Ray-Quadrics

Ergun Akleman
Visualization Laboratory,
216 Langford Center,
Texas A&M University,
College Station, Texas 77843-3137.
Phones: (409)-845-6599 (o), 845-3465 (l), 696-4445 (h)
E-mail: ergun@viz.tamu.edu, Fax: (409)-862-1571

Abstract

In this paper, we introduce the theory of ray-quadrics. Ray-quadrics are capable of providing intuitive shape design, fast rendering, and a wide variety of shapes from human heads to solids with any finite number of holes.

Ray-quadrics are non-polynomials; however, their intersections with the parametric equations of rays starting from the origin simplify to quadrics. This simplification makes real-time rendering and interactive shape design possible.

In order to construct ray-quadrics formulas which are meaningful for solid modeling, we introduce a new set of functional operators:

- Set-difference. This operation gives exact and approximate set-difference of star solids. Approximate set-difference smooths out the sharp edges and corners that result from exact set-difference operations.
- Border-intersection and flesh. The border-intersection operation gives the intersection of the outer surfaces of two ray-linearly represented star solids. The result is space curves. The flesh operator creates a flesh around these space curves, so that the space curves serve as a skeleton.

By using these operators over the the formulas that represent star solids with the same center, we are able to construct ray-quadric representations of toroids and solids with many holes.

Since they provide a large variety of shapes, intuitive shape construction, real-time rendering and interactive shape design, ray-quadrics are suitable for use as building blocks in implicit-equation-based modeling tools such as Constructive Solid Geometry and soft objects.

CR Categories and Subject Descriptors: I.3.7 [Computer Graphics]: Three Dimensional Graphics and Realism, I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling.

Additional Keywords and Phrases: Interactive Sculpting, Implicit and Parametric Representations, Solid Modeling.

1 Introduction

In our experience, to be effective and useful in a wide variety of applications, a mathematical tool for shape (curve, surface or solid) modeling should possess the following properties:

1. *Design:* The user should be able to design a shape by using simple and intuitive operations.
2. *Control:* The user should be able to change the shape by using a control shape.
3. *Computation:* The computation of the designed shape should be simple and fast.
4. *Organic Appearance:* The user should be able to generate an organic appearance.

In computer graphics, most widely used mathematical tools for modeling are based on either parametric or implicit representations. These two representations possess distinctly different modeling properties:

1. *Design:* Implicit representations inherently provide a simple implementation of geometric operations such as union and intersection by construction of formulas. This property of implicit forms allows simple and intuitive design by exact or approximate set operations over volumetric shapes.
2. *Control:* All parametric representation based methods, such as Bezier, B-Spline or NURBs, use control points to describe and manipulate a shape. Although control points are essential to describe fine details, so far no implicit representation based method exists that provides interactive construction and manipulation with *control shapes*. Witkin and Heckbert suggested an alternative control paradigm based on particles [WH94].
3. *Computation:* Parametric representations inherently provide easy surface sampling for rendering by just substituting parameter values, whereas, sampling an implicit surface is a root-finding process and, in general, can be quite difficult.
4. *Organic Appearance:* Implicit representations easily provide blobby appearances and bulges [Blo91]. Since

blobby appearances and bulges depends on underlying fine details similar to flesh of biological objects which depends on underlying fine details such as bone structure, the blobby appearances and bulges can give the look of biological objects. In fact, the implicitly based modeling tools that give control of blobbyness such as Wyvill's soft objects [WT90] or Blinn's exponential functions [Bli82] have long been successful in modeling organic looking shapes. In general, parametric representations are not well suited to creating an organic appearance.

However, by viewing flesh as a low frequency detail component that can be shaped independently, hierarchical approaches can be successful in facial modeling. Forsey and Bartels introduced a hierachical method, hierachical B-splines, that provides control of different levels of details [FB88] and Forsey used hierarchical B-splines to model faces.

1.1 Ray-Quadrics as a Dual-Representation

A modeling tool that capitalized on the different strenghts of both implicit and parametric techniques would greatly extend the power of geometric modeling tools. Such a tool could be used as a building block for implicit representations because of its implicit properties, while also providing fast and simple rendering because of its parametric properties. In fact, quadrics [Bli82] and superquadrics [Bar81], both of which are popular mathematical tools in computer graphics, provide both representations. In this paper, we introduce another mathematical tool that has such a dual nature: ray-quadrics. Ray-quadrics, like hyperquadrics [Han88], are a superset of superquadrics. Whereas hyperquadrics generalize superquadrics by going to higher-dimensions, ray-quadrics arise by introducing non-homogenous functions. But, as we shall see, they behave like homogenous functions in a sense that will become clear.

The modeling properties of ray-quadrics can be summarized as follows:

1. *Design:* A subset of ray-quadrics can be constructed by using geometric operators over ray-linears [Akl93], [Akl96]. This new set of operators provides exact and approximate set difference, border-intersection and flesh operation. By using these operators over the ray-linear formulas of star shapes with the same center, ray-quadric representations of toroids and shapes with many holes can be constructed.

   Ray-linear representations of half-spaces that include the origin provide the simplest construction primitives (building blocks). By using the approximate intersection and union operators of Ricci [Ric73] ray-linear representations of convex and star shapes with the same center can be constructed from ray-linear formulas of half-spaces. The approximate union and intersection operations can be considered global blending operations. These operations over non-negative ray-linear representations of half-spaces are closely related to the super-elliptic local blending of Rockwood [RO87].

   As a result, ray-quadric representations provide simple and intuitive operations.

2. *Control:* Exact set operations over half-spaces result in control shapes for ray-quadrics. Since the control shapes we use for ray-linears are star polyhedra, the control shapes for ray-quadrics would be two star polyhedra with the same center.

3. *Computation:* Ray-quadrics are similar to ray-linears: their intersections with a parametric equation of a ray that starts from the origin simplify to quadrics. In fact, this simplification phenomenon suggests the name *ray-quadric.* Because of this simplification, ray-quadric implicit forms can easily be parameterized like ray-linears. Therefore, once the related parametric equations are obtained, computing the shapes is simply the evaluation of the related parametric equations.

4. *Organic Appearance:* Control of organic appearance is accomplished by changing blending parameters. These blending parameters smooth out the sharp edges and corners resulting from exact set operations. There are four types of blending parameters: global (union), local (intersection), set-difference and flesh. The global blending parameter smooths out sharp edges resulting from the exact union of convex polyhedra. Blobby effects come mostly from this global blending parameter. Local blending parameters smooth out sharp edges and corners of convex polyhedra that result from the intersection of half-spaces. Different combinations of global and local blending parameters create different looks. Generally speaking, we can say that more blended shapes look fleshier, whereas less blended shapes look more robotic and less organic. The set difference operator smooths out sharp edges resulting from the exact set-difference of two star polyhedra. The flesh operator creates a flesh around the space curves generated by the border-intersection operator.

A shape that is describable using a ray-quadric formula (a ray-quadric shape) is restricted in the following sense: it must have a center such that each ray originating from this point intersects the shape at at most two points as shown in Figure 1. To construct other types of shapes, ray-quadrics

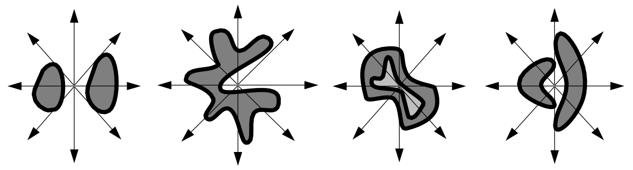

Figure 1: Examples of ray-quadric shapes.

can be used as building blocks in implicit representation based modeling tools such as constructive solid geometry [RV82], Ricci's constructive geometry [Ric73], soft objects

[WT90] or Blinn's exponential functions [Bli90]. Since ray-quadrics are not polynomials, they are not compatible with polynomial based implicit methods [Sed85], [SN86], [BI92].

The remainder of this paper is organized as follows. In the next section, we provide the definition of ray-quadrics. Based on the definition of ray-quadrics we describe, in Section 3, the parameterization of ray-quadric implicit forms and develop the concept of a guide shape. Section 4 explains how to construct convex and star shapes by using non-negative ray-linears. In Section 5, we introduce symmetric set difference, set difference and border-intersection and flesh operations and show how to construct ray-quadric shapes by using these operations. In Section 6, the properties of the ray-constant part are discussed. Finally, a conclusion is given in Section 7.

2 Definition of Ray Quadrics

Let $\mathcal{V}$ be a vector space. We use the letter $\mathbf{v}$ to stand for an element of $\mathcal{V}$ and $\mathbf{0}$ for the zero vector. In two-dimensional space $\mathbf{v} = [x, y]$, in three-dimensional space $\mathbf{v} = [x, y, z]$. Further, let $\Re^+$ denote the set of non-negative real numbers. We use the variable t to stand for an element of $\Re^+$. For the ease of formulation, we call in the following these elements the *positive* reals, which, however, should be understood to include zero. A value of α, however, will denote a truly positive real, i.e., $\alpha > 0$.

Definition 1 *A function $F : \mathcal{V} \longrightarrow \Re$ is a* ray-quadric *iff it has the form $F(\mathbf{v}) = A(\mathbf{v}) + B(\mathbf{v}) + C(\mathbf{v})$*
$\forall \mathbf{v} \in \mathcal{V}$ in which $A, B, C : \mathcal{V} \longrightarrow \Re$ satisfy
$A(\mathbf{v}t) = A(\mathbf{v})\, t^2$, $B(\mathbf{v}t) = B(\mathbf{v})\, t$ and $C(\mathbf{v}t) = C(\mathbf{v})$
$\forall \mathbf{v} \in \mathcal{V}$ and $\forall t \in \Re^+$, The class of ray-affines *are obtained by taking $A(\mathbf{v}) = 0$ for all $\mathbf{v}$. $B(\mathbf{v})$ is called a ray-linear and $C(\mathbf{v})$ is a ray-constant. A, B and C, in general, will be called ray-polynomials.*

$A(\mathbf{v})$, $B(\mathbf{v})$ and $C(\mathbf{v})$ are like homogenous functions. However, unlike the homogenous case, t is restricted to positive reals for ray-polynomials. Note that because of this restriction any homogenous function is a ray-polynomial, but the opposite is not true. Ray-polynomials accept absolute value; for instance, $|\mathbf{v}_0 \bullet \mathbf{v}|$ is a ray-linear and $\mathbf{v}_0 \bullet \mathbf{v} |\mathbf{v}_0 \bullet \mathbf{v}|$ is a second-degree ray-polynomial where $\mathbf{v}_0$ is any constant vector such as $\mathbf{v}_0 = [1, 0, 0]$. The inclusion of absolute value is a crucial step for the construction of formulas. The approximate set operations of Constructive Geometry of Ricci [Ric73] need non-negative functions, and the absolute value lets us construct non-negative functions.

2.1 Ray-Quadric Shapes

Definition 2 A *ray-quadricly represented shape* is one that is described by the *ray-quadric implicit representation*

$$S = \{\mathbf{v} \mid F(\mathbf{v}) = A(\mathbf{v}) + B(\mathbf{v}) + C(\mathbf{v}) \le 0\}, \qquad (1)$$

where $F(\mathbf{v})$ is ray-quadric.

Since $\mathbf{v} = \mathbf{v}_1 t$ is an equation of a ray originating from the zero vector, $\mathbf{0}$, in the direction of $\mathbf{v}_1$,

$$S(\mathbf{v}_1) = \{\mathbf{v} = \mathbf{v}_1 t \mid A(\mathbf{v}_1 t) + B(\mathbf{v}_1 t) + C(\mathbf{v}_1 t) \le 0\}$$

gives the intersection of the ray and the shape S. Because of the ray-quadric property of $F(\mathbf{v})$, this inequality simplifies to a univariate quadric inequality

$$S(\mathbf{v}_1) = \{\mathbf{v} = \mathbf{v}_1 t \mid A(\mathbf{v}_1) t^2 + B(\mathbf{v}_1) t + C(\mathbf{v}_1) \le 0\}.$$

Thus, the equation

$$A(\mathbf{v}_1) t^2 + B(\mathbf{v}_1) t + C(\mathbf{v}_1) = 0$$

gives the border of $S(\mathbf{v}_1)$ whose solution for t is

$$t_{1,2} = \frac{-B(\mathbf{v}_1) \mp \sqrt{B(\mathbf{v}_1)B(\mathbf{v}_1) - 4A(\mathbf{v}_1)C(\mathbf{v}_1)}}{2A(\mathbf{v}_1)}$$

if $A(\mathbf{v}) \ne 0$; otherwise we have

$$t = \frac{-C(\mathbf{v}_1)}{B(\mathbf{v}_1)}$$

provided that $B(\mathbf{v}_1) \ne 0$. Note that, for some t to be an acceptable solution, it has to be a positive real. If there are no acceptable solutions, $S(\mathbf{v}_1)$ is empty. If there is one acceptable solution t we put $t_1 = 0$ and $t_2 = t$. Otherwise, there are two solutions. Let t_2 denote the larger one. Although t_1 and t_2 are functions of $\mathbf{v}$, only when a further clarification is needed, will they be written as $t_1(\mathbf{v})$ and $t_2(\mathbf{v})$.

For an acceptable shape, S (acceptable for shape design), $S(\mathbf{v}_1)$ should be bounded, and not include $\mathbf{v}_1\, t$ for arbitrarily large t. More precisely, it should be the case that

$$S(\mathbf{v}_1) = \{\mathbf{v} = \mathbf{v}_1 t \mid t \in [t_1, t_2]\}.$$

This corresponds to the case shown in Figure 2.a. The complement shape in Figure 2.b is not bounded and therefore not acceptable. In both cases, S is shown with black color. To exclude these, it should be the case that for all $\mathbf{v}$,

$$A(\mathbf{v}) \ge 0 \text{ and}$$
$$A(\mathbf{v}) = 0 \Rightarrow B(\mathbf{v}) > 0.$$

In the following we assume that this condition is satisfied.

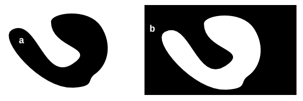

Figure 2: Bounded and unbounded shapes.

3 Parameterization of Ray-Quadrics

Let the directions of the vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ be the same. Then $\mathbf{v}_2 = \alpha \mathbf{v}_1$ where α is a truly positive real. The equations of the rays that are in the directions of $\mathbf{v}_1$ and $\mathbf{v}_2$

are $\mathbf{v} = \mathbf{v}_1 t$ and $\mathbf{v} = \mathbf{v}_2 u = \alpha \mathbf{v}_1 u$, where both t and u are positive reals. The intersection equations will be in the form of

$$A(\mathbf{v}_1)t^2 + B(\mathbf{v}_1)t + C(\mathbf{v}_1) = 0 \text{ and} \tag{2}$$

$$A(\mathbf{v}_2)u^2 + B(\mathbf{v}_2)u + C(\mathbf{v}_2) = 0. \tag{3}$$

By replacing $\mathbf{v}_2 = \alpha \mathbf{v}_1$ in equation 3, we find

$$A(\mathbf{v}_1)\alpha^2 u^2 + B(\mathbf{v}_1)\alpha u + C(\mathbf{v}_1) = 0$$

This equation means that if t_* is a solution of the equation 2, then one of the solutions of the equation 3, u_*, is equal to t_*/α. Thus, we find that the intersection points are the same for both rays:

$$\mathbf{v}_* = \mathbf{v}_1 t_* = \frac{\alpha \mathbf{v}_1 t_*}{\alpha} = \mathbf{v}_2 u_*$$

Since the position of the intersection points depends only on the directions of the vectors, in order to find intersection points in a given direction only one vector is enough. As a result, in order to give a parametric description of a ray-quadric shape, it is sufficient to provide a parametric equation of for some vector family that give only one vector in each direction.

3.1 Guide Shapes and Parameterization

Let the shape S be represented by the parametric equation

$$S = \{\mathbf{v} = \mathbf{f}(\mathbf{s}) \mid \forall \mathbf{s} \in \mathcal{P}\},$$

where $\mathcal{P}$ is a parameter space. This shape provides us a vector family that can be used for parameterization and computation of ray-quadrics if every ray originating from $\mathbf{0}$ intersects the shape at at most one point. So, for $\mathbf{s}_1 \neq \mathbf{s}_2$, the vectors $\mathbf{f}(\mathbf{s}_1)$ and $\mathbf{f}(\mathbf{s}_2)$ have different directions. Such a shape is a star with center $\mathbf{0}$. When such a star shape is used for parameterization of ray-quadrics, we call it a *guide shape*. Note that for parameterization of solid shapes, the guide shapes will be surfaces.

A natural example of a guide surface is a sphere. A parametric equation of a sphere is $x = \sin\theta \sin\psi$, $y = \cos\theta \sin\psi$, and $z = \cos\psi$. Using this parametric equation of a sphere in a ray-affine implicit representation: $B(\mathbf{v}) + C(\mathbf{v}) \leq 0$; we find the parametric equations for any ray-affinely represented shape

$$x = -\frac{\sin\theta \sin\psi \; C(\sin\theta\sin\psi, \cos\theta\sin\psi, \cos\psi)}{B(\sin\theta\sin\psi, \cos\theta\sin\psi, \cos\psi)},$$

$$y = -\frac{\cos\theta \sin\psi \; C(\sin\theta\sin\psi, \cos\theta\sin\psi, \cos\psi)}{B(\sin\theta\sin\psi, \cos\theta\sin\psi, \cos\psi)},$$

$$z = -\frac{\cos\psi \; C(\sin\theta\sin\psi, \cos\theta\sin\psi, \cos\psi)}{B(\sin\theta\sin\psi, \cos\theta\sin\psi, \cos\psi)}.$$

This parametric equation of a sphere is not a good choice since it generates congestions at both poles.

In general, a guide shape does not have to be represented by only one parametric equation. In fact, most of the shapes in computer graphics are represented by more than one parametric equation. For instance, a shape that consists of polygons can be considered a shape represented by a set of linear parametric equations, or a shape represented by patches can be considered a shape defined by a set of bilinear parametric equations.

3.2 General Guide Shapes

Definition 3 A parametrically represented shape $S = \cup_{i=1}^{n} S_i$ is the one that is represented by by a set of parametric equations, $\mathbf{f}_i : \mathcal{P} \longrightarrow \mathcal{V}$, where

$$S_i = \{\mathbf{v} = \mathbf{f}_i(\mathbf{s}) \mid \forall \mathbf{s} \in \mathcal{P}\}, \tag{4}$$

and $\mathcal{P}$ is a given parameter space. This parametrically represented shape S is a *bounded star shape* if any ray originating from $\mathbf{0}$ intersects the shape S exactly at one point.

Since any ray originating from $\mathbf{0}$ intersects a bounded star shape S exactly at one point, the difference of intersection points and $\mathbf{0}$ give us the equation of a vector family. In other words, we can use the parametric equation of a bounded star shape as a parametric equation of vector family for parameterization of ray-linear implicit representations.

Definition 4 A parametrically represented bounded star shape S is called a *guide shape* if it is used for parameterization of ray-linearly represented shapes.

The related parametric equation for a ray-quadric shape is in the following form: $S = \cup_{i=1}^{n} S_i$, where

$$S_i = \begin{cases} \{\mathbf{f}_i(\mathbf{s})t \mid t_1 \leq t \leq t_2\} & \text{if } t_1, t_2 \geq 0, \\ \emptyset & \text{otherwise,} \end{cases}$$

where $t_1 = t_1(\mathbf{f}_i(\mathbf{s}))$ and $t_2 = t_2(\mathbf{f}_i(\mathbf{s}))$.[1] For the rendering quality of the shape, the points where $t_1 = t_2$ must be determined as precise as possible. Otherwise large errors can occur as shown in Figure 3. For the shapes rendered in this paper, we used bisection method [PFTT86] on the equation $B(\mathbf{f}_i(\mathbf{s}))^2 - 4A(\mathbf{f}_i(\mathbf{s}))C(\mathbf{f}_i(\mathbf{s})) = 0$ in order to determine the points where $t_1 = t_2$.

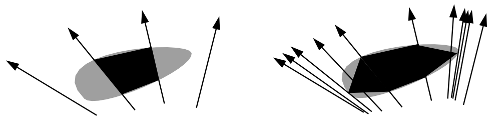

Figure 3: Improving the rendering quality of the shape.

For the figures in this paper, we use icosahedra or cubes as guide surfaces since they give better distributions. When polyhedra with planar faces are used as guide shapes, the parametric equations becomes simpler. For instance, let

[1]Note that if there is one acceptable solution, t, we put $t_1 = 0$ and $t_2 = t$ and if there are two solutions. we chose larger one as t_2.

the i^{th} square face of a cube be given by four vectors: $\mathbf{v}_{i,0,0}$, $\mathbf{v}_{i,0,1}$, $\mathbf{v}_{i,1,1}$, $\mathbf{v}_{i,1,0}$ for $i = 1, 2, \ldots, 6$. The parametric equation for the i^{th} face is $f_i(u, v) = \mathbf{v}_{i,0,0}(1 - u - w) + \mathbf{v}_{i,0,1}w + \mathbf{v}_{i,1,0}u$, where $u, w \in [0, 1]$. Using these parametric equations of the faces of a cube in a ray-linear equation we find the parametric equation of the boundary of a given ray-quadric shape.

4 Ray-Affines and Star Shapes

If ray-affine formulas are used as in the following implicit form,

$$S = \{\mathbf{v} \mid B(\mathbf{v}) + C(\mathbf{v}) \leq 0\},$$

only star shapes can be generated [Akl96]. However, for a ray-affine implicit form, to find a parametric equation of the shape is easier. The parametric equation of shape represented by a ray-affine inequality is the following:

$$\mathbf{v}(\mathbf{s}) = \frac{-C(\mathbf{f}(\mathbf{s}))}{B(\mathbf{f}(\mathbf{s}))}\mathbf{f}(\mathbf{s})$$

Ray-affine implicit forms also provide a powerful shape design methodology based on the constructive geometry of Ricci.[2]

4.1 Always-Positive Ray-Linears

Let $B^+(\mathbf{v})$ be an non-negative ray-linear and $C(\mathbf{v}) = -1$, then the following ray-affine implicit form

$$S = \{\mathbf{v} \mid B^+(\mathbf{v}) - 1 \leq 0\}. \tag{5}$$

is exactly like the implicit form used in the constructive geometry of Ricci. Moreover, non-negative ray-linears are closed under the approximate union and intersection operations of Ricci. Let $B_1^+(\mathbf{v})$ and $B_2^+(\mathbf{v})$ be non-negative ray-linears. If we apply generalized form of Ricci's union and intersection operations, we get

$$B^+(\mathbf{v}) = \sqrt[p]{B_1^+(\mathbf{v})^p + B_2^+(\mathbf{v})^p}.$$

This formula is also an non-negative ray-linear [Akl93] for all real p values since

$$B^+(\mathbf{v}t) = \sqrt[p]{B_1^+(\mathbf{v})^p + B_2^+(\mathbf{v})^p}\ t.$$

Note that this generalized operation includes both approximate union and approximate intersection operations depending on the sign of p. The operation is an approximate union operation for the values of $p < -2$ and it is intersection operation for the values of $p > 2$.[3]

[2]Although Ricci's operations provide a very powerful tool for shape modeling, because of the difficulty of rendering of the shapes described by using these operations, the method has not been used in its full power.

[3]Note that it is not possible include Ricci's approximate set difference since $\frac{1}{B^+(\mathbf{v})}$ is not ray-linear. In the next section, we introduce another set-difference operator.

We use non-negative ray-linears in order to develop the modeling tool. As explained above, non-negative ray-linears are closed under Ricci's exact and approximate union and intersection operators. For ease of reading, we always refer to non-negative ray-linears as ray-linears. Therefore, in order to define a control shape we can apply exact set operations iteratively to generate a structure that can be expressed as nested exact unions or exact intersections over a starting set of ray-linears. By replacing each exact union operator $(min(.,.))$ by an approximate union operator ($\sqrt[-p]{(.)^{-p}+(.)^{-p}}$) and each exact intersection operator $(max(.,.)$) by an approximate intersection operator ($\sqrt[p]{(.)^p + (.)^p}$), we obtain ray-linear implicit representations of smooth approximations of the ray-linearly represented control shape constructed by exact set operations. The starting set of ray-linear building blocks can be any ray-linear function. In the next section, we explain the building blocks we use.

4.2 Building Blocks for Ray-Linears

As is widely known, the distance function $B^+(x, y) = \sqrt[p]{|x|^p + |y|^p}$ leads to the following implicit inequality which provides a large number of shapes depending on the blending parameter p:
$S = \{[x, y] \mid \sqrt[p]{|x|^p + |y|^p} \leq 1\}$. Note that the distance function is a ray-linear function. When p goes to ∞, S becomes square: ■ $= \{[x, y] | max(|x|, |y|) \leq 1\}$.

Since the maximum operator is an intersection operator over the implicitly represented shapes this square shape can be viewed as a control shape that is constructed by the intersection of two infinite symmetric stripes that are given by following implicit inequalities: ▬ $=$ $\{[x, y] \, ||x| \leq 1\}$ and ▮ $= \{[x, y] \, ||y| \leq 1\}$. Note that both x and y are linears but are not always non-negative. On the other hand, absolute values of these linear functions, $|x|$ and $|y|$, are ray-linears.

4.2.1 Symmetric Stripes

In general, linear functions define half-spaces

$$S = \{\mathbf{v} \mid L(\mathbf{v}) - 1 \leq 0\}$$

that include $\mathbf{0}$. A linear function is a ray-linear-homogenous polynomial. In three dimension, it is in the form of $L(\mathbf{v}) = L([x, y, z]) = \mathbf{v}_0 \bullet \mathbf{v} = ax + by + cz$ where $\mathbf{v}_0 = [a, b, c]$. However, since linear functions $L(\mathbf{v})$ are not non-negative, another form should be used. Remembering that absolute value functions are included by ray-linear, let us examine $|L(\mathbf{v})|$. This function is an non-negative ray-linear, and, it provides symmetric strips

$$S = \{\mathbf{v} \mid \ |L(\mathbf{v})| - 1 \leq 0\}.$$

As a result, in order to describe a family of simple ray-linear building blocks we simply take the absolute values of linears. Let $L_i(\mathbf{v})$ be a linear function, then $|L_i(\mathbf{v})|$ is ray-linear. Note that the implicit inequality $|L_i(\mathbf{v})| \leq 1$ describes an n-dimensional infinite symmetric stripe. Therefore, $S = \{\mathbf{v}|max(|L_1(\mathbf{v})|, \ldots, |L_n(\mathbf{v})|) \leq 1\}$ is an n-dimensional control shape generated by the intersection of such n-dimensional infinite symmetric stripes. For instance, in two dimension the following equation gives an hexagonal control shape: $S = \{[x, y] \ |max(|\sqrt{2}x|, |\sqrt{2}y|, |x + y|) \leq 1\}$. As explained in the previous section, the inequality $\sqrt[p]{|\sqrt{2}x|^p + |\sqrt{2}y|^p + |x+y|^p} \leq 1$ gives a smooth approximation of a hexagonal control shape. An example of the minimum operator in two dimension is a star octagon shape given as the union of two squares: $S = \{[x, y] \ |min(max(|x|, |y|), max(|x + y|/\sqrt{2}, |x - y|/\sqrt{2}) \leq 1\}$ and the shape that can smoothly approximate this star octagon is given by the inequality: $\sqrt[-p3]{(\sqrt[p1]{|x|^{p1} + |y|^{p1}})^{-p3} + (\sqrt[p2]{|x+y|^{p2} + |x-y|^{p2}})^{-p3}} \leq 1$, where p_1, p_2 and $-p_3$ are blending parameters.

Since infinite stripes are symmetric around the origin, their intersection can only generate symmetric shapes: *symmetric* polygons in two-dimensions or *symmetric* polyhedra in three-dimensions. Even-sided star or regular polygons with parallel edges are *symmetric* polygons in two-dimensions. Some examples of *symmetric* polyhedra in three-dimensions are the cube, octahedron, icosahedron, dodecahedron, cuboctahedron and great dodecahedron. Another example of the use of symmetric stripes are non-toroidal super-quadrics [SP91] which are given by the inequality: $\sqrt[p2]{(\sqrt[p1]{|x|^{p1} + |y|^{p1}})^{p2} + |z|^{p2}} \leq 1$. The cube is the control shape for non-toroidal super-quadrics.

Symmetry is not always a desired feature for solid modeling. For instance, in two-dimensions not only irregular polygons but even regular polygons with an odd number of edges such as regular triangles or regular pentagons cannot be used as a control shape. In three-dimensions the regular tetrahedron cannot be used as control shape. Since any convex or star polyhedron can be described as a combination of exact intersections and unions of half-spaces, we investigate the possibility of ray-linear representation of half-spaces.

4.2.2 Half-Spaces

As mentioned earlier, half-spaces can be described by linear inequalities of the form $L(\mathbf{v}) \leq 1$, however, this representation is not useful for our purposes since L is not always non-negative. Recall that since Ricci's approximate operators include roots, functions that are not always non-negative cannot be used. In order to find an alternative representation, we assume that there exists an operation $H(L(\mathbf{v}))$ such that $H(L(\mathbf{v}))$ is a ray-linear and describes the same half-space as linear function L, giving

$$\{\mathbf{v}|L(\mathbf{v}) \leq 1\} = \{\mathbf{v}|H(L(\mathbf{v})) \leq 1\}. \qquad (6)$$

We can show that such an operation $H(L(\mathbf{v}))$ indeed exists.[4]

First note that $|L(\mathbf{v})| \leq 1$ will create two inequalities: **1:** $L(\mathbf{v}) \leq 1$ if $L(\mathbf{v}) \geq 0$ and **2:** $-L(\mathbf{v}) \leq 1$ if $L(\mathbf{v}) < 0$. The second inequality is an unwanted one that creates symmetric stripes. This unwanted inequality can be eliminated by simply equating the negative part to zero. This operation does not affect the shape since $L(\mathbf{v}) - 1$ will still be negative. In other words, if we can ensure $H(L(\mathbf{v})) = 0$ when $L(\mathbf{v}) < 0$ and $H(L(\mathbf{v})) = L(\mathbf{v})$ when $L(\mathbf{v}) \geq 0$, $H(L(\mathbf{v}))$ will be a ray-linear that satisfies the condition given in equation 6. One function that qualifies is

$$H(L(\mathbf{v})) = 0.5L(\mathbf{v}) + 0.5|L(\mathbf{v})|. \qquad (7)$$

By using half-spaces as building blocks, we obtain ray-linear implicit representations which are easily parameterizable. Therefore, once the related parametric equations are obtained, computing the shapes is simply the evaluation of the related parametric equations. However, the complexity parametric equations that represents star solids increases as the number of building blocks (in this case half-spaces) increases. Therefore, there has been a need for faster evaluation of the parametric equations.

4.3 Interactive Construction of Smoothly Blended Star Solids

In this section, we briefly present a computationally efficient method which we developed earlier in order to guarantee the interactive construction of ray-linearly represented solids. Based on our method, computation of a new solid shape when a new half-space is added or when the position of an existing half-space is changed can be performed in constant time and in space linear in the number of half-spaces. Because of this method, it has been possible to interactively construct ray-linearly represented solids.

This method , by reusing the previous computation, makes the computation of these parametric equations independent of the number of half-spaces. As a result of this independency, when a user adds a new convex polyhedron or changes the position of a vertex of a convex polygon, a new solid shape can always be computed in constant time.

Based on this fast construction algorithm, we have also developed a three-dimensional modeling tool in order to construct smoothly blended star solids. By using this program, it is possible to construct with union and intersection

[4]In order to represent half-spaces Ricci used exponential of linear functions [Ric73]:

$$S = \{\mathbf{v} \mid a.e^{L(\mathbf{v})} - 1 \leq 0\},$$

where a is a positive real number. Note that $a.e^{L(\mathbf{v})}$ is non-negative but not ray-linear. However, we observe that these exponential building blocks result in ray-exponentials which seem to be useful for modeling and need a further investigation. In addition, note that we cannot use affine building blocks since they do not result in ray-linears.

operations over half-spaces that includes origin. These operations can only generate star shapes which may seem a big restriction; however, star shapes includes many interesting shapes [Akl96]. For instance, all the faces in figure 7 are star shapes. Each face is given by one non-negative ray-linear formula. Each formula is designed by approximate union operations over ray-linear formulas that give smooth approximation of general convex prisms. The formulas of each convex prism are constructed from formulas of two two-dimensional convex shapes. We designed these formulas by using two two-dimensional projections that give front and side views of a face. Designing each face took approximately fifteen minutes. A formula for a face is represented by around 70 two- or three-digit decimal numbers. These numbers are either coefficients of linear functions that define half-spaces or smoothing coefficients of approximate unions and intersections (p's). Small changes in coefficients do not considerably effect the shapes, so for the purpose of shape design this technique is robust.

5 Ray-Quadrics

In this section we introduce ray-quadric formulas, based on non-negative ray-linears, that leads to some intuitive operations over star shapes with the same center.

5.1 Symmetric Set-Difference Operation

The following ray-quadric inequality provides symmetric set-difference operation over two star shapes represented by non-negative ray-linears

$$S_1 \otimes S_2 = \{\mathbf{v} \mid (B_1^+(\mathbf{v}) - 1)(B_2^+(\mathbf{v}) - 1) \leq 0\}.$$

The multiplication of $B_1^+(\mathbf{v}) - 1$ and $B_1^+(\mathbf{v}) - 1$ is less than zero iff one of them is smaller than zero and the other one larger than zero. The multiplication operation over formulas provides a symmetric set-difference operation over shapes. Note that the symmetric set-difference is like an exclusive-or operation over boolean formulas.

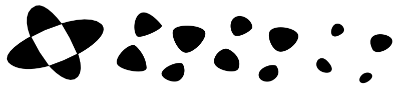

Figure 4: Exact and approximate symmetric set difference operations

Symmetric set-difference over star shapes with the same center does not generate interesting shapes. The results will look like the union of two star shapes with the same center. Let us rewrite the inequality without parenthesis to observe this phenomenon, yielding

$$S_1 \otimes S_2 = \{\mathbf{v} \mid (B_1^+(\mathbf{v})B_2^+(\mathbf{v}) - B_1^+(\mathbf{v}) - B_2^+(\mathbf{v}) + 1) \leq 0\}.$$

Since this inequality is a ray-quadric, the intersection inequality with a ray originating from $\mathbf{0}$ is in the form of $(xy)t^2 - (x+y)t + 1 \leq 0$, where $x = B_1^+(\mathbf{v}_1)$ and $y = B_2^+(\mathbf{v}_1)$ and $\mathbf{v}_1$ is the direction of the ray. The intersection with the ray occurs iff $\Delta = (x+y)^2 - 4xy \geq 0$. However, $(x+y)^2 - 4xy = (x-y)^2$ and it is always bigger than or equal to zero. This means that $S_1 \otimes S_2$ is like a crust completely covering an inside hole. The shape will not be smooth at the edges where $\Delta = 0$. In order to create holes on crust and smooth out the edges, we need a slight modification in inequality. It is possible to change Δ, by adding a new constant term ϵ^2 to the formula. The resulting approximate symmetric set difference is

$$(S_1 \otimes S_2)_\epsilon = \{\mathbf{v} \mid (B_1^+(\mathbf{v}) - 1)(B_2^+(\mathbf{v}) - 1) + \epsilon^2 \leq 0\}.$$

The new Δ will be in the form of $\Delta = (x+y)^2 - 4xy(1+\epsilon^2) = (x-y)^2 - \epsilon^2 xy \geq 0$. It is clear that bigger ϵ yields smaller Δ. As a result of smaller Δ, the intersection points with the ray either come closer or are eliminated completely (when Δ becomes negative). As a result of this modified operation sharp edges are smoothed out by creating holes as shown in Figure 4. The symmetric set-difference operation is not by itself interesting for shape modeling. We provided this discussion in order to simplify the explanation of a more complicated approximate set-difference operation in the next section.

5.2 Set-Difference and Approximate Set-Difference Operations

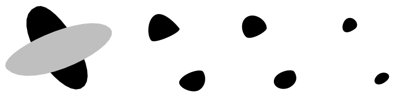

Figure 5: Exact and approximate set difference operations

A slightly modified version of symmetric set-difference gives the set difference of two star shapes represented by non-negative ray-linears as

$$S_1 - S_2 = \{\mathbf{v} \mid (B_1^+(\mathbf{v}) - C_0(\mathbf{v}))(B_2^+(\mathbf{v}) - C_0(\mathbf{v})) \leq 0\}$$

where ray-constant $C_0(\mathbf{v}) = \text{sign}(B_2^+(\mathbf{v}) - B_1^+(\mathbf{v}))$ and with

$$\text{sign}(z) = \begin{cases} +1 & \text{if } z \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

To show why this operation gives set difference, we can start from the observation that

$$S_1 - S_2 = \{\mathbf{v} \mid B_1^+(\mathbf{v}) \leq 1 \wedge B_2^+(\mathbf{v}) \geq 1\}.$$

So we have to establish the equivalence of the two conditions

$$B_1^+(\mathbf{v}) \leq 1 \wedge B_2^+(\mathbf{v}) \geq 1, \text{ and}$$
$$(B_1^+(\mathbf{v}) - C_0(\mathbf{v}))(B_2^+(\mathbf{v}) - C_0(\mathbf{v})) \leq 0,$$

for all $\mathbf{v}$.

If for a given $\mathbf{v}$, $B_2^+(\mathbf{v}) < B_1^+(\mathbf{v})$, the first of these conditions is not satisfied. In this case $C_0(\mathbf{v}) = -1$, so the second condition reduces to

$$(B_1^+(\mathbf{v}) + 1)(B_2^+(\mathbf{v}) + 1) \leq 0,$$

which is also clearly not satisfied, since $B_2^+(\mathbf{v})$ and $B_1^+(\mathbf{v})$ are non-negative.

If, on the other hand, $B_2^+(\mathbf{v}) \geq B_1^+(\mathbf{v})$, we have $C_0(\mathbf{v}) = +1$, so the second condition reduces to

$$(B_1^+(\mathbf{v}) - 1)(B_2^+(\mathbf{v}) - 1) \leq 0,$$

which is the case iff $B_1^+(\mathbf{v}) \leq 1 \wedge B_2^+(\mathbf{v}) \geq 1$ or $B_1^+(\mathbf{v}) \geq 1 \wedge B_2^+(\mathbf{v}) \leq 1$. Combining this with the condition $B_2^+(\mathbf{v}) \geq B_1^+(\mathbf{v})$, we obtain the first condition:

$$B_1^+(\mathbf{v}) \leq 1 \wedge B_2^+(\mathbf{v}) \geq 1.$$

Set difference of two star shapes generates sharp edges. To smooth out these edges, the approximation idea in the previous section is used. By adding a constant term ϵ^2 to the inequality, the approximate set-difference operation

$$(S_1 - S_2)_\epsilon = \{\mathbf{v} \mid (B_1^+(\mathbf{v}) - C_0(\mathbf{v}))(B_2^+(\mathbf{v}) - C_0(\mathbf{v})) + \epsilon^2 \leq 0\}$$

is obtained. The result of the approximate operation is represented in Figure 5. Examples of approximate set differences for different ϵ are shown in Figures 8 and 9. In the next section, we introduce another set of operators: Border-intersection and flesh.

5.3 Border-Intersection and Flesh Operators

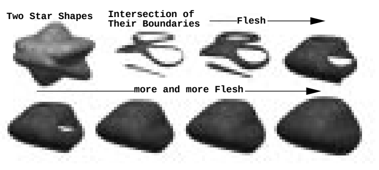

Figure 6: Border-intersection and flesh operations

In this section, we are interested in the intersection of *borders* of two shapes represented by general ray-affines. In two-dimensional space this operation gives a set of points, in three-dimension a set of space curves. Let δS denote the border of S. Then

$$\delta S = \{\mathbf{v} \mid (B(\mathbf{v}) + C(\mathbf{v}))^2 \leq 0\}$$

Then, the intersection of borders of two shapes S_1 and S_2 can be given by a ray-quadric inequality:

$$\delta S_1 \cap \delta S_2 = \{\mathbf{v} \mid (B_1(\mathbf{v}) + C_1(\mathbf{v}))^2 + (B_2(\mathbf{v}) + C_2(\mathbf{v}))^2 \leq 0\}$$

where $C_i(\mathbf{v})$'s can be zero. For instance,

$$(\sqrt{x^2 + z^2} - 1)^2 + y^2 \leq 0$$

gives a space circle. To generate a flesh around the space curves, it is enough to subtract a constant ϵ^2 from the inequality:

$$\delta S_1 \cap_\epsilon \delta S_2 = \{\mathbf{v} \mid (B_1(\mathbf{v}) - C_2(\mathbf{v}))^2 + (B_2(\mathbf{v}) - C_2(\mathbf{v}))^2 - \epsilon^2 \leq 0\}$$

provides a flesh around this intersection. The results of border intersection and flesh operation is represented in Figure 6. For instance, $(\sqrt{x^2 + z^2} - 1)^2 + y^2 - \epsilon^2 \leq 0$ is an equation of a torus.

Examples of border-intersection and flesh operators are shown in Figures 10 and 11. The shapes in Figure 10 are represented by the inequality

$$\left(\sqrt[-16]{(x^2 + y^2)^{-8} + (y^2 + z^2)^{-8} + (z^2 + x^2)^{-8}} - 1 \right)^2 +$$
$$\left(\sqrt{x^2 + y^2 + z^2} - 1 \right)^2 - \epsilon^2 \leq 0.$$

The first part of this formula is union of three perpendicular cylinders and the second part is a sphere. The border-intersection operation creates six circular space curves and ϵ changes the tickness of the flesh as shown in Figure 10. The shapes in Figure 11 are described by generating a flesh around the intersection of the surfaces of an icosahedron and a sphere. Since the intersection points are actually vertices of a truncated icosahedron [Wil79], the resulting shapes resemble a soccer ball. Note that changing the thickness of flesh, ϵ, changes the topology of the shape; however the change is expected and intuitive.

6 Ray-Constants

The ray-constant part of ray-quadrics has not been examined up to this point. In fact, ray-constants are responsible for the greater choice of functions. The next theorem states this fact.

Theorem 1 *[Akl93] Let $C_1(\mathbf{v})$, $C_2(\mathbf{v})$, ..., $C_n(\mathbf{v})$ be ray-constants and let f be any function from $\Re^n$ to $\Re$. Then*

$$f(C_1(\mathbf{v}), C_2(\mathbf{v}), \ldots, C_n(\mathbf{v}))$$

is also ray-constant.

An example of ray-constant functions is the angle between two vectors. For instance

$$\frac{x}{\sqrt{x^2 + y^2 + z^2}}$$

gives the cosine of the angle between the vectors $[1, 0, 0]$ and $[x, y, z]$. It is possible to attach a unique value to every ray with ray-constants. In other words, each ray-constant function can be considered a parameter. Using more than one ray-constant function it is possible to give a distinct set of parameters to each ray. Then by using a parametric function over this set of ray-constant functions another ray-constant function will be obtained. Examples of ray-constant functions are shown in Figures 12 and 13. We generated the shapes in Figure 12 by using set difference and border-intersection and flesh operations over two planet-like shapes. First, by using a sinusoidal ray-constant function we described a mountain range which covers planets like a spiral. We chose the shape of the planets as the same except for the position of mountain ranges. The highest peak of a mountain in one planet corresponds with the lowest point of a valley in the other planet. This relation between valleys and mountains created holes as shown in Figure 12.

Each teapot in Figure 13 consists of four shapes: body, lid, handle and spout. All lids and bodies are star shapes which are described by approximate union and intersection operations. We designed handles and spouts by border-intersection and flesh operation over a generalized toroidal figure. The thickness of the flesh is described by a ray-constant term which is a Bezier curve depending on angles. We designed the overall shapes of toroidal handle and spout in two dimensions by using approximate union and intersection operations. The first teapot and the last elephanTpot have the same formulas except for the Bezier coefficients of the ray-constant term. For the first three shapes, only the smoothing parameter p of the approximate unions and intersections is different.

7 Conclusion

We introduced a new building block for implicit representations: ray-quadrics. Ray-quadrics provide a large variety of shapes with intuitive shape construction, real-time rendering and interactive shape design.

In order to represent shapes that cannot be represented by a single ray-quadric formula, ray-quadric formulas can be used in constructive solid geometry [RV82] or in Blinn's exponential functions [Bli82] in the form of $\sum_{i=0}^{n} e^{-A_i(\mathbf{v}-\mathbf{v}_i)-B_i(\mathbf{v}-\mathbf{v}_i)}$, or in the soft object equations [WT90]. Ray-quadrics can be deformed by using deformations [Bar84], [SP91] or superposed by using convolution [Blo91].

Higher-degree ray-polynomials can also be used to represent shapes that cannot be represented by a single ray-quadric formula. For instance, it is possible to represent a teapot with *only one ray-cubic inequality.* The main problem is in finding intuitive geometric operators which give higher-degree ray-polynomials. In addition, rendering will not be as easy as for ray-quadrics. It seems that these two problems may limit the usage of ray-polynomials to ray-quadrics.

8 Acknowledgments

I would like to thank to Graphics, Visualization and User interface Center at Georgia Tech and Computer Science Department at Yildiz Technical University. I also would like to thank Lambert Meertens, Donald House and Jason Rosson for their helpful suggestions.

References

[Akl93] E. Akleman. Construction of convex and star shapes with yontsal functions (always non-negative ray-linears). *International Symposium on Computer and Information Sciences*, 8, November, 1993.

[Akl96] E. Akleman. Interactive construction of smoothly blended star solids. *Proceedigns of Graphical Interface'96*, May, 1996.

[Bar81] A. H. Barr. Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications*, 1(1), 1981.

[Bar84] A. H. Barr. Global and local deformations of solid primitives. *Computer Graphics*, 18(3), 1984.

[BI92] C. Bajaj and I. Ihm. Smoothing polyhedra using implicit algebraic splines. *Computer Graphics*, 26(2), 1992.

[Bli82] J. I. Blinn. A generalization of algebraic surface drawing. *ACM transaction on Graphics*, 1(3), 1982.

[Bli90] J. I. Blinn. The algebraic properties of homogeneous second order surfaces. In B. L. M. Wyvill and J Bloomenthal, editors, *Modeling and Animating with Implicit Surfaces.* ACM SIGGRAPH tutorial, 1990.

[Blo91] J. Bloomenthal. Convolution surfaces. *Computer Graphics*, 25(4), July 1991.

[FB88] D. Forsey and R. Bartels. Hierarchical b-spline refinement. *Computer Graphics*, 22(4), 1988.

[Han88] A. J. Hansen. Hyperquadrics: Smoothly deformable shapes with convex polyhedral bounds. *Computer Vision, Graphics and Image Processing*, 44, 1988.

[PFTT86] W. H. Press, B. P. Flannery, S. A. Teukolsky, and Vetterling W. T. *Numerical Recipes.* Cambridge University Press, 1986.

[Ric73] A. Ricci. A constructive geometry for computer graphics. *The Computer Journal*, 16(2), May 1973.

[RO87] A. P. Rockwood and J. Owen. Blending surfaces in solid geometrical modeling. In G. Farin, editor, *Geometric Modeling.* SIAM, Philadelphia, 1987.

[RV82] A. A. G. Requicha and H.B. Voelcker. Solid modeling: A historical summary and contemparay assessment. *IEEE Computer Graphics and Applications*, 2(2), March 1982.

[Sed85] T. W. Sederberg. Piecewice algebraic surface patches. *Computer Aided Geometric Design*, 2, 1985.

[SN86] T. W. Sederberg and Goldman R. N. Algebraic geometry for computer aided geometric design. *IEEE Computer Graphics and Applications*, 6(6), June 1986.

[SP91] S. Sclaroff and A. Pentland. Generalized implicit functions for computer graphics. *Computer Graphics*, 25(4), July 1991.

[WH94] A. P. Witkin and P. S. Heckbert. Using particles to sample and control implicit surfaces. *Computer Graphics*, 27(3), 1994.

[Wil79] R. Williams. *The Geometrical Foundation of Natural Structure.* Dover, NewYork, NW, 1979.

[WT90] G. Wyvill and A. Trotman. Ray tracing soft objects. In B. L. M. Wyvill and J Bloomenthal, editors, *Modeling and Animating with Implicit Surfaces.* ACM SIGGRAPH tutorial, 1990.

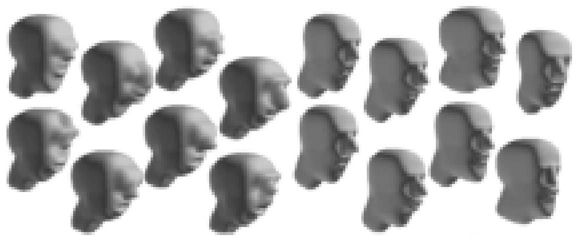

Figure 7: Each face is a star shape that is given by a single ray-linear formula. All these faces constructed in approximately fifteen minutes by changing parts such as the nose, chin or lips.

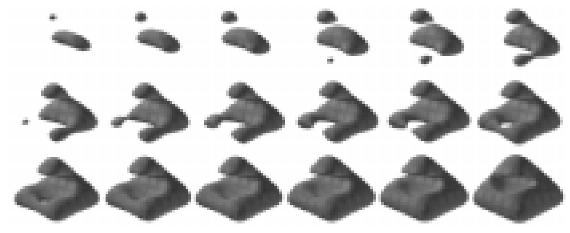

Figure 8: Shapes that are are generated by approximate set difference operations.

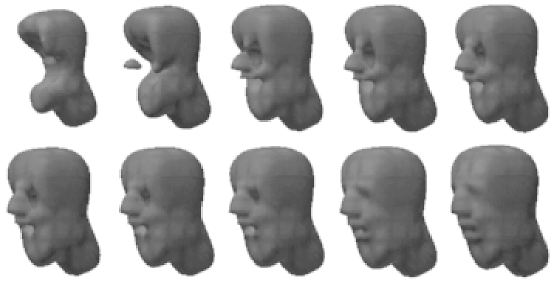

Figure 9: Shapes that are are generated by approximate set difference operations.

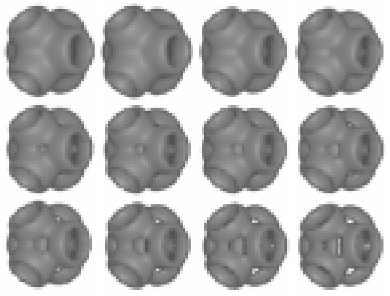

Figure 10: Shapes that are generated by border-intersection and flesh operations.

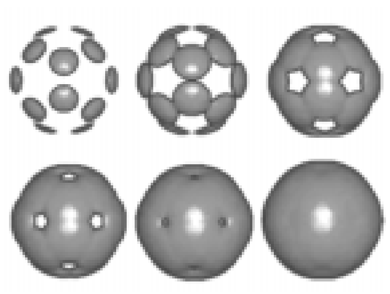

Figure 11: Soccer balls that are generated by border-intersection and flesh operations.

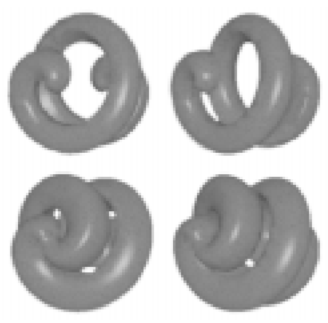

Figure 12: The effect of ray-constant.

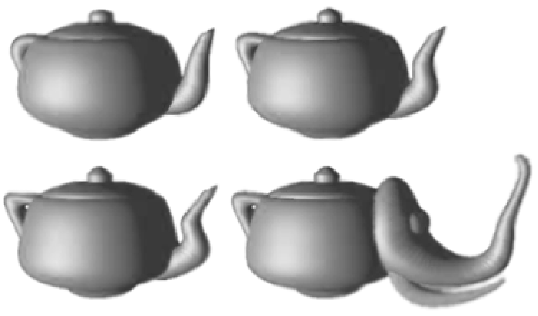

Figure 13: Three teapots and one elephanTpot. Elephant tusks are also ray-quadrics.