classes in order and Oh with the symmetry type.

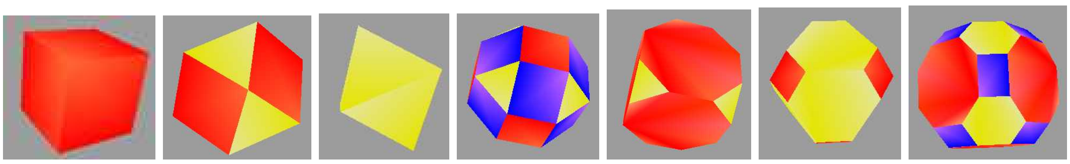


Figure V.2 Example Polyhedra that are created with 7 types of points and Oh symmetry type

V.2 Similarities Between Polyhedra and Special Polyhedra

For the classes of locations 4,5,6 and 7 there may be infinite number of seed point positions thus polyhedra.

For example the 4th class of seed point location leads to the following polyhedra metamorphosis.

We start with a cube at seed point (0, 0)

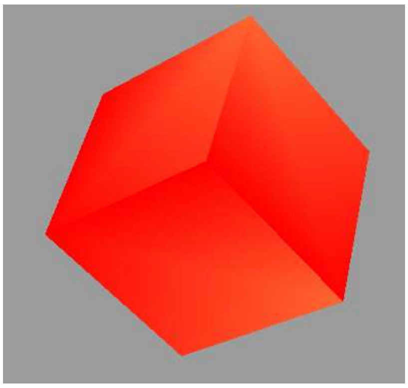


Figure V.3 A cube with seed point (0, 0) and Symmetry Type=Oh

When x coordinate is slightly changed a truncated-cube is obtained.

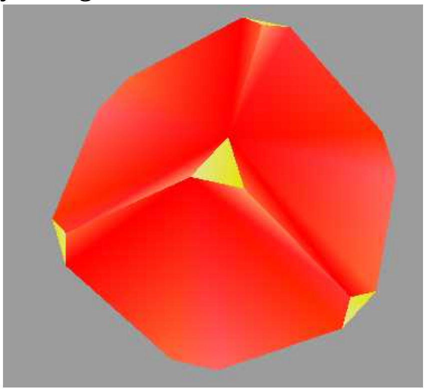


Figure V.4: A truncated cube like polyhedron with seed point (0.15, 0) and Symmetry Type=Oh

Continuing increasing the x coordinate

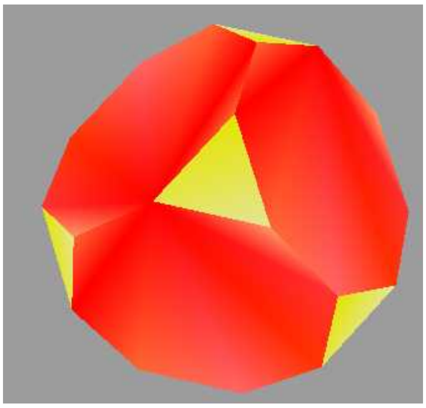


Figure V.5 A regular truncated cube with seed point (0.30, 0) and Symmetry Type=Oh.

This truncated cube has all Regular Hexagons and Equilateral Triangles.

Continuing increasing the x coordinate

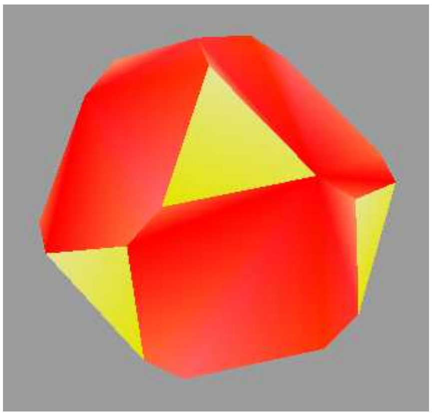


Figure V.6 A truncated cube like polyhedron with seed point (0.40, 0) and Symmetry Type=Oh.

It can be seen that the polyhedra is gradually metamorphosing to octahedron, actually by reaching the top right corner (0.50, 0) an octahedron is obtained.

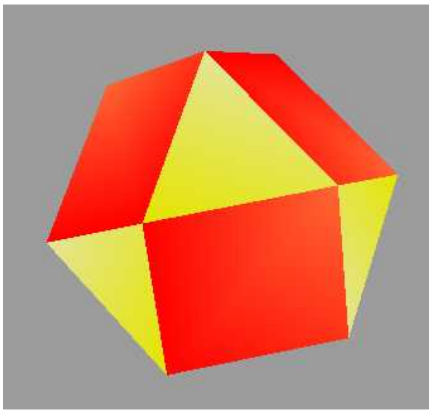


Figure V.7 A cubeoctahedron at seed point (0.50, 0) and Symmetry type=Oh

The metamorphosis is complete and we have four kinds of polyhedra.

1- Cube: All faces are squares
2- Truncated-cube: Faces are octagons or equilateral triangles
3- A deformed truncated-cube: Faces are either deformed octagons or deformed equilateral triangle.
4- Cubeoctahedron: All faces are either squares or equilateral triangles.

This operation can be done for the other classes of seed point location as well.
For every location type a special point leads to a special polyhedra either **Platonic** solid or **Archimedean** solid.

To describe the generic metamorphosis idea, the figure V.8 is depicted. The figure as whole is like a right triangle and is actually the outputs of seed points of the seed point triangle accordingly. It is the metamorphosis of Oh symmetry type. The users of Tesellation3D application are encouraged to find the other metamorphoses for other type of symmetries. (O, T, Th, T and Td)

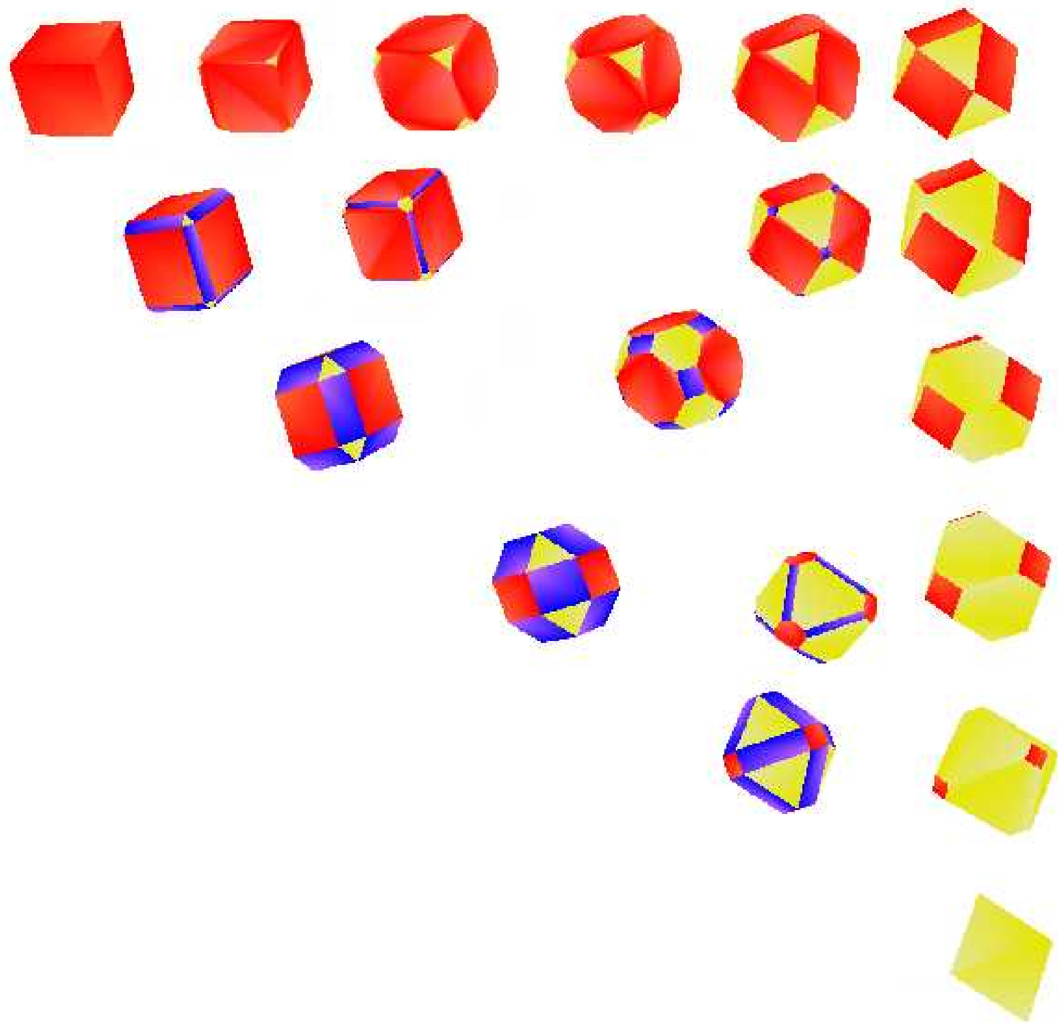


Figure V.8 Polyhedra metamorphosis with symmetry type Oh

V.3 Coloring Polyhedra

Coloring the polyhedra is crucial so that faces and edges should be easily recognized. Furthermore the metamorphosis of polyhedra should be tracked more easily with a good coloring of faces of polyhedra.

There are 2 face-coloring schemes that can be used for coloring the faces of polyhedra.

V.3.1 Coloring with number of Edges

This is a simple and straightforward coloring method, which is also used in this thesis and Tesellation3D application. A main color is given with respect to the number of edges at that edge. The main colors used for each edge number is as the following. These colors are preferered as they give better visualization.

- Edge Number = 3 then main Color = Black (0,0,0)
- Edge Number = 4 then main Color = Blue (0,0,255)
- Edge Number = 6 then main Color = Green (0,255,0)
- Edge Number = 8 then main Color = Red (255,0,0)

With cubic symmetry only 3,4,6 and 8-sided polygons (faces) can be created. Thus these 4 main colors are sufficient.

After main color is determined for a vertex a tone is given to that vertex to differentiate vertices. Each tone is added as RGB level (Red, Green, Blue) as with the following formulae.

```
0.4*edgeIndex+color.x,
0.2*edgeIndex+color.y,
0.1*edgeIndex+color.z);
```

Where edge index is incremental variable for each vertex (i.e. Square has 4 edges). And *color.x*, *color.y* and *color.z* stand for the main color of the vertices.

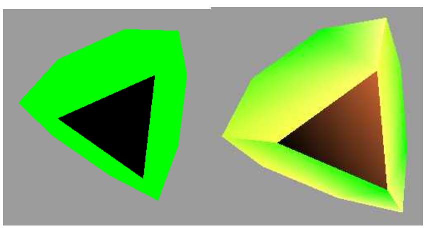


Figure V.9 Polyhedra without color tones(left), polyhedra with color tones(right).

As in figure V.9 (left) every pixel of the triangle is black. It is impossible to notice the edges of the truncated tetrahedron.

However in figure V.9 (right) it is easier to differentiate the faces. The triangle is blackish at the left most side and brownish at the right most and even lighter at the upper most side of the triangle.

V.3.2 Coloring with Face Types

While it is simple and straightforward to color faces with the number of edges at that face there is even a better coloring scheme especially for tracking the polyhedra metamorphosis.

With the "coloring with edge number" approach the following figure is obtained while tracking the metamorphosis of a polyhedra with symmetry type Oh.

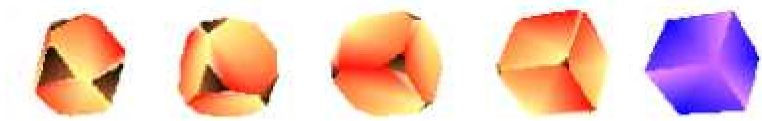


Figure V.10 A cubeoctahedron metamorphoses to a cube with "face's edge number" coloring scheme

As the user carries the metamorphosis it is noticeable that the red faces (octagons) are getting larges where as the brown faces (triangles) are getting smaller and smaller. Finally the triangles converges to a point and brown faces are totally disappeared also squares emerged instead of the octagons.

However it is somewhat deceiving for the last picture in figure V.10 is colored with blue even it has squares. Instead it should be with the same color (red) with the octagon just before it metamorphoses to a square as in the following figure.

Figure V.11 A cube octahedron metamorphoses to a cube with "face's type" coloring scheme

With this approach the continuity of colors is be achieved. The colors for faces are given depending on which kind of rotation axis the face lies. There are 3 types of rotation axis as stated in the previous chapters thus there are 3 kind of colors.

Face Color	Rotation Axis Description
Red	joins the centers of opposite faces (depicted with yellow axes)
Yellow	joins diagonally opposite vertices (depicted with light blue axes)
Blue	joins midpoints of opposite edges.(depicted with pink axes)

Table V.3 Three rotation axis types and the face colors that lies on them.

PART VI

CREATING A 3D VIRTUAL ENVIRONMENT WITH JAVA3D

VI.1 Introduction

When all the required information is obtained it is time to render the polyhedra Because polyhedron is a 3D object a 3D virtual world has to be created. Later the polyhedra should be rendered accordingly. This section briefly explains the creation of such a virtual environment and rendering procedure of the polyhedra using Java3D.

VI.2 Java 3D Graphics Engine

Java3D is a young yet promising Graphics Engine that is built upon the java programming language and runtime environment. It possesses essential tools and functions to create 3D environments; create/manipulate/destroy objects that are part of this environment. The java 3D includes but not limited to the following functions

1. Virtual world creation
2. Geometry object creation
3. User interaction management
4. Animation
5. Lights
6. Textures

In this thesis “virtual world creation”, “geometry object creation”, “user interaction” are extensively used and will be discussed in the following sections.

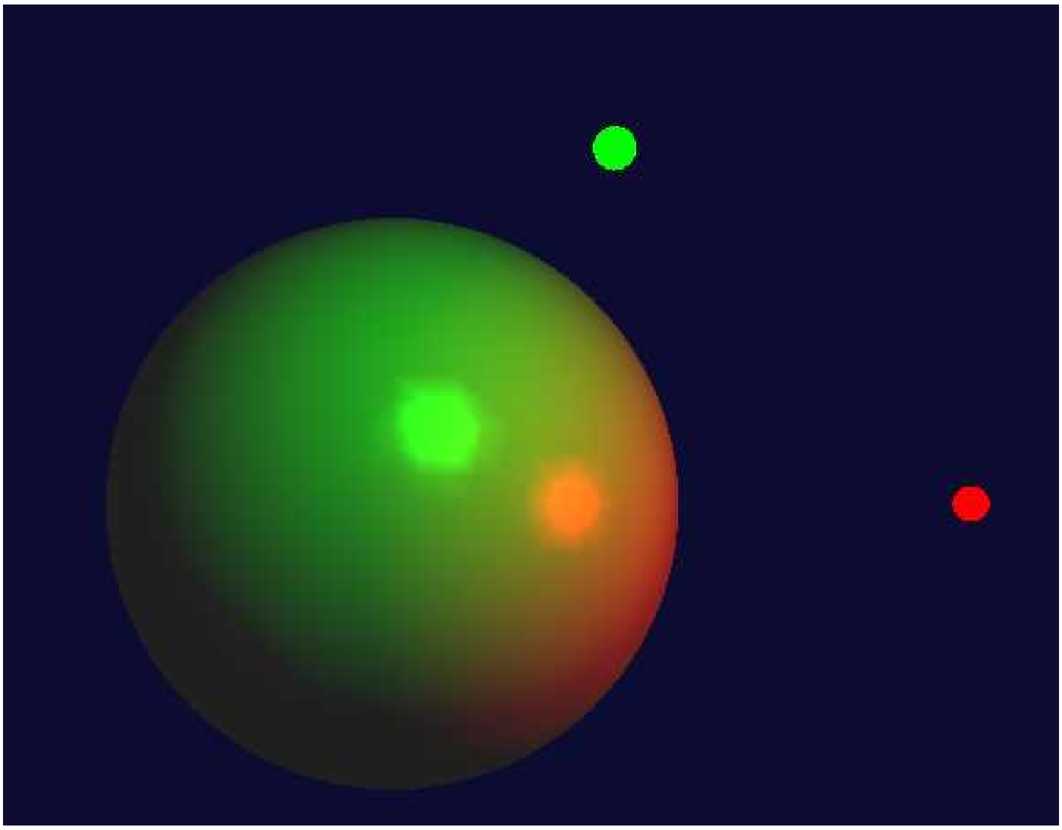


Figure VI.1 A Java3D application demonstrating the fundamentals of Java3D.

In figure VI.1 a sample Java3D application is depicted to demonstrate its basic features.

VI.3 Virtual World Creation with Java3D

The term virtual world refers to the three dimensional virtual space where objects populate.

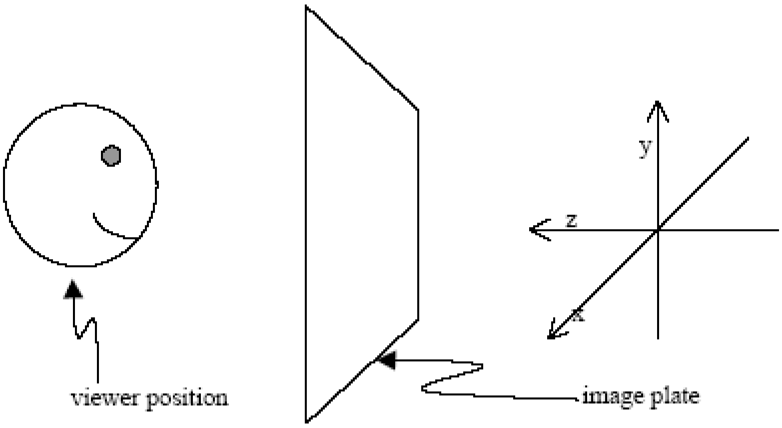


Figure VI.2 the viewer and the virtual 3D world.

In Java3D there are several ways to create a virtual 3D world. The easiest method, which is also used in this thesis, is via the following code segment.

SimpleUniverse su=new SimpleUniverse()

In this thesis only one polyhedron is drawn in this virtual world. Below is the coordinate system and a cube drawn on top of it.

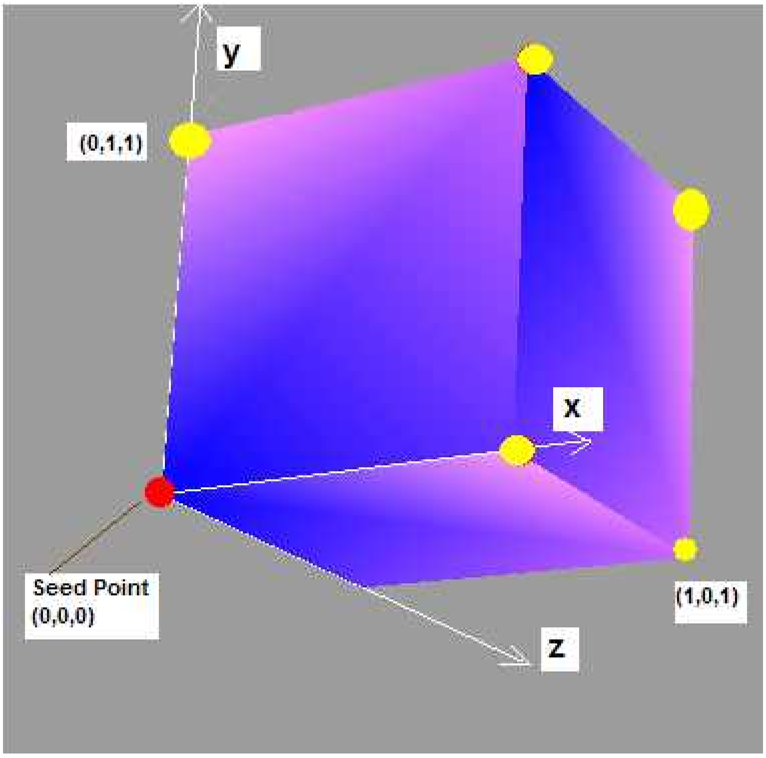


Figure VI.3 the coordinate system and the cube.

In figure VI.3 the seed point is represented with red where the symmetry points with yellow.

VI.4 Geometry Object Creation

The procedure of creating a geometry object (our polyhedron) involves the following steps.

1. Creating a Shape3D object (our polyhedra)
    a. Setting the vertices of the polyhedra
    b. Setting the colors of the vertices
    c. Defining the drawing type

2. Defining a behavior to the object for user interaction.
3. Add this Shape3D object to a branch group and compile scene.

VI.4.1 Creating a Shape3D object

Once we have found the vertices of the polyhedron and determined the faces by taking the convex hull we can create a Shape3D object that can display the polyhedra. In Java3D there are many ways to create a Shape3D object. The easiest one is *GeometryInfo* class. This class can represent a Shape3D object as an array of points and for each point there can be an associated color. Below is a code snippet to create a Shape3D object.

```
Shape3D polyhedron= new Shape3D()
GeometryInfo gi=new GeometryInfo ()
gi.setCoordinates ( polyhedronVertices)
gi.setColors ( polyhedronVerticeColors)
polyhedron.setGeometry ( gi )
```

The *polyhedronVertices* are an array of 3D Points, which is described in part 3. The *polyhedronVerticesColors* are each points color for each vertex, which is described in part 3 as well.

VI.4.2 Defining a behavior to the object

In Java3D a behavior is the motion of an object. The cause of this motion can be user input, a time interval and so on. In this thesis the users can interact with the polyhedron that is created in three ways as described in the following table.

MouseBehavior	**Action in Response to Mouse Action**	**Mouse Action**
MouseRotate	rotate visual object in place	left-button held with mouse movement
MouseTranslate	translate the visual object in a plane parallel to the image plate	right-button held with mouse movement
MouseZoom	translate the visual object in a plane orthogonal to the image plate	ALT key held with mouse movement

Table VI.4 The 3 interaction types to a polyhedra with mouse in the virtual world.

In Java3D to support all these mouse interaction there is a helper class called *OrbitBehaviour* the following is a code snippet to enable orbit behavior thus the mouse interaction.

```
OrbitBehavior orbit = new OrbitBehavior();
BoundingSphere bounds = new BoundingSphere(new Point3d(0,0,0),1);
orbit.setSchedulingBounds(bounds);
su.getViewingPlatfor.viewingPlatform.setViewPlatformBehavior(orbit);
```

VI.4.3 Creating a BranchGroup

In Java3D every object in the virtual universe must be inside a branch group Branch Group is like a container. It will apply operations to all objects it contains. In this thesis and the Tesellation3D application there is only one object (the polyhedron) thus the BranchGroup does not have much use. Once the Shape3D object is created. It can be connected to a branch group in the following way.

```
BranchGroup objRoot = new BranchGroup();
objRoot.addChild(polyhedron);
objRoot.compile();
```

Above the polyhedron that is created is connected to the BranchGroup. Finally compiler method of the BranchGroup is called. With this call the Java3D makes the required optimizations on the scene before drawing it. Finally the BranchGroup is connected to the Virtual world we have created at the beginning.

```
su.addBranchGroup( objRoot )
```

VI.5 Three Dimensional Geometry Structures

There are some fundamental three dimensional geometry operations and data structures used throughout the thesis and the Tesellation3D application. In this section their underlying mathematical formulas are given.

VI.5.1 Point, Line and Plane in 3D

A three dimensional point has 3 components x,y and z . The (0,0,0) point is referred as the origin point. An example point is as the following

P_1= (0.35, 0, 0.17)

Line is the shortest path between a point pair. A general line equation is as the following.

$$\frac{x - x_1}{x_1 - x_2} = \frac{y - y_1}{y_1 - y_2} = \frac{z - z_1}{z_1 - z_2}$$

Where point 1 is (x1,y1,z1) and point 2 (x2,y2,z2)

Plane is an unbounded two-dimensional shape, which slices the 3d space. For every point pair on this plane, the line that passes through these points also lies on this plane. The geometrical equation of a plane is as follows.

$$Ax + By + Cz + D = 0$$

VI.5.2 Rotation Axes for Cubic Symmetry

In this thesis there are totally 13 rotation axes to get rotational symmetries of a seed point.

Following is the table of rotation axes and are depicted in figure VI.4

Axis Type	**Axis Description**
Three axes of 4 fold symmetry	joins the centers of opposite faces (depicted with yellow axes)
Four axes of 3-fold symmetry	joins diagonally opposite vertices (depicted with light blue axes)
Six axes of 2-fold symmetry	joins midpoints of opposite edges.(depicted with

	pink axes)

Table VI.5 The rotation axes used for cubic symmetry.

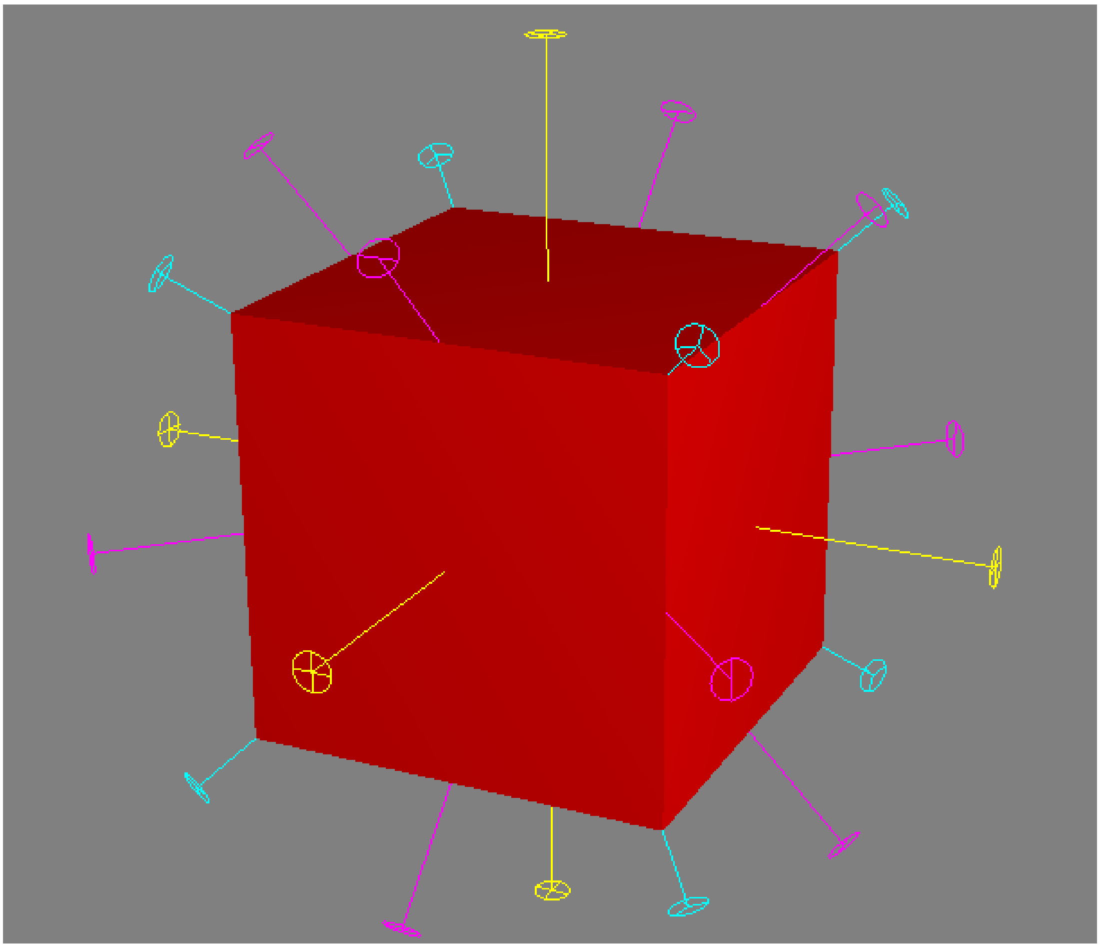


Figure VI.4 The thirteen rotation axes

VI.5.3 Reflection Planes for Cubic Symmetry

In this thesis there are 6 diagonal and 3 horizontal planes to make reflection symmetries of the seed point..(Refer to part II)

The 6 diagonal symmetry planes are as follows and are depicted in figure IV.5.

Plane Equation	**Plane Description**
x+y-1=0	diagonal, depicted as p1
x-y=0	diagonal, depicted as p2
y+z=-1=0	diagonal, depicted as p3
x+z-1=0	diagonal, depicted as p4
x-z=0	diagonal, depicted as p5

y-z=0	diagonal, depicted as p6

Table VI.6 The diagonal reflection planes used for cubic symmetry.

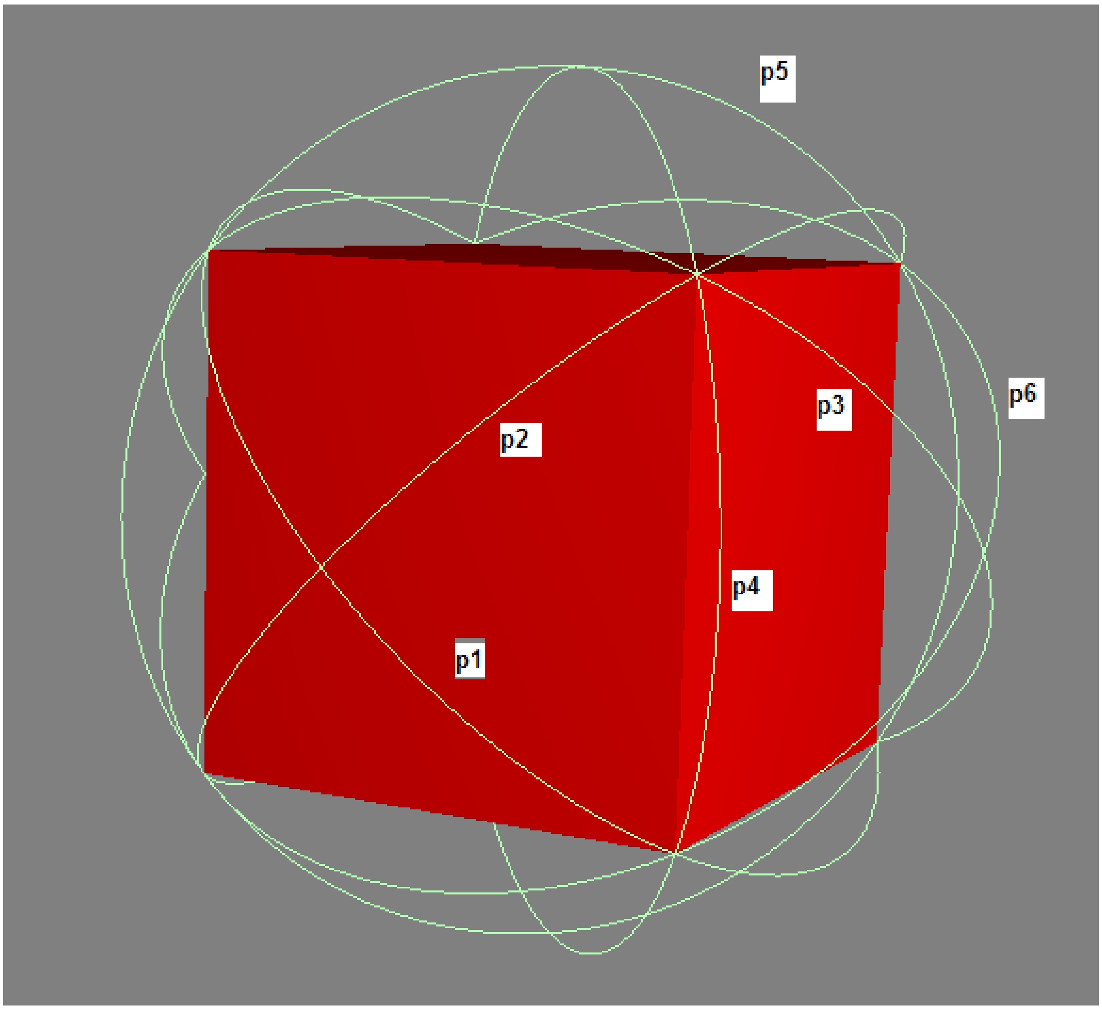


Figure VI.5 The six diagonal symmetry planes

The 3 horizontal symmetry planes are as follows and are depicted in figure VI.6.

Plane Equation	**Plane Description**
2x-1=0	horizontal, depicted as p7
2y-1	horizontal, depicted as p8
2z-1=0	horizontal depicted as p9

Table VI. 3 The horizontal planes used for reflection symmetry.

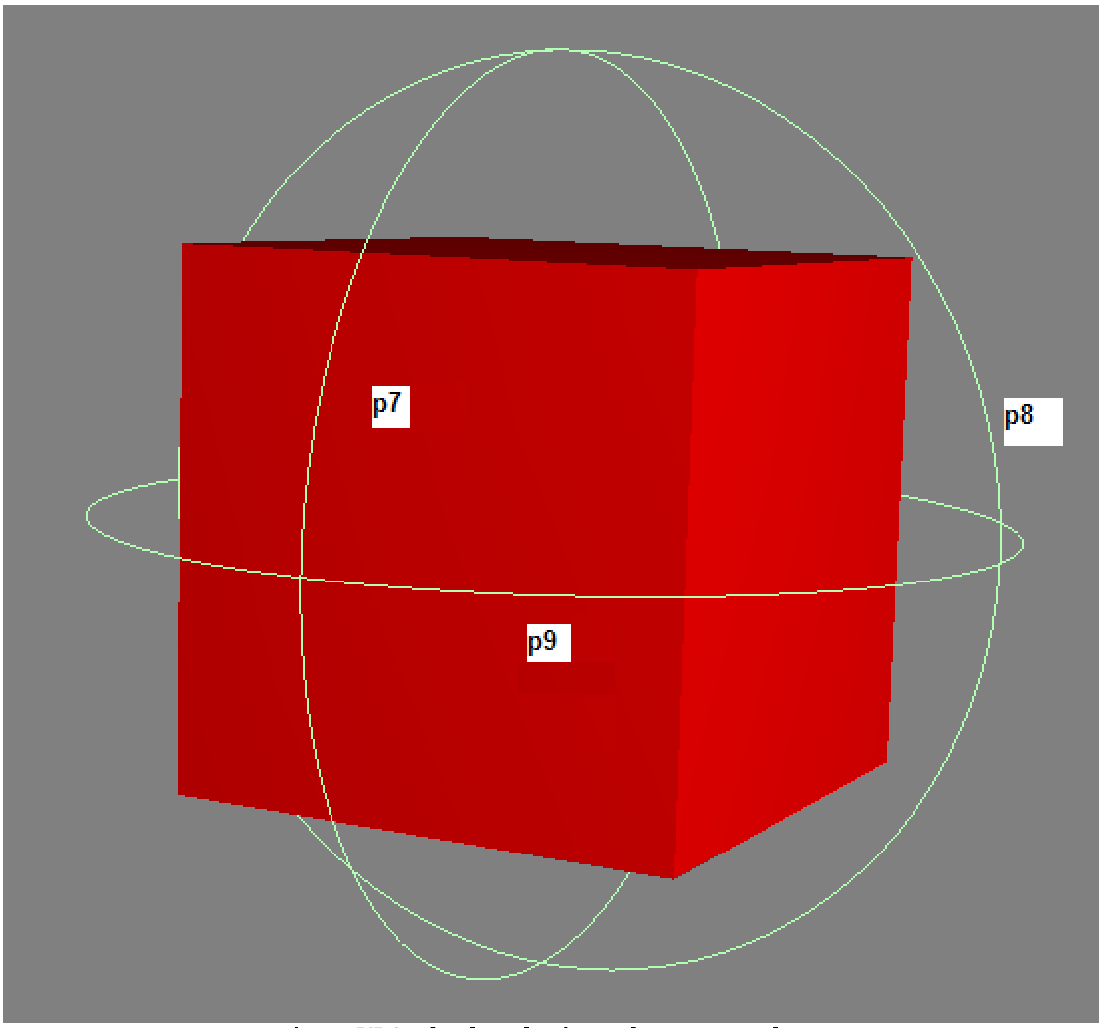


Figure VI.6 The three horizontal symmetry planes

VI.6 Three Dimensional Geometry Operations

There are six fundamental geometry operations that are used through out this thesis and they will be listed below however their implementation will not be given.

- Distance of a point to another point
- Distance of a point to a plane
- Rotating a point about an arbitrary axis.(for finding rotational symmetries)
- Finding mirror symmetry of point on a plane.(for finding reflection symmetries)
- Finding if an axis (line) pass through a face (polygon) lies
- Finding if a point lies within a face (polygon).

Thorough descriptions and algorithm details of these operations are available at the "Geometry Fundamentals" web page [FGO]

PART VII

RESULTS AND CONCLUSIONS

In this thesis the Cubic Symmetry type is used to create polyhedra metamorphosis in an interactive 3D environment. The results of this study and thesis are as follows.

1. Creating polyhedra using Cubic Symmetry is extensively studied and implemented. The idea of creating polyhedra from cubic symmetry is simple at the first glance however to implement it is a complex task. The procedures of such an application is outlined and implemented in this thesis.

2. An Interactive 3D environment is created for users to test, see and study 3D Polyhedra, with this application: Tessellation 3D, a user can visualize for example the tetrahedron far more easier than by imagining (without seeing) it.

PART VIII

DISCUSSIONS AND EVALUATIONS

VIII.1 Evaluation of Study

This study fills a gap between the theory of creating polyhedra using cubic symmetry and implementing it in a virtual environment.

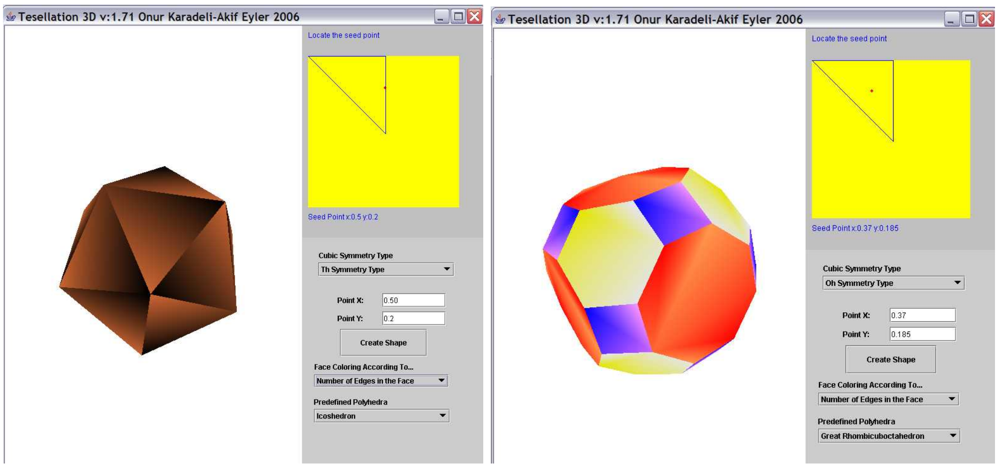


Figure VIII.1 Simulation with Tessellation 3D Application with Tessellation 3D Application

There are some subjects that can be further extended or improved related with this study and Tessellation 3D application.

- In this thesis the z coordinate in seed point is taken as 0. However making it parametrical would be an interesting subject.
- Creating Polyhedra Metamorphosis for other types of symmetries can be studied and implemented. (C,Ch,…)

VIII.2 How to benefit from this Study and Tessellation 3D

This study is informative and encouraging for anyone that is interested with Polyhedra. It is also much more easier to explore 3D Objects with the Tessellation 3D application.

With this study, anyone who is interested with Polyhedra is highly encouraged to model and implement 3D applications by supplying the working application, full source code and documentation.

Furthermore The Java 3D world is contributed a working application that be an example to others Developers and Students worldwide and aid to the development of the Java 3D Graphics Engine.

REFERENCES

Books

[PET] Peter R. Cromwell, *Polyhedra*, Cambridge University Press, ISBN: 0-521-66405-5 **(1997)**

[SHA] Alan Holde, *Shapes, Space and Symmetry*, Columbia University Press, ISBN: 0-486-26581-9 **(1971)**

[CHA] Michael Field, Martin Golubitsky, *Symmetry in Chaos*, Oxford University Press, ISBN: 0198536895 **(2005)**

[J3D] Daniel Salman, *Java 3D Programming*, Manning Publications, ISBN: 1-930110-35-9 **(2002)**

[ADJ] H.M. Deitel, P.J. Deitel, S.E. Santry , *Advanced Java 2 Platform: How to Program*, Prentice Hall Publishment, ISBN: 0-13-089560-1 **(2002)**

[GAM] David H. Eberly , *3D Game Engine Design*, Academic Press , ISBN: 1-55860-593-2 **(2001)**

Articles

[MET] Chao-Hung Lin, Tong-Yee Lee, "*Metamorphosis of 3D Polyhedral Models Using Progressive Connectivity Transformations*",IEEE Transactions on Visualization And Computer Graphics, January-February **(2005)** (Vol 11 No 1) pp 2-12

[STO] C.H.Ahn, S.Y.Lee, "*Regular Polyhedra for Storage Efficient Hybrid FE-BE Method*", Conf. Computation of Electromagnetic Fields, Sorrento, Italy, (**1991)**, 7

Internet Resources

[MAT] Mathworld Geometry and Polyhedra database, http://mathrowld.wolfram.com
Last accessed 10 Dec, **2005**

[CST] Peter Cromwell's website, http://liv.ac.uk/~spmr02/
Last accessed 10 Dec, **2005**

[GEO] Geometric Definitions Regular Polygon,
http://planetmath.org/encyclopedia/RegularPolygon.html
Last Accessed 30.12.2005

[TUT] Java3D Tutorial, http://java.sun.com/developer/onlineTraining/java3d/
Last Accessed 30.12.2005

[SYM] Types of symmetry operations, cogprints.org/3915/01/7clustersCog.pdf
Last Accessed 30.12.2005

[FGO] Fundamental 3D Geometry Operations,
http://astronomy.swin.edu.au/~pbourke/geometry/
Last Accessed 30.12.2005

[NAM] Polyhedra Naming: http://www.georgehart.com/virtual-polyhedra/naming.html
Last Accessed 30.12.2005

Tools

[STE] Great Stella Polyhedra Navigator application and documentation,
http://web.aanet.com.au/robertw/Stella.html

[STO] Java Convex Hull API by John E. Lloyd, www.cs.ubc.ca/~lloyd/

[PAI] Microsoft Paint Application

RESUME

Onur KARADELİ

Communication

tel

02164176127 – 0544 6743194

email

okaradeli@yahoo.com

address

T.C.K. 1. Bölge Loj. A2 Blok No:3

- Küçükyalı / İ stanbul/TURKEY PK: 34841

Work and Software Experiences

- **Cybersoft I.T. of Turkey (2003 Computerworld Honors award winner)** As a Software Engineer in **Cybersoft I.T.** have attended several large scale projects where the whole software lifecycle is used which includes **Sekerbank** Banking , **MKK Registration** , ECA-Emar online enterprise automation. Have created Web services for these applications and modeling (Rose) , documentation as well all by using the core Java Platform and J2EE methodology
(2004-2006)

- **Software Engineer in Koçbank I.T. department Special Projects group.** Responsibility to develop and maintain a wide range of **POS** applications and interfaces involving many programming languages (**Java/C++/Delphi/Linux Shell Script**) and DBMSes (**Oracle,SQLServer**)
(2002-2003 **1 year)**

Education

- **M.S.** Marmara Uni.Computer Science Engineering (**2002-...)**
  - Entered in **1st place**
- **B.S.** Marmara Uni. Computer Science Eng. (1998-2002)
  - Cumulative GPA: **3.15 / 4, Honor student**
  - **LES: 74.6 1 / 80**
  - Kayseri Science High School (KFL.)