1.5in 0.6in 1.0in 0.8in 20pt 0.25in 9pt 0.3in

UNIVERSITY OF BUEA

FACULTY OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

Implementation Of A Heart-shaped Primitive

A thesis submitted to the Faculty of Science
in partial fulfilment of the requirements for the degree of
Master of Science in Computer Science
By

Isaac Kamga Mkounga
(SC09B676)

Supervisor:
Prof. Emmanuel KAMGNIA

Date:
July 2015

Declaration

I, Isaac Kamga Mkounga, Matriculation Number (SC09B676) declare that this thesis titled, *Implementation Of A Heart-shaped Primitive* and the work presented in it are my own. I confirm that this work was done wholly while in candidature for a Master of Science in Computer Science at the University of Buea. Where I consulted the published work of others, this has been clearly acknowledged. This work has not been previously submitted for a degree or any other qualification at this University or any other institution.

Signed: ________________ Date: ________________

Isaac Kamga Mkounga
(SC09B676)

Dedication

To My Family

Certification

This is to certify that this research project titled *Implementation Of A Heart-shaped Primitive* is the original work of Isaac Kamga Mkounga, Matriculation Number (SC09B676), a Master of Science in Computer Science student of the Department of Computer Science in the Faculty of Science at the University of Buea.

Supervisor: ______________________ Date: ______________________

Prof. Emmanuel Kamgnia

HOD/CSC: ______________________ Date: ______________________

Dr Denis Nkweteyim

Dean/FS: ______________________ Date: ______________________

Prof Ayonghe Samuel

Acknowledgement

Firstly, I thank our Lord Jesus Christ for giving me life and good health during my stay at University Of Buea and for granting me the patience to go through this program.

I am grateful to the Vice Chancellor of the University Of Buea and her collaborators for maintaining an enabling environment where graduate students can learn and express their utmost research potential. I am especially indebted to Professor Joyce B.M. Endeley, currently the Director of the Higher Technical Teachers' Training College of the University of Buea in Kumba and Professor Theresa NkuoAkenji, the Deputy Vice Chancellor for Internal Control and Evaluation who authorized the connection of internet facilities to the Postgraduate Computer Science Laboratory during the implementation of this project. I wish to sincerely thank all the staff of the Department of Computer Science for their advice and stimulating conversations during the writing of this thesis. Special thanks go to Professor Emmanuel Kamgnia and Dr. Denis Nkweteyim for meticulously proofreading through and correcting this work while making their wide spectrum of research experience and books on graphics available to me.

My appreciation goes to the Google Open Source Programs Office (OSPO) – Mary Radomile, Carols Smith, Cat Allman and Stephanie Taylor and the United States Army Research Laboratory BRL-CAD team for their sponsorship of this project under the auspices of the 2013 Google Summer Of Code program. Special thanks go to Mr. Christopher Sean Morrison and Mr. Erik Greenwald for their valuable advice and mentorship during the implementation of this project. I am also indebted to *Nominal Animal* and the mathematician Dr. Titus Piezas for tips on ray-tracing the heart surface.

Many thanks go to my family for their assistance and encouragement throughout this program.This thesis is dedicated to them. I also thank the friends I made during the course of this program whether from University of Buea or the Developers community, for bringing all the fun they came along with– I appreciate your company and I'm happy to know I was not alone.

Abstract

In this thesis titled Implementation Of A Heart-shaped Primitive, we aimed at demonstrating the engineering of a heart-shaped primitive within BRL-CAD package.Using a case study approach, we designed the heart-shaped primitive's data structure, wrote necessary callback functions and tested them using BRL-CAD's testing infrastructure. We showed that the Laguerre-based root solver is indeed a sure-fire iterative method for finding roots of polynomials and ascertained it's stability on sextic equations. This work provides a guideline for the development of primitives within Computer-Aided Design (CAD) software by highlighting the implementation of geometrically-useful properties for any primitive within BRL-CAD.

Contents

List of Figures

List of Tables

Abbreviations

CAD	**C**omputer **A**ided **D**esign
BRL-CAD	**B**allistic **R**esearch **L**aboratory **C**omputer **A**ided **D**esign
CSG	**C**onstructive **S**olid **G**eometry
NURBS	**N**on-**U**niform **R**ational **B-S**plines
NMG	**N**on - **M**anifold **G**eometry

Chapter 1

INTRODUCTION

1.1 Historical Background

Throughout our long history, we humans have always sought for means to express our creativity–ways to communicate our ideas through writing, sculpting, painting, carving, architecture and drawing.As a matter of fact, paleolithic cave representations of animals at least 32,000 years ago in Southern France, ink drawings and paintings of human figures as well as writings in hieroglyphics on papyrus in the pyramids of ancient Egypt is indicative of the fact that our need to express our individuality goes back to antiquity.Before the renaissance, drawing was treated as a preparatory stage for painting and sculpting.The wide availability of drawing instruments such as pens and pencils and most especially paper made master draftsmen like Leonardo Da Vinci, Raphael and Michelangelo around the world to lift drawing to an art in its own right.Thus, drawing stood out as the most popular and fundamental means of public expression in human history and is one of the simplest and most efficient means of communicating visual ideas.[1]

The drawing board era where paper, pens, pencils, rulers and ink prevailed has been relegated to the background in this information age which is powered by ubiquitous computer technology. Craving the unity of science and art, this essentially binary-sequenced revolutionary device called the computer married the artistic and engineering forms of drawing into an androgynous one called ComputerAided Geometric Design or geometric modeling for short. Geometric modeling currently involves the use of computers to aid in the creation, manipulation, maintenance and analysis of representations of the geometric shapes of two and threedimensional objects [2]. It is the outgrowth of convergent motivations and developments from several works of life as outlined below.

- In the 1950s, the need to automate the engineering drawing process led to electronic drawings which could be archived and modified more easily, could be easily verified and errors could be eliminated from mechanical designs without introducing new ones.These computer drafting systems allowed designers to produce drawings of objects by projecting threedimensional objects unto twodimensional surfaces.
- Then in the 1960s, there was a pressing need for software in the automobile, shipbuilding and aircraft industries to produce computer-compatible descriptions of geometric shapes which can be machined from wood and steel into stamps and dies for the manufacturing and assembling of car parts,ship hulls as well as wings and fuselages using computer numerically controlled tools.
- Later in the 1970s, the growing need for computers to render realistic images of objects as well as animate solid objects pushed research institutes like Xerox Palo Alto Research Center (Xerox PARC) and Apple Computers to make significant contributions to graphical user interface design and computer graphics.

These needs and problems could only be solved by research in fields such as graphics, animation and applications from algebraic geometry. The work of various computer scientists and mathematicians lead to the active development of several commercial packages sponsored by companies such as

Renault, Citroen, Ford and Boeing who could afford the computers capable of performing such lengthy calculations.

Today,geometric modeling is also referred to as Computer Aided Design (CAD), is pronounced “cad” and is routinely used in the design and manufacturing of engineering and architectural structures such as buildings, car parts, ship hulls and aircraft artillery as well as to specify special effects in cartoon movies, music videos and television shows. Indeed, CAD packages provide facilities for designing shapes of solid physical objects and specifying their motion in a way that art and science can unite to create cool designs.

Even though significant progress had been made in basic research and the functionality of commercially available solid modelers like Apple Computer’s RenderMan, many solid modelers especially within the open source community are still limited in their geometric features. The open source community is a selforganizing collaborative social network of programmers driven by a passion to solve problems using computers. It has several thousands of its projects on sites that offer services like bug tracking, mailing lists and version control viz Github and Source Forge.These projects are constantly being improved upon by thousands of programmers putting in time and effort to write and debug software without direct monetary pay.

In this thesis, we document the process of developing a heart-shaped primitive, a set of callback functions and procedures which compute geometrically useful properties of solids such as wireframe plotting, database importation and exportation,ray tracing, bounding box calculations, just to name a few, within the Ballistic Research Laboratory Computer Aided Design (BRL-CAD) software package.

BRL-CAD was initiated by the United States Army Research Laboratory in 1983, the same agency which created the E.N.I.A.C., the world’s first general purpose computer in the 1940s, to model military systems for the United States government. According to [3], BRL-CAD became born again in 2004 when it joined the open source community with portions of its source code licensed under the Lesser General Public License (LGPL) and Berkeley

Software Distributions (BSD) licenses and has been credited as being the oldest open source repository in continuous development. It supports a wide variety of geometric representations including an extensive set of traditional implicit primitive shapes as well as explicit primitives made from collections of uniform Bspline surfaces, Nonuniform Rational Bspline (NURBS) surfaces, Nonmanifold geometry (NMG) and purely faceted polygonal mesh geometry.

BRL-CAD also focuses on solid modeling aspects of Computer Aided Design. Figure 1.1 below shows a threedimensional model of a Goliath tracked mine, a German engineered remote controlled vehicle used during World War II. This model was created by students new to BRL-CAD in the span of about 2 weeks, starting from actual measurements in a museum.

1.2 Importance Of This Work

This work is significant to several stakeholders for several reasons ;

- By raytracing the heart's surface, it demonstrates to the scientific community that the Laguerre zerofinder is indeed a surefire iterative method for finding roots of polynomials and that Laguerre-based root solvers are stable on sextic equations.
- This work incorporates more geometric modeling functionality into the free and open source software community through BRL-CAD, the oldest open source repository in continuous development[3] by going beyond traditional CSG primitives shapes such as tori, spheres, boxes and ellipsoids towards the more complex heart-shape based on a sextic equation. This work provides a guideline for the development of primitives within open source CAD software.
- Given that BRL-CAD is used within governments to model military artillery and for engineering and analysis purposes within academia, this heart-shaped primitive gives BRL-CAD a more loving aura – an

environment where artists can produce cartoon animations as well as design cards, royal seals and banners, gifts and presents for family and communal celebrations such as weddings, family reunions and Valentine's day for entertainment purposes.

1.3 Thesis Organisation

This thesis is divided into five (5) chapters. Chapter 1 introduces the study and chapter 2 reviews the literature in the field of geometric modeling. In Chapter 3, we state the problems we intend to solve and our project design. In chapter 4, we discuss the interesting results which we obtained. Finally, in chapter 5, we state the contribution of our work and give possible research directions which can proceed from it.

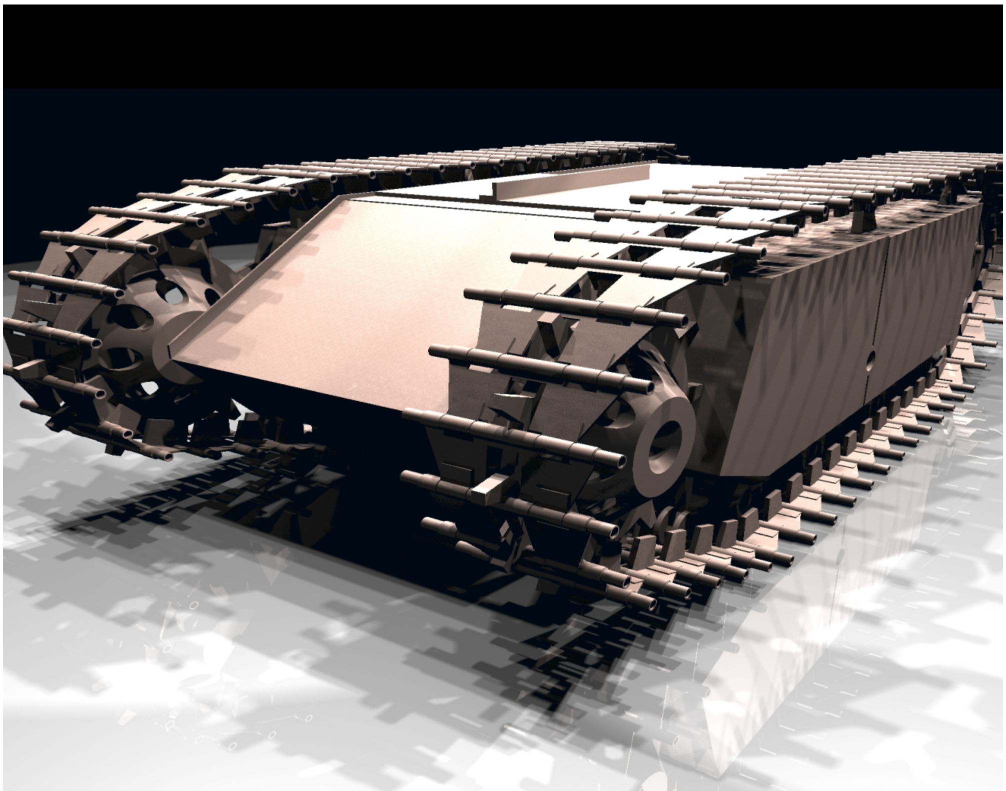

FIGURE 1.1: Model of a Goliath tracked mine

Chapter 2

LITERATURE REVIEW OF GEOMETRIC MODELING

With the advent of computers which could perform millions of floating point operations in unit time and which are still growing faster, researchers who believed computers could aid the processes of mechanical design and manufacturing were faced with a critical issue – how to represent physical reality using computer software. They sought the best data structures to represent this reality and the most appropriate algorithms to manipulate these representations.

BRL-CAD supports a wide variety of geometric representations including an extensive set of traditional implicit primitive shapes as well as explicit primitives made from collections of uniform Bspline surfaces, nonuniform rational Bspline (NURBS) surfaces, non-manifold geometry (NMG) and purely faceted polygonal mesh geometry. Consequently, in this chapter, we review the existing work done by scholars in the field of geometric modeling which have been applied to the development of BRL-CAD. First of all, it introduces the issue of representation and the notion of representation schemes.Then, it summarizes developments in wireframe modeling, surface modeling, solid modeling and non-manifold modeling (aka nonmanifold geometry or nmg for short) with a keen eye on the algorithms underlying them.

As we progress in our literature review from older forms of geometric modeling to newer ones, we will discover that representation schemes were closely linked to algorithmic efficiency and that it has always been normal to expect designers to switch to newer ones in response to the improvements in algorithmic performance.Despite these enhancements in algorithmic efficiency within the designer community, we cannot say with complete certainty whether traditional representation schemes can be relegated to the background. We can only conclude that old and new representation paradigms coexist and that research led to representation schemes which supplemented the repertoire of geometric modeling.

2.1 Representation Schemes

A representation $\mathbf{R}$ of a solid or representation for short is a subset of three-dimensional Euclidean space denoted $\mathbb{E}^3$ which models a physical solid. According to [5], Requicha and Tilove stated that point set topology provided a formal language for describing the geometric properties of solids and they also threw more light on the mathematical characteristics of solids such as a solid's interior, boundary, complement, closure, boundedness and regularity. Requicha [4] insisted that to be computationally useful, a representation should formally capture the following properties ;

- ***Rigidity:*** Representations should have an invariant configuration irrespective of their location and orientation.
- ***Homogeneity:*** A representation should have an interior.
- ***Finiteness:*** A representation must occupy a finite amount of space.
- ***Boundary determinism:*** A representation must unambiguously determine the interior of that solid.

- ***Closure***: Representations of solids which are manipulated by rigid motions and regularized boolean operations should produce representations of solids too.

These formal characteristics leave representations no choice than to be bounded, closed, regularized and semianalytic, hence their coinage rsets according to [5]. An ***r-set*** is simply a regular and bounded set in $\mathbb{E}^3$.

A representation scheme is simply a relation between physical solids and their representations which can be characterized by the following properties;

- ***Domain***: A representation scheme must represent quite a number of useful geometric solids.
- ***Unambiguity***: A representation scheme should produce representations which intuitively capture the properties of the physical solid so that it can be easily distinguished from other representations.
- ***Uniqueness***: A representation scheme should uniquely represent a solid object within a software's database.
- ***Validity***: Representation schemes should yield representations of solids which do not exist or are valid.
- ***Closure***: A Representation scheme which transforms (reflects, scales, rotates) a representation should yield other representations too.
- ***Compactness***: Representation schemes should yield representations which save space and allow efficient algorithms to determine desirable physical characteristics.

2.2 Wireframe Modeling

For rectilinear objects whose edges are straight lines and whose faces are planar, the ordered pair of vertices $\mathbf{V} \in \mathbb{E}^3$ and edges $\mathbf{E} \in \mathbb{E}^3$ denoted by $(\mathbf{V}, \mathbf{E})$ is the object's wireframe. In a practical sense, it is the skeleton of an object wherein joints are vertices and bones are edges. In [6], a six step algorithm to generate an object's wireframe was developed wherein an object's wireframe was represented by a vertex table and an edge table. Although the work in [6] had drawbacks such as not checking the validity of input data, wireframe modeling has always provided designers with a chance to experiment with the final result of their models through sketching and it is frequently used to preview complex models. However, the use of only edge information left wireframe models ambiguous on rectilinear polyhedra talk less of topological ones. Figure 2.1 below shows the wireframe of a sphere in greyscale.

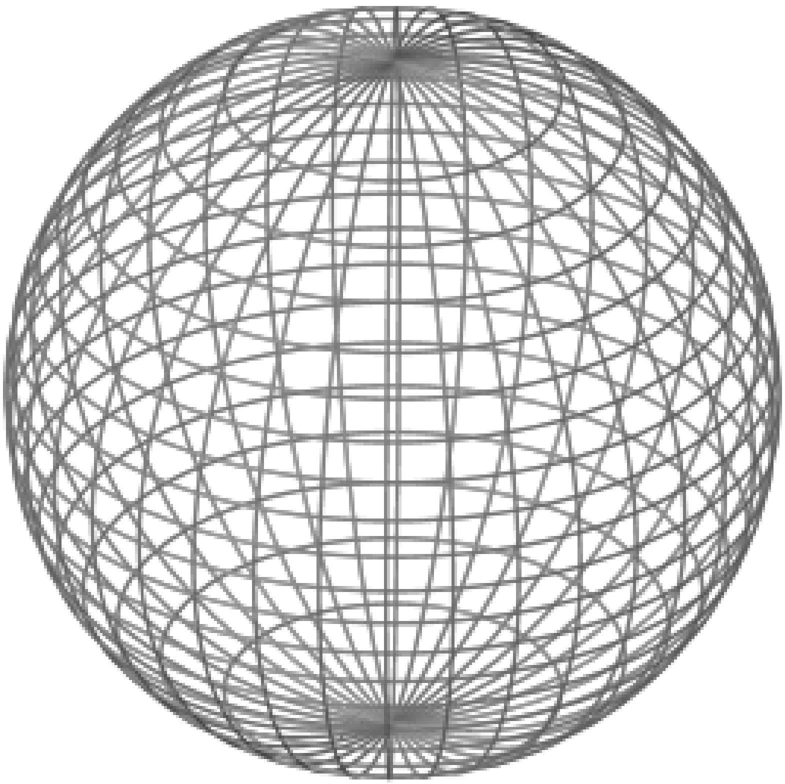

FIGURE 2.1: A wireframe of a sphere

2.3 Surface Modeling

After breakthroughs in wireframe modeling, research efforts in geometric modeling were directed towards extending the geometric coverage of CAD packages by incorporating complex freeform surfaces and curves. In this section, we emphasize on algebraic surfaces and curves used within BRL-CAD as it is the basis for Bezier surfaces and NURBS.

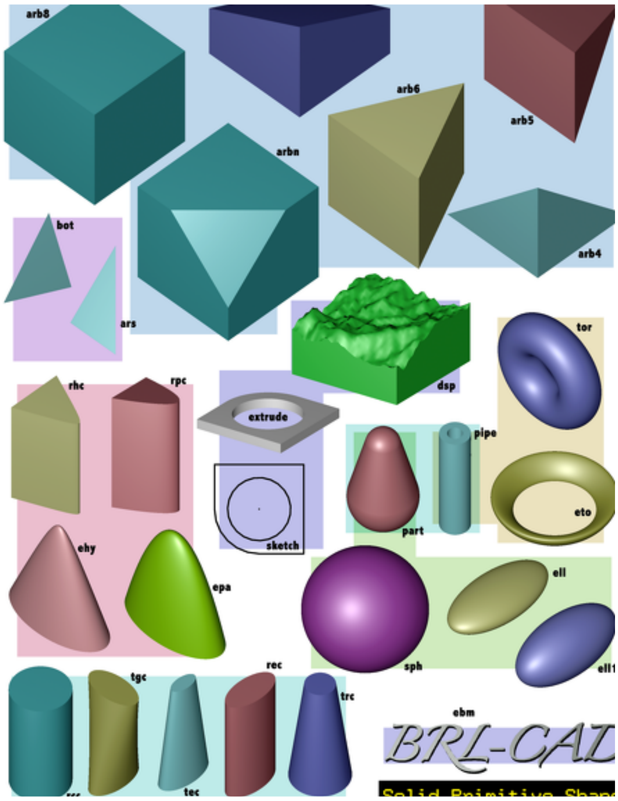

FIGURE 2.2: BRLCAD Solid Primitive Shapes

Figure 2.2 above shows a collection of some primitives used within the BRL-CAD package before the heart-shaped primitive was developed – several of which are implicitly and/or parameterically represented by algebraic equations.On the BRL-CAD's ideas page[7], there is a list of primitives which have not yet been implemented such as the Steiner surface, the ring cyclide surface, the quartoid, Wallis' conical edge solid, etc.

2.3.1 Implicit Representation

An algebraic surface in $\mathbb{E}^3$ is expressed as the set of points satisfying an irreducible polynomial equation

$$\mathbf{g(x,y,z) = 0}$$

in the unknowns x,y and z.
A polynomial ***f(x,y,z)*** over a field **F** is said to be irreducible over **F** if the degree of ***f(x,y,z)*** is positive and its only factors are c and *cf(x,y,z)* where c is a nonzero constant in **F**.

The requirement of irreducibility is so that a surface represented by an equation should not be decomposed into two separate surfaces, each of which can be described by an implicit equation.

2.3.2 Parametric Representation

Some algebraic surfaces possess a parametric representation which consists of a system of equations similar to the ones listed in (1) below;

$\mathrm{x} = \mathrm{h}_1(u, v)$
$y = h_2(u, v)$
$z = h_3(u, v)$
where h_i are rational functions and, u and v are restricted to particular closed intervals in $\mathbb{R}$.

As an example, the unit sphere given implicitly by $x^2 + y^2 + z^2 – 1 = 0$ can be parameterized by equation (2) viz

$\mathrm{x} = (1 - \mathrm{s}^2 - t^2)/(1 + s^2 + t^2)$
$y = 2s/(1 + s^2 + t^2)$
$z = 2t/(1 + s^2 + t^2)$

Also, some algebraic curves possess parametric forms. A parameterization of the unit circle is given by the system of equations in (3) below;

$\mathrm{x} = (1 - \mathrm{t}^2)/(1 + t^2)$
$y = 2t/(1 + t^2)$

When the parametric representation is employed, it is easier to generate points on an algebraic surface or curve as compared to the implicit representation. Also, parametric equations are useful for interactive design since changes in their polynomial coefficients alter the surface's shape in an intuitive manner.

Lots of geometric operations could become faster if both aforementioned representations are made available within CAD packages. Thus, the problem of how to convert from one representation to the other is of great practical importance.

2.3.3 Implicitization

Implicitization is the process of converting a parametric representation into an implicit representation.

Sederberg[7] demonstrated in his thesis that, in principle, it is always possible to convert a parametric surface or curve into implicit form using classical elimination theory developed in the early 20^{th} century. In fact, Sylvester resultants require evaluating a determinant whose entries are coefficients of powers of the variable to be eliminated in several phases.

Although Sederberg's work stimulated lots of research interests in resultant-based methods, this sense of enjoyment within the research community was shortlived due to the following factors;

- Unfaithfulness: Polynomials derived using resultant-based methods give birth to phantom solutions.

- Ineffectiveness: Using floating point arithmetic, resultant-based methods become inaccurate.
- Inefficiency: Evaluating resultants entail huge amounts of computation and is expensive.

Another method of Implicitization is the Grobner basis technique introduced by Buchberger[8] where it was learned that expressing a polynomial as a linear combination of a Grobner basis facilitates finding the solution of the nonlinear system of equations as much as an **LU** - decomposition brings a system of linear equations to heel. A polynomial basis is a set of polynomials which can be used to express any polynomial and can be viewed as a vector space over the field of coefficients $\mathbb{F}$. In [9], Lazard constructed a Grobner basis with respect to a term ordering known as the elimination order. In [10], Hoffmann improved upon a basis conversion algorithm developed in [11] by first constructing a Grobner basis with respect to a term ordering different from the elimination order and finally built the final polynomial in which all variables have been eliminated term by term. This algorithm was known to be the fastest elimination technique for geometry applications that was implemented before the 1990s.

Although Grobner basis methods are more efficient and effective than resultant-based ones, implicitization is fairly expensive and limited in practice.

2.3.4 Parameterization

Parameterization is the process of converting an implicit representation of an object into its parametric equivalent, if it exists. Parameterization is not always possible since not all implicit surfaces can be expressed as rational parameterizations. According to Noether's theorem, a plane algebraic curve possesses a rational parameterization if and only if it has genus zero. [13] used a numerically stable Jacobi rotation adapted from [12] to parameterize several conics and parametric surfaces. Table 2.1 below shows a list of parameterizations of some popular conics – circle, ellipse, hyperbola and parabola.

TABLE 2.1: Implicit and parametric equations of some BRL-CAD primitives

	Implicit Form	Parametric Form
Circle	$x^2 + y^2 - r^2 = 0$	$x = \frac{r(1-t^2)}{(1+t^2)}$, $y = \frac{2rt}{(1+t^2)}$
Ellipse	$\frac{x^2}{a^2} + \frac{y^2}{b^2} - 1 = 0$	$x = \frac{a(1-t^2)}{(1+t^2)}$, $y = \frac{2bt}{(1+t^2)}$
Hyperbola	$\frac{x^2}{a^2} - \frac{y^2}{b^2} - 1 = 0$	$x = \frac{a(1+t^2)}{(1-t^2)}$, $y = \frac{2bt}{(1-t^2)}$
Parabola	$y^2 - 2px = 0$	$x = \frac{t^2}{2p}$, $y = t$

Appendix A

Appendix Title Here

Write your Appendix content here.