Implicit Modelling II
by Brian Wyvill

With a little help from his students
Overview

• Constructive Solid Geometry
• Polygonization
• Precise Contact Modelling
**Constructive Solid Geometry (CSG)**

Primitives are combined using boolean set operations: Union, Intersection, Difference. Each primitive represents a half space, i.e., the set of points defining the half space.

E.g.

- Sphere
- Cylinder
- Plane

Boolean expression (u = union, d = difference, i = intersection)

\[ d(\text{sphere}, \text{cylinder}) \quad u(\text{sphere}, i(\text{i(\text{cylinder}, \text{plane1}), \text{plane2}})) \]

The cylinder is infinite in extent; it is first intersected with two half space planes.
CSG Implementation

Boolean Expressions are usually represented as a binary tree.
CSG Intersections with Voxels

BlobTree

object 1: in \( \bullet \) out

object 2: out \( \bullet \) in

DIFF(1,2) in \( \bullet \) out
**Boolean Operations (Ricci 1973)**

Union and intersection of primitives, A and B may be respectively defined as a composition of the field values, $F_A$, $F_B$

\[
F_A \cup F_B = \max(F_A, F_B)
\]

\[
F_A \cap F_B = \min(F_A, F_B)
\]

Difference use $-\min (F_A, F_B)$

(- in this case inverts inside and outside )
**CSG – Min and Max**

\[ f_{A|B}(p_0) = \text{Max}(f_A(p_0), f_B(p_0)) \]

Depending on position of \( p_0 \)

\[ f_A(p_1) = \text{Max}(f_A(p_1), f_B(p_1)) \]

\[ f_B(p_2) = \text{Max}(f_A(p_2), f_B(p_2)) \]
CSG - Min

BlobTree

\[ f_{A|B}(p_0) = \min(f_A(p_0), f_B(p_0)) \]
Depending on position of \( p_0 \)

\[ f_B(p_1) = \min(f_A(p_1), f_B(p_1)) = 0 \]

\[ f_A(p_2) = \min(f_A(p_2), f_B(p_2)) = 0 \]

Intersection
Difference

Min(f_A(p_0), f_B(p_0)) = 1 - f_{A|B}(p_0)
Depending on position of p_0

Min(f_A(p_1), 1 - f_B(p_1)) = f_A(p_1)
Min(f_A(p_1), 1 - f_B(p_1)) = 0

Object A

Object B

P_0

P_1

P_2

Field = 0

Movie
Polygonization Problem

X is the true intersection point for $C_1$ and $C_2$

Segment $P_1P_2$ is far from $x$.

Estimate for $x$ s. t. $f_1(x) = f_2(x) = 0$

We can apply a first order Taylor expansion to the difference: $n_{12} = x - P_{12}$

\[ 0 = f_1(x) = f_1(P_{12} + n_{12}) = f_1(P_{12}) + (n_{12}, \nabla f_1(P_{12})) \]
\[ 0 = f_2(x) = f_2(P_{12} + n_{12}) = f_2(P_{12}) + (n_{12}, \nabla f_2(P_{12})) \]
\( \lambda_1 = \frac{-f_1}{(\nabla f_1, \nabla f_1)} \)

so

\( n_1 = \frac{-f_1 \nabla f_1}{(tf_1, tf_1)} \)

and similarly

\( n_2 = \frac{-f_2 tf_2}{(tf_2, tf_2)} \)
Adaptive Polygonization

BlobTree
Csoft Wheel before and after removal of artifacts
Canmore Coffee Grinder

BlobTree
Ray Traced Canmore Coffee Grinder

by

Kees van Overveld and Brian Wyvill
Building the Piano

Parametric Bounding Curve

Cylinders intersected with bounding plane and parametric curve
Model of 9ft. Steinway Concert Grand

Plant by Dr. Prusinkiwicz
**Blending to Intersection Planes**

With this scheme a gang cannot be blended here.

Solution Blend becomes a node in the CSG tree known as the **BlobTree**.
Traversing The BlobTree

N - indicates a node in the BlobTree
L (N) - left child R (N) - right child

function F returns the field value for the node N at the point M

function F(N, M)

1. Primitive: F( M)
2. Warp: F( L (N) , w( M)) (warp is a unary operator)
3. Blend: F( L (N) , M)+ F( R (N) , M))
4. Union: max( F( L (N) , M), F( R (N) , M))
5. Intersection: min( F( L (N) , M), F( R (N) , M))
6. Difference: min( F( L (N) , M), 1-F( R (N) , M))

end
BlobTree Examples

Candle Stick Model by Eric Galin

Stamingo by Andy Guy
Unwanted Blending Problem

Global Blending

Controlled Blending

BlobTree
Controlled Blending

Blending Group
\[ f(P) = \max(f(A)+f(B), f(B)+f(C), f(C)+f(D), f(E)+f(F)) \]
Problems with Controlled Blending

1. The Blend Graph (part of the BlobTree)
2. Critical Points (e.g. saddle points Zero Gradient)
Building an Implicit Shell Model

By

Callum Galbraith

Premyslaw Prusinkiewicz

and Brian Wyvill

University of Calgary
Precise Contact Modeling

- **PCM:** Marie-Paul Gascuel (Cani) '93.

![Diagram](image)

**Interpenetration region**
Negligible D

**Undeformed region**
Zero D

**Interpenetration region**
$$D_i(P) = 0.5 - f_j(P)$$

**Propagation region**
$$D_i(P) = B(f_j(P))$$
Controlled Blending and PCM

BlobTree

Controlled Blending with Precise Contact Modelling
The Helico-Spiral

center of similitude of the shell

helico-spiral \( \mathcal{H} \)

shell axis

generating curve \( \mathcal{C} \)

shell surface (rim)

\( z_0 \)

\( r_0 \)

\( z \)

\( x \)

\( y \)
The helico-spiral

\[ \theta = t \]
\[ r = r_0 \xi_r^t \]
\[ z = z_0 \xi_z^t \]
\[ \theta_{i+1} = \theta_i + \Delta \theta \]
\[ r_{i+1} = r_i \lambda_r \]
\[ z_{i+1} = z_i \lambda_z \]
Example Parametric Shell Models
Example Parametric Shell Models
Limitations of the Parametric Model

BlobTree
Implicit Shell Model Construction

The parametric shell model sweeps a generating curve C along a helico-spiral H. The surface is constructed by connecting instances of C placed at fixed intervals along H with a polygon mesh.

The implicit model sweeps a generating implicit field F along the helico-spiral. The surface is constructed by blending instances of F placed at fixed intervals along H.
An Implicit Shell

Using a point primitive as the generating surface we can generate a simple shell. The distribution of primitives along the helico-spiral is the same in each image.
Shell Sections For Controlled Blending

Five point primitives make up a single shell section. As their size is increased the shell section takes shape.

1 section  2 sections  3 sections  5 sections  10 sections

Each section is blended only with it’s two neighboring sections along the helico-spiral.
Using Controlled Blending

Main shell body modelled without controlled blending.

Main shell body modelled with controlled blending.

BlobTree
It is difficult to know exactly where the surface of the shell will be when blending instances of the generating field $F$.

The final surface depends upon:
- the function defining $F$.
- the density of instances of $F$.
- the method used to blend instances of $F$. 
Varying Density of Generating Fields

Distance $D$ between instances of generating field $F$ distributed along a line. $F$ is a point primitive defining a sphere of radius 2. Note: the radius to the zero surface is 4.
Varying the Blending Function

**BlobTree**

Distance $D$ between instances of generating field $F$ and parameter $N$ in Ricci Blend.

- $D = 4, N = 2$
- $D = 1, N = 2$
- $D = 1, N = 4$
- $D = 1, N = 20$
Description of *Murex cabritii*:

- A smallish, oval aperture in a strongly convex body whorl.

- A long slender canal below the main body whorl, narrowly open, with three axial rows of five to six spines.

- Each whorl has three varices (ridges) which bear several sharp curving spines.

- Beaded axial riblets (or small bumps) are present between varices.
The Generating Surface for *Murex cabritii*

On the left are two views of the generating surface used to define the main body whorl for *Murex cabritii*.

A point primitive scaled parallel to the helico-spiral describes the main aperture, while a similarly scaled bent cone models the long slender canal.

On the right is the shell model created by this generating surface.
Adding Axial Spines

Spines are modelled with cone primitives and blended with the shell. On the right each spine has been randomly bent to create a more natural model.
Modelling a Varix

Spines are modelled with a cone primitive.

A 2D texture-map is applied for a more realistic look.

A bend warp is applied to generate the correct shape.

On the left a series of spines of varying height have been added.

To create the ridge of the varix, the model on the right includes two torii blended with the spines.
Modelling the Bumps

Bumps blended with the main body whorl using the standard blend. The larger point primitives of the main body whorl overwhelm the smaller point primitives which model each bump.

Using a Ricci blend we can limit the amount of blending between the body whorl and the bumps. The parameter N in the Ricci blend was set to 400.

To create more realism, each bump is randomly scaled using a normal distribution.
Creating an Opening in the Shell

The inside of the shell is modelled as a whorl slightly smaller than the main whorl.

After using a CSG difference operation to cut away the inside from the main shell, we notice two problems. First, the inside cuts into the previous whorl. And second, the opening is too large.
Creating an Opening in the Shell

To stop the opening from cutting into the previous whorl, an instance of the previous whorl is first cut out of the inside shell model.
Finally an implicit surface is constructed to decrease the size of the opening. It is blended to the shell after the opening is created.
Texturing the Main Shell Body

BlobTree
The Final Shell
Using The BlobTree

Python Model and Animation Description

Python InterfaceTo BlobTree

Ray Tracer  BlobTree Software  Polygonizer

BlobTree