# Modeling a *Murex cabritii* sea shell with a structured implicit surface modeler

Callum Galbraith, Przemyslaw Prusinkiewicz, and Brian Wyvill.
University of Calgary.

## Abstract

Implicit surface modeling systems have been used since the mid−1980s for the generation of cartoon−like characters. Recently implicit models combined with constructive solid geometry (CSG) have been used to build engineering models with automatic blending. This work is based on a structured implicit modeling system which includes CSG, warping, 2D texture mapping and operations based on the *BlobTree*, and its application to the generation of a complex and visually accurate biological model of the sea shell *Murex cabritii*. Since the model is purely procedurally defined and does not rely on polygon mesh operations, it is resolution independent and can be rendered directly using ray tracing. An interface has been built for the *BlobTree* using an interpreted programming language (Python). The language interface readily allows a user to procedurally describe the shell based on numeric data taken fromthe actual object.

## Reference

Callum Galbraith, Przemyslaw Prusinkiewicz, and Brian Wyvill: Modeling a *Murex cabritii* sea shell with a structured implicit surface modeler. *The Visual Computer* vol. 18, pp. 70−80.

The Visual Computer

# Modeling a *Murex cabritii* sea shell with a structured implicit surface modeler

Callum Galbraith,
Przemyslaw Prusinkiewicz,
Brian Wyvill

Department of Computer Science, University of Calgary, Calgary, Alberta T2N 1N4, Canada
E-mail: {callum,pwp,blob}@cpsc.ucalgary.ca

Published online: 15 March 2002


Implicit surface modeling systems have been used since the mid-1980s for the generation of cartoon-like characters. Recently implicit models combined with constructive solid geometry (CSG) have been used to build engineering models with automatic blending. This work is based on a structured implicit modeling system which includes CSG, warping, 2D texture mapping and operations based on the *BlobTree*, and its application to the generation of a complex and visually accurate biological model of the sea shell *Murex cabritii*. Since the model is purely procedurally defined and does not rely on polygon mesh operations, it is resolution independent and can be rendered directly using ray tracing. An interface has been built for the *BlobTree* using an interpreted programming language (Python). The language interface readily allows a user to procedurally describe the shell based on numeric data taken from the actual object.

**Key words:** Modeling – Seashells – Implicit surfaces – Constructive solid geometry

*Correspondence to:* C. Galbraith

## 1 Introduction

The seemingly simple mathematical character of shells, which yield a great variety of beautiful shapes, has attracted much attention from computer modelers. Two motivations for such work are to synthesize realistic images that can be incorporated into computer-generated scenes and to gain a better understanding of the mechanism of shell formation (Fowler et al. 1992, Meinhardt 1995). This paper is concerned with the first of these two goals and is based upon implicit modeling techniques using the *BlobTree* (Wyvill et al. 1999).

The *BlobTree* has made possible the construction of much more complex models than the cartoon-like characters depicted in movies such as Wyvill (1988). In the *BlobTree* system models are defined by expressions which combine implicit primitives using blending, warping, and boolean set operations in an homogeneous fashion. The *BlobTree* also incorporates controlled blending (Guy and Wyvill 1995), and 2D texture mapping (Tigges and Wyvill 1998, Tigges 1999), without which it is difficult to capture naturally occurring shapes and patterns.

The first shell model intended specifically for use in computer graphics was developed by Kawaguchi (1982). He created shell models using polygon meshes. Other methods of modeling shells have included the use of inter-penetrating spheres, and generalized cylinders. Fowler et al. (1992) reviewed previous work on shell modeling and extended the field by introducing free-form parametric curves to capture the shape of shell aperture and by using reaction–diffusion methods to incorporate pigmentation patterns into the models.

Fowler et al. (1992) describes several open problems in the modeling of shells. Two of these are

- *Modeling of spines.* Previous methods of modeling have been able to capture small perturbations of the surface of the shell. Large modifications of the shape such as the spines in *Murex cabritii* (Fig. 1) have not been captured by existing methods.
- *Capturing the thickness of shell walls.* The parametric representations used thus far typically model the shell walls as mathematical surfaces which have no actual thickness. Rendering the inside and outside differently can produce the illusion of a substantive wall, but the opening of the shell is not properly visualized.

In this work both of the above problems are addressed using the *BlobTree*. A model of *Murex*


Digital Object Identifier (DOI) 10.1007/s003710100136

**Fig. 1.** *Murex cabritii*

*cabritii* is described which includes the large spines, shell walls of non-zero thickness, and allows different textures to be applied to different parts of the shell while automatically blending the textures where these parts join. Our model is resolution independent and can be polygonized at an arbitrary resolution, as well as ray traced directly, for higher quality images.

This paper is organized as follows: Sect. 2 discusses existing methods that have been combined to build the model. Section 3 presents the method of model construction. The obtained results are presented and discussed in Sect. 4.

## 2 Background

Background work will be considered in two parts. Formulas that describe the geometry of shells will be discussed in Sect. 2.1. The *BlobTree*, which is used to construct the model, is introduced in Sect. 2.2.

### 2.1 Modeling Shell geometry

As reviewed in Fowler et al. (1992) and Meinhardt (1995), the surface of a shell without protrusions may be defined by sweeping a closed generating curve $C$ in the shape of the aperture of the shell along a logarithmic helico-spiral $S$. The scale of the generating curve increases in geometric progression as the angle of rotation around the shell's axis increases arithmetically.

The helico-spiral is conveniently described in a cylindrical coordinate system. The radius $R$ (distance of a point, $P$, on the helico-spiral from the shell axis) is an exponential function of the angle of revolution $\theta$ around the axis:

$$R(\theta) = R_0 \rho^{\left(\frac{\theta}{360^\circ}\right)} \,,\; R_0 > 0 \,,\; \rho > 1 \,,\; \theta \geq 0 \,, \tag{1}$$

where $R_0$ is the initial radius and $\rho$ is the ratio of the radii corresponding to a rotation of 360°. The vertical displacement, $H$, of point $P$ increases in proportion to the radius:

$$H(\theta) = R(\theta) \cot \beta \,,\; \beta > 0 \,, \tag{2}$$

where $\beta$ is the angle between the axis of the spiral and a line, $L$, passing through successive whorls of the helico-spiral (Fig. 2). A whorl is defined as a single turn or volution of a spiral shell.

The size of the generating curve $C$ at point $P$ can easily be determined under the assumption that $C$ is a circle of radius $D$ lying in the plane including the shell axis and the point $P$, and that the circles in consecutive whorls are tangential to each other. From Fig. 2 we then obtain:

$$D(\theta) = \frac{R(\theta)}{\sin \beta} \frac{\rho - 1}{\rho + 1} \,. \tag{3}$$

In the case of non-circular generating curves, Eq. (3) remains useful as an approximate indicator of the curve size.

### 2.2 The BlobTree

The major advantage of implicit surface modeling systems has been the use of automatic blending between skeletal elements. Recent developments in such systems include the addition of space warping, which provides a method of implementing deformations (Crespin et al. 1996), and Boolean operations

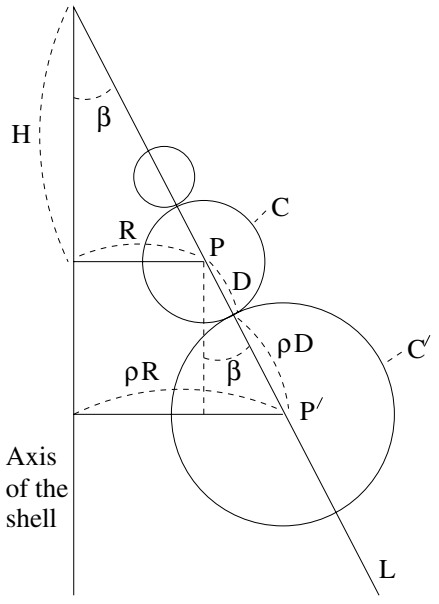



**Fig. 2.** One half of a longitudinal cross-section of a turbinate shell, illustrating Eqs. (2) and (3)

used in constructive solid geometry (CSG) systems (Pasko et al. 1995, Wyvill 1996). CSG systems typically use a tree structure to describe the relationship of Boolean set operations such as union and intersection between half-space primitives.

The *BlobTree* (Wyvill et al. 1999) has been introduced as a method of organizing all of these operations in a manner that enables global and local operations to be exploited in a general and intuitive fashion. In the *BlobTree*, an implicit surface model is defined using a tree data structure which combines implicit surface primitives as leaf nodes, with arbitrary operations such as blending, warping, and Boolean operations as interior nodes. We refer to the structure as the *BlobTree*.

One advantage of the *BlobTree* is that it is easily extended to incorporate new functionality. Several problems have been associated with the use of implicit surfaces as a general modeling method. Of great importance is the ability to make objects blend selectively (locally) rather than globally, and also the lack of a natural coordinate system to allow 2D texturing. These problems have recently been addressed, and their solutions have been incorporated into the *BlobTree* (Guy and Wyvill 1995, Tigges and Wyvill 1998, Tigges 1999). The nature of the *BlobTree* cleanly allows us both local and global texturing of implicit models.

Models are defined by expressions which combine implicit primitives and the operators $\cup$ (union), $\cap$ (intersection), $-$ (difference), $+$ (blend), $\diamond_n$ (super-elliptic blend), $c$ (controlled blend), $w$ (warp), $t$ (translate), $s$ (scale), $r$ (rotate), and $m$ (2D texture map). At the lowest level these operators act on one or more primitives. The result of each operation is a *BlobTree*, and may be passed to another operator. The operators listed above are n-ary with the exception of warp, affine transformations and 2D texture mapping, which are unary operators. An example of a simple *BlobTree* model is given in Fig. 3.

The affine transformations are the standard ones and are defined as: $t(x, y, z)(B)$ – translate *BlobTree B* by $(x, y, z)$; $s(x, y, z)(B)$ – scale *BlobTree B* by $(x, y, z)$; $r(axis, \theta)(B)$ – rotate *BlobTree B* by $\theta$ about the given axis using the right-hand rule.

Blending operators are of particular importance to the model construction described in Sect. 3 and are examined in detail.

*Super elliptic blending* allows the modeler to control the amount of blending using the method introduced in Ricci (1973), and it achieves a large range of blends. Standard blending is referred to as $B_a + B_b$ (i.e. the sum of the functions $f_{B_a}$ and $f_{B_b}$). Super elliptic blending will be denoted as $B_a \diamond_n B_b$ and is defined as:

$$f_{B_a \diamond_n B_b} = \left( {f_{B_a}}^n + {f_{B_b}}^n \right)^{\frac{1}{n}} . \tag{4}$$

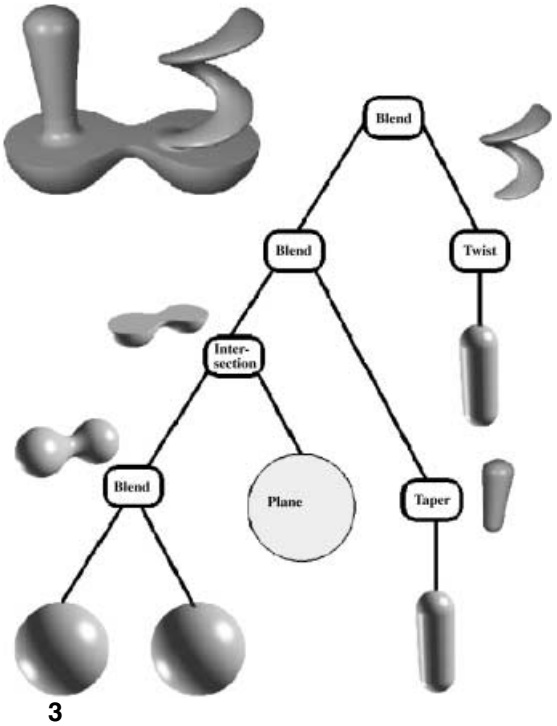
The standard blending operator $+$ is a special case of Eq. (4) with $n = 1$. Moreover:

$$\lim_{n \to +\infty} \left( {f_{B_a}}^n + {f_{B_b}}^n \right)^{\frac{1}{n}} = \max \left( f_{B_a}, f_B \right) . \tag{5}$$

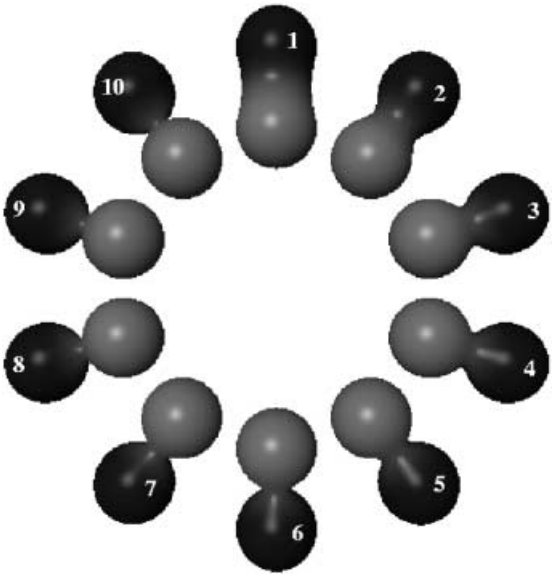
Thus, when $n$ varies from 1 to infinity, it creates a set of models interpolating between blending $A + B$ and union $A \cup B$. Figure 4 shows a series of blends where $n$ is varied between $n = 1$ and 10, which illustrates this effect.

This generalized blending is associative, i.e. $f_{(B_a \diamond_n B_b) \diamond_n B_c} = f_{B_a \diamond_n (B_b \diamond_n B_c)}$. Figure 3 shows the nodes to be binary or unary, but the binary nodes can easily be extended using the above formulation to n-ary nodes.

*Controlled blending* allows us to blend one *BlobTree*, $B_a$ with another *BlobTree*, $B_b$, and to blend $B_b$ with




3




4

**Fig. 3.** A Sample *BlobTree*
**Fig. 4.** Super-elliptic blending

a third *BlobTree*, $B_c$, without blending $B_a$ with $B_c$, as described in Guy and Wyvill (1995). It is defined here as

$$c(b_1, b_2, \ldots, b_n)(B_1, B_2, \ldots, B_m)\,,$$
$$b_i = (j_i, k_i),\ j_i, k_i \in \{1, \ldots, m\} \quad \forall i \in \{1, \ldots, n\}\,, \tag{6}$$

where $m$ *BlobTrees* are being included in the controlled blend, and each $b_i$ defines a blend between *BlobTree* $B_{j_i}$ and *BlobTree* $B_{k_i}$. In the current implementation blends within a controlled blend are limited to pairwise blends and super-elliptic blending is not available; however these can easily be added to the *BlobTree*.

## 3 Modeling *Murex cabritii*

To model *Murex cabritii* requires a description of the parts of the shell. The model is derived from observations made from Fig. 1, and from a textual description of the shell found in Rehder (1981) which describes the following features:

- 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 four to five spines.
- Each whorl has three varices (ridges) which bear several sharp-curving spines.
- Beaded axial riblets (or small bumps) are present between varices.

For the remainder of this paper a whorl is defined as a part of the main body formed by a rotation about the axis of the shell, beginning immediately after a varix, and ending after three varices have been formed. From Fig. 1 we have estimated that a whorl corresponds to a rotation of 348° about the axis of the shell, thus the angle between successive varices is equal to 116°.

The shell was modeled with seven whorls. Five to six spines were modeled in the axial rows rather than four to five as described above. The bumps occur periodically both parallel and perpendicular to the helico-spiral. Five sets of bumps were added along the helico-spiral between each pair of varices. The $y$-axis in the standard coordinate system is defined as the axis of rotation of the shell for the remainder of this paper. The following parameters were used to define the helico-spiral for the model:

$$\beta = 22.5^\circ ,$$
$$\rho = 1.3 ,$$
$$R_0 = 0.2 ,$$
$$D(\theta) = \frac{R(\theta)}{\sin\beta}\frac{\rho - 1}{\rho + 1} = 0.341 \cdot R(\theta) . \quad (7)$$

Construction of the implicit model of *Murex cabritii* will be discussed next. Section 3.1 describes building the main body whorl of the shell. Adding the spines and bumps to the shell is discussed in Sect. 3.2. Creating an opening in the shell is described in Sect. 3.3, and the application of 2D textures is discussed in Sect. 3.4.

### 3.1 Main body whorl

The formulas in Sect. 2.1 can be used to calculate position [Eqs. (1) and (2)] and size Eq. (3) of a generating curve along a helico-spiral, so that successive curves placed along the helico-spiral and connected in a polygonal mesh approximate the surface of a shell. Fowler et al. (1992) used piecewise Bezier curves to construct generating curves, which were applied to model a great variety of shells.

We used a similar method to create the implicit model. A generating implicit surface was described using a skeletal implicit point primitive. The placement of an instance of the generating surface on the helico-spiral at any angle, $\theta$, was performed in three steps:

1. Scale by $D(\theta)$ Eq. (3).
2. Translate by ($R(\theta)$,$H(\theta)$,0) [Eqs. (1) and (2)].
3. Rotate by $\theta$ about the $y$-axis.

Equation (8) defines the function $B_\theta = P(B, \theta)$ which takes an arbitrary *BlobTree*, $B$, and returns a new *BlobTree*, $B_\theta$, which tranforms $B$ as described above.

$$B_\theta = P(B, \theta) ,$$
$$P(B, \theta) = r(Y, \theta)\big(T(B, \theta)\big) ,$$
$$T(B, \theta) = t\big(R(\theta), H(\theta), 0\big)\big(S(B, \theta)\big) ,$$
$$S(B, \theta) = s\big(D(\theta), D(\theta), D(\theta)\big)(B) . \quad (8)$$

Equation (9) describes the *BlobTree* for a whorl ${B_{\mathrm{w}}}_a^b$, where $\theta_\mathrm{s}$ is the interval between adjacent instances of the generating surface defined by the *BlobTree* $B_\mathrm{g}$, $a$ and $b$ define the start and end of the whorl, and $b - a + 1$ is the number of generating surfaces used in the whorl:

$${B_{\mathrm{w}}}_a^b = \sum_{i=a}^{b} B_{\mathrm{g}_i} ,$$
$$B_{\mathrm{g}_i} = P(B_\mathrm{g}, \theta_\mathrm{s} * i) . \quad (9)$$

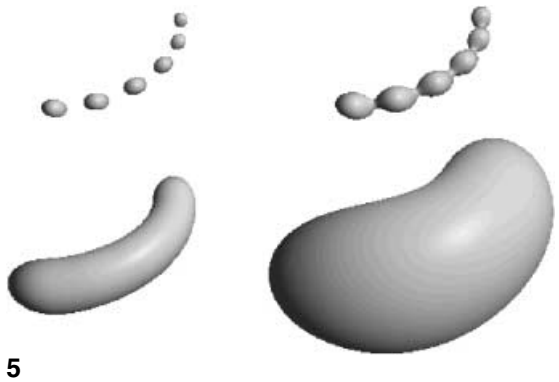
The symbol $\sum$ is used to represent the blend of multiple *BlobTrees*. Consecutive surfaces along the whorl are automatically blended together. Figure 5 shows a series of point primitives placed along a helico-spiral: as the field defined by each primitive is increased, the resulting surface tends toward a shell whorl with a circular aperture.

To avoid unwanted blending between consecutive whorls, controlled blending was used. A whorl consists of three sections, each of which is contained between two successive varices and corresponds to a 116° rotation about the axis of the shell. A section was created by placing six instances of the generating surface on the helico-spiral using ${B_w}_a^{a+5}$ from Eq. (9) with $\theta_\mathrm{s} = \frac{116}{6}^\circ$. To create the whole body $B_m$ with seven complete whorls, 21 sections are combined using controlled blending Eq. (6), as shown in the following:

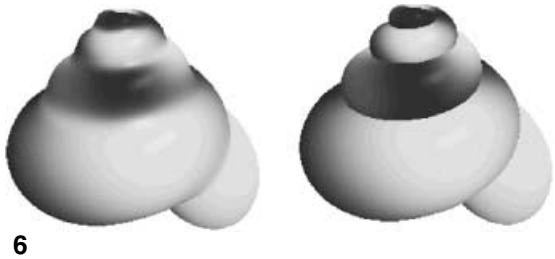
$$B_m = c(L_{\mathrm{blendpairs}})(L_{Blobtrees}) ,$$
$$L_{\mathrm{blendpairs}} = \{(i, i + 1) , i \in \{1, 2, \ldots , 20\}\} ,$$
$$L_{Blobtrees} = \{{B_{\mathrm{w}}}_i^{i+5} , i \in \{0, 6, 12, \ldots , 120\}\} . \quad (10)$$

Thus, each section is blended with its two immediate neighbours, but not with any other sections. The resulting surface is smooth along the helico-spiral, but adjacent whorls do not blend together. Figure 6 shows the effect of controlled blending, using a point primitive as the generating surface.

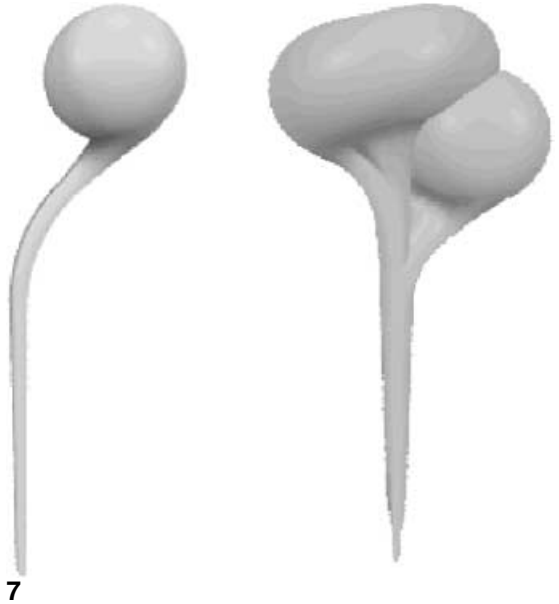
To incorporate the long slender canal below the main body whorl, a cone primitive, bent with a warp operator, was placed below the point primitive. The



5



6



7

**Fig. 5.** Six point primitives placed on a helico-spiral. As the size of the field produced by each primitive increases, the resulting surface forms part of the main body whorl of a shell

**Fig. 6.** Each whorl of a shell is composed of three sections (shown in Fig. 5). On the *left* all sections blend with all other sections; on the *right* controlled blending constrains each section to blend only with its two neighbours along the helico-spiral

**Fig. 7.** On the *left* is the generating surface used for the model of *Murex cabritii*; on the *right* is the whorl this surface defines

generating surface and the resulting whorl it defines are shown in Fig. 7. To reduce the complexity of the model, the canal was only modeled in the last $1\frac{1}{3}$ whorls where it could be seen.

### 3.2 Adding varices, bumps and spines

*Varices* are the spiny ridges extending out from the main body whorl at even intervals of 116° around the axis of the shell. The varix $B_v$ was modeled as a series of curving spines of varying size, with the relative size and location of each spine within a varix determined separately for each whorl (Table 1). Individual spines were modeled using cone primitives bent by 30° using a warp operator. The placement of each spine is given by

$$
\begin{aligned}
B_{\mathrm{s}} &= \sum_{i=1}^{n} r(Z, \alpha_i)\big(T_k(B_k)\big)\,, \\
T_k(B_k) &= t(x_k, 0, 0)\big(S_k(B_k)\big)\,, \\
S_k(B_k) &= s(\delta_i, \delta_i, \delta_i)(B_k)\,,
\end{aligned}
\tag{11}
$$

for a series of spines $B_{\mathrm{s}}$, where $n$ defines the number of spines, $B_k$ defines a spine lying on the $x$-axis with

**Table 1.** Relative size [$\delta_i$ in Eq. (11)] of curving spines at each of 3 varices per whorl in the model of *Murex cabritii*. Angle indicates rotation in the plane of the generating curve from a horizontal orientation

| | Whorl | | | | | | |
|---|---|---|---|---|---|---|---|
| Angle (°) | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| −90 | | | | | | | 0.55 |
| −80 | | | | | | | 0.64 |
| −70 | | | | | | | 0.55 |
| −60 | | | | | | | 1.14 |
| −50 | | | | | | | 0.55 |
| −40 | | | | | | | 1.92 |
| −30 | | | | | | | 0.55 |
| −20 | | | | | | | 0.82 |
| −10 | | | | | | 0.55 | 0.55 |
| 0 | | | | | | 0.8 | 1.44 |
| 10 | | | | | | 1.55 | 0.64 |
| 20 | | | | | 0.9 | 0.64 | 0.64 |
| 30 | | | | | 0.7 | 0.7 | 0.55 |
| 40 | | | | 0.8 | 1.5 | 0.55 | 1.62 |
| 50 | | 0.8 | 0.9 | 1.5 | 0.64 | 0.96 | 0.72 |
| 60 | | | | 0.7 | 1.06 | 0.6 | 0.64 |
| 70 | | | | | | | 0.55 |
| 80 | | | | | | | 0.5 |

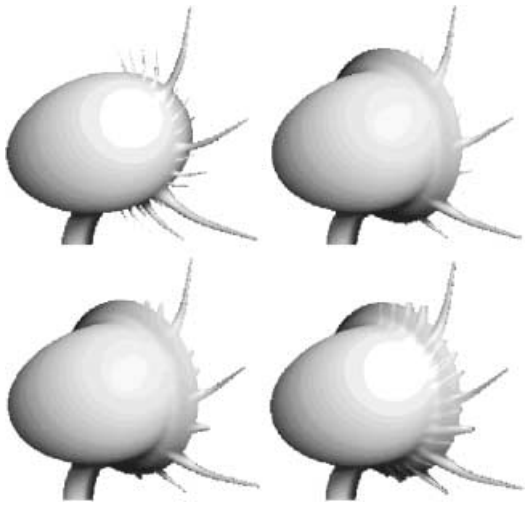


**Fig. 8.** Creation of a varix. *Top left*: bent cones are placed as curved spines. *Top right*: two concentric tori blend spines together. *Bottom left*: smaller spines are modeled with wider tips. *Bottom right*: All spines scaled by $\delta_{\max}$ along helico-spiral Eq. (12)

its base at the origin and the tip bent in the direction of the positive $y$-axis, $x_k$ is the distance to the edge of the shell, and $\alpha_i$ and $\delta_i$ are given by Table 1.

Most of the spines in the varix of *Murex cabritii* are not free standing, but are blended together in a ridge. Two concentric torus primitives were added to connect the spines to each other near the shell surface. To make the shorter spines stand out from the ridge, they were modeled with a thicker top, and when scaling the spines by the $\delta_i$ (from Table 1), $\delta_{\max}$ (the maximum value of all $\delta_i$ from Table 1) was used to scale each spine along the $z$-axis to increase the width of spines across the varix. Equation (12) shows these operations. $B_{\mathrm{k}i}$ is a bent spine with a variable width of tip and defines the *BlobTree* for a varix $B_{\mathrm{v}}$; $B_{\mathrm{r}}$ is the *BlobTree* for the two tori. The effect of each of these operations can be seen in Fig. 8.

$$
\begin{aligned}
B_{\mathrm{s}} &= \sum_{i=1}^{n} r(Z, \alpha_i)\big(T_{\mathrm{k}}(B_{\mathrm{k}i})\big)\,, \\
T_k(B_{\mathrm{k}i}) &= t(x_k, 0, 0)\big(S_k(B_{\mathrm{k}i})\big)\,, \\
S_k(B_{\mathrm{k}i}) &= s(\delta_i, \delta_i, \delta_{\max})(B_{\mathrm{k}i})\,, \\
B_{\mathrm{v}} &= B_{\mathrm{s}} + B_{\mathrm{r}}\,. \qquad (12)
\end{aligned}
$$

To include a varix at an arbitrary position along the helico-spiral, $B_{\mathrm{v}}$ is placed using Eq. (8):

$$B_{\mathrm{v}\theta} = P(B_{\mathrm{v}}, \theta)\,. \qquad (13)$$

*Bumps* were modeled using single point primitives which were scaled by $(\zeta_x, \zeta_y, \zeta_z + 0.8)$, where $\zeta_d = n(s, \frac{s}{10})$, $d \in \{x, y, z\}$, $s$ is the default size of a bump and $n(\mu, \sigma)$ returns a pseudo-random number with a normal distribution in which $\mu$ is the mean and $\sigma$ is the standard deviation. The number of bumps in each set of bumps is determined by the number of spines defined for the current whorl. One bump was placed for every second curved spine using the same method employed to place the curved spines Eq. (11).

Super-elliptic blending was employed to blend the bumps with the surface of the shell. This was required to avoid the tendency of the whorl surface to blend too smoothly with the bumps, as can be seen in Fig. 9. To create a much more abrupt blend, a value of $n = 400$ was used in Eq. (4). Such a high value of $n$ was required due to the fact that the implicit primitives defining the whorl defined a much larger and stronger field than that produced by the bump primitives. Figure 9 shows two whorl sections with five

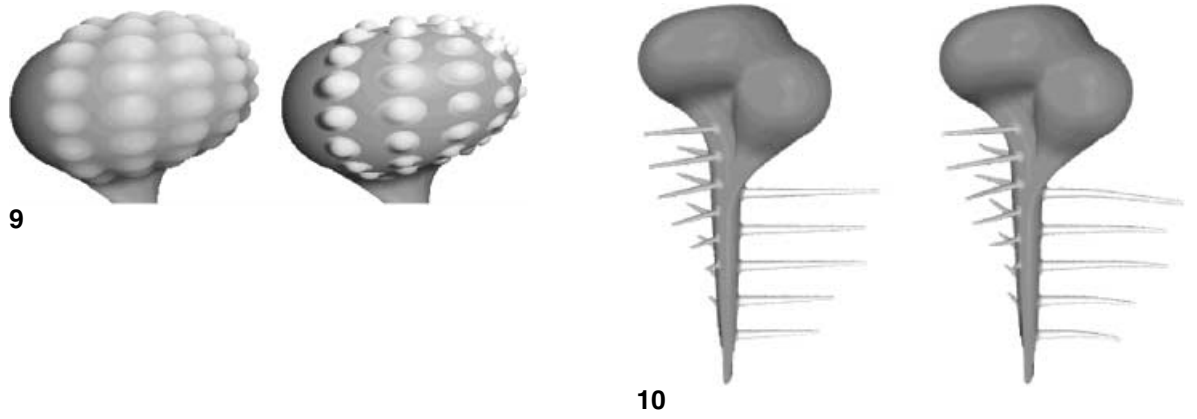


**Fig. 9.** Creation of bumps. *Left*: similar bumps blended to shell using the + operator. *Right*: randomly scaled bumps blended to shell with $\diamond_{400}$ operator. The value 400 was required due the great disparity in strength and extent of field between the large whorl and small bumps

**Fig. 10.** Axial rows of 5–6 spines. *Left*: spines are straight. *Right*: each spine randomly bent 3° to 9° 1–3 times

sets of bumps on them, one with regular blending and no random scaling, and the other employing both super-elliptic blending and random scaling.

*The axial rows of spines* protruding from the lower canal were modeled using cone primitives. There is one row of spines below each of the three varices on the last whorl of the shell. The spines were placed at even intervals from each other along the canal. Three instances of the spines were then transformed using Eq. (8) using the same angle at which the last three varices are formed on the main body of the shell. The relative sizes and number of spines were determined separately for each row, as specified in Table 2.
The spines are not perfectly straight in nature, so each spine was randomly bent by 3° to 9° one to three times using a warp operator. The spines can be seen in Fig. 10 with and without the random bending warps.

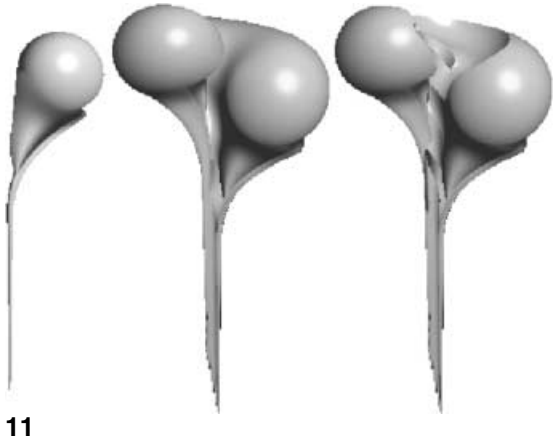
**Table 2.** Relative size of axial spines below last 3 varices in the model of *Murex cabritii*. Varix number corresponds to the order in which they were formed (e.g. varix 3 is the last varix formed and is at the opening of the shell)

| | Varix | | |
|---|---|---|---|
| Spine | 1 | 2 | 3 |
| 1 | 0.96 | 0.95 | 1.00 |
| 2 | 0.92 | 0.99 | 0.90 |
| 3 | 0.84 | 0.76 | 0.92 |
| 4 | 0.68 | 0.51 | 0.70 |
| 5 | 0.45 | 0.35 | 0.60 |
| 6 | 0.25 | 0.28 | |

### 3.3 Creating an opening

Combining the elements described thus far provides a good approximation of the exterior of a *Murex cabritii* shell. To construct an opening in the shell, a solid model, $B_{\text{opening}}$, was constructed which matched the shape of the hollow portion of the shell. A CSG difference operation removed $B_{\text{opening}}$ from the model of the shell, creating the interior space.

$B_{\text{opening}}$ was created using the same method described for the main body whorl. A similar generating surface was created which was slightly smaller in each dimension orthogonal to the helico-spiral. The basis of the generating surface was formed from a single point primitive, slightly smaller than that of the main whorl. The inner canal was modeled by four slender cones, bent as in the main body whorl's generating surface. Four cones were required because a single cone was too slender to describe a sufficient arc of the whorl. Four additional cones were used to extend the inner edge of the surface to the edge of the previous whorl. Equation (14) shows how the opening was carved out of the shell. $B_{\text{shell}}$ is the complete shell without the opening; $B_{\text{inside}}$ defines a whorl generated with the generating surface for the





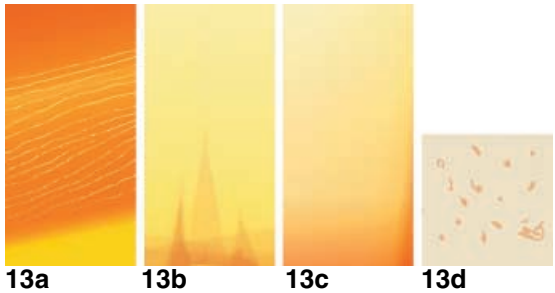



**Fig. 11.** On the *left* is the generating surface used for the inside of *Murex cabritii*; in the *center* is the whorl this surface defines; and on the *right* is the final surface which will be cut out of the main shell to create the opening

**Fig. 12.** On the *left* is the opening which is carved out by Eq. (14); on the *right* is the final shape of the opening after adding in $B_{\text{inside wall}}$ as in Eq. (15)

**Fig. 13a–d.** Textures and their corresponding uses in the model of *Murex cabritii*: **a** Main body whorl; **b** Spines in varices; **c** Axial rows of spines; **d** Bumps on main whorl

inside of the shell; and $B_{\text{w}a}^{\,b}$ is from Eq. (9):

$$B_{Murex} = B_{\text{shell}} - B_{\text{opening}} ,$$
$$B_{\text{opening}} = B_{\text{inside}} - B_{\text{w94}}^{108} . \qquad (14)$$

The previous whorl is subtracted from the inside whorl to create $B_{\text{opening}}$, which in turn is subtracted from the main body whorl. This keeps the previous whorl intact. The generating surface used to create the hollow portion of the shell, the whorl it defines, and the surface which is used to cut out the opening from the main shell are shown in Fig. 11.

Observation of sea shells similar to *Murex cabritii* (*Murex troschel*) reveal that the opening is roughly circular. The opening which is carved out in Eq. (14) is not circular. An inside wall was modeled separately then added into the model after carving out the opening. Equation (15) shows the final combination which was used to define $B_{Murex}$, where $B_{\text{inside wall}}$ is the *BlobTree* for the interior wall. The opening with and without the inside wall is shown in Fig. 12.

$$B_{Murex} = (B_{\text{shell}} - B_{\text{opening}}) \cup B_{\text{inside wall}} . \qquad (15)$$

### 3.4 Texturing the shell

The model captures most of the form of *Murex cabritii*, but to get a more realistic image four 2D textures and two separate texturing methods were applied to the model. The textures are shown in Fig. 13, and were all created using standard paint programs.

The main body whorl is textured in parts using the method introduced in Tigges and Wyvill (1998).

This method works by first mapping the texture to a bounding parametric surface, $S$, with a known 2D parameterization. The $uv$ coordinates for texturing on any point $p_i$ on the implicit surface $I$ are determined by following a combination of the gradient of the field, and a vector normal to $S$, towards $S$. When $S$ is reached at point $p_s$, the $uv$ coordinates for point $p_s$ are used for texturing $p_i$.

Figure 13a shows the texture applied to each section in $L_{Blobtrees}$ [defined in Eq. (10)]. The parts of the main body whorl where two textured sections are blending are positioned so that they are covered by a varix. This conceals discontinuities in the pattern resulting from by the blending of two sections separately textured with the same texture map.

The texturing method described above is computationally expensive, but was required to achieve the desired effect on the main whorl. A faster method of texturing was used for all of the other textured parts of the model. These include the spines in the varices, the spines below the main body whorl and the bumps on the main body whorl. In this method textures are mapped to a known 2D parameterization of each of the primitives. When blending two or more textured *BlobTrees*, the resulting colour at a point, $P$, in space is determined using a linear combination of the colour of each *BlobTree* scaled by its field value at point $P$. This method is fully described in Tigges (1999).

An inherent feature of the texturing methods implemented in the *BlobTree* is that all of the textures are automatically blended with each other. This gives our model a natural look where separately textured parts of the model are joined. The use of all of these textures can be seen in Fig. 14.

**Fig. 14.** Model of *Murex cabritii*

## 4 Results and conclusion

We have presented an application of a structured modeling technique that combines implicit surfaces (CSG) and 2D texture mapping. This combination of techniques is incorporated in an implementation of the *BlobTree* developed at the University of Calgary (Wyvill et al. 1998). A procedural interface is available for the description of models, allowing exact and concise definition of models which are easily manipulated. Specifically the equations for *BlobTrees* shown in this paper were implemented in the Python programming language, which allowed us to create functions that defined each individual part of the shell and then combine these functions in other functions until we had a function which defined the entire sea shell.

Using the *BlobTree*, a realistic model of *Murex cabritii* was built (Fig. 14). This model not only demonstrates that implicit surfaces are a valid choice for modeling natural forms, but that they are capable of creating models where traditional methods fail. Specifically, large protrusions on a sea shell's surface have been captured by switching from a parametric to an implicit definition of the shell form.

Our model does not rely on polygon mesh operations, it is resolution independent and can be rendered directly using ray tracing. Figure 14 took approximately 2 h to ray trace on a cluster of fourteen 500 MHz DEC Alphas. This represents a significant amount of computation. Work is under way to improve the efficiency of our rendering methods.

The following areas of the model remain open to improvement: the opening was modeled by observing the opening on similar shells (*Murex troschel*); the position and number of spines and bumps were based on a single view of the shell; the number and

placement of these features were arbitrary and suddenly change from one whorl to another; the textures were created in a paint program and pasted on to give a good approximation only; the varices do not extend to the lower canal.

The controlled blending used in the current model does not allow much flexibility. The blends can only be specified between two objects at once, and super elliptic blends are not allowed. An exploration of this area might prove quite fruitful. A major extension of the model would be to use reaction–diffusion techniques to place spines and bumps on the shell.

*Acknowledgements.* We would like to thank Mark Tigges and Andrew Guy, for providing the tools with which this model was built, and the MACI project, for high-performance computing, who made available to us the DEC Alpha cluster. This work was partially supported by grants from the Natural Sciences and Engineering Research Council of Canada.

## References

1. Crespin B, Blanc C, Schlick C (1996) Implicit sweep objects. Proc. Eurogr. '96 15:165–174
2. Fowler DR, Meinhardt H, Prusinkiewicz P (1992) Modeling seashells. SIGGRAPH '92 Conference Proceedings. ACM, New York, pp 379–387
3. Guy A, Wyvill B (1995) Controlled blending for implicit surfaces. Proc. Implicit Surf. '95 1:107–112
4. Kawaguchi Y (1982) A morphological study of the form of nature. Comput Graph 16(3):223–232
5. Meinhardt H (1995) The algorithmic beauty of sea shells. Springer, New York
6. Pasko A, Adzhiev V, Sourin A, Savchenko V (1995) Function representation in geometric modeling: concepts, implementation and applications. Visual Comput 11(8):429–446
7. Rehder HA (1981) The Audobon Society field guide to North American seashells. Knopf, New York
8. Ricci A (1973) Constructive geometry for computer graphics. Comput J 16(2):157–160
9. Tigges M, Wyvill B (1999) A field interpolated texture mapping algorithm for skeletal implicit surfaces. In: Proceedings of Computer Graphics International '99, IEEE Computer Society, Los Alamitos, CA, pp 25–33
10. Tigges M, Wyvill B (1998) Texture mapping the BlobTree. Proc. Implicit Surf. '98 3:123–130
11. Wyvill B (1988) The great train rubbery. SIGGRAPH '88 Electron. Theatre Video Rev., Issue 26
12. Wyvill B, van Overveld K (1996) Polygonization of implicit surfaces with constructive solid geometry. J Shape Model 2(4):257–274
13. Wyvill B, Guy A, Tigges M (1998) Jungle software projects; available at http://www.cpsc.ucalgary.ca/~jungle/software/index.html
14. Wyvill B, Galin E, Guy A (1999) Extending the CSG tree. Warping, blending and Boolean operations in an implicit surface modeling system. Comput Graph Forum 18(2):149–158

CALLUM GALBRAITH is a PhD student at the University of Calgary. He holds a BSc in Computer Science from the University of Calgary. His research is in the area of modeling of natural phenomena for computer graphics purposes, focusing on the application of implicit surfaces to this domain.

PRZEMYSLAW PRUSINKIEWICZ is a professor of computer science at the University of Calgary. He holds an MSc and PhD, both in Computer Science, from the Technical University of Warsaw. His research is focused on the modeling, simulation, and visualization of biological structures, in particular plants. He is a co-author of two books, "The Algorithmic Beauty of Plants", and "Lindenmayer Systems, Fractals, and Plants", and numerous research papers in this area.

BRIAN WYVILL is a professor of computer science at the University of Calgary. He obtained his BSc in Physics from the University of London and PhD in Computer Graphics from Bradford University. He has been involved in computer animation since the mid-1970's including some sequences for the movie, Alien. His recent research interests have centred around implicit modelling.