Modeling lobed leaves

Lars Mündermann, Peter MacMurchy, Juraj Pivovarov and Przemyslaw Prusinkiewicz
Department of Computer Science, University of Calgary
Calgary, Alberta, T2N 1N4 Canada
lars|peterm|juraj|pwp@cpsc.ucalgary.ca

Abstract

In contrast to the extensively researched modeling of plant architecture, the modeling of plant organs largely remains an open problem. In this paper, we propose a method for modeling lobed leaves. This method extends the concept of sweeps to branched skeletons. The input of the model is a 2D leaf silhouette, which can be defined interactively or derived from a scanned leaf image. The algorithm computes the skeleton (medial axis) of the leaf and approximates it using spline curves interconnected into a branching structure (sticky splines). The leaf surface is then constructed by sweeping a generating curve along these splines. The orientation of the generating curve is adjusted to properly capture the shape of the leaf blade near the extremities and branching points of the skeleton, and to avoid selfintersections of the surface.

The leaf model can be interactively modified by editing the shape of the silhouette and the skeleton. It can be further manipulated in 3D using functions that control turning, bending, and twisting of each lobe.

Reference

L. Mündermann, P. MacMurchy, J. Pivovarov, P. Prusinkiewicz: Modeling lobed leaves. *Proceedings of Computer Graphics International − CGI 2003*, pp. 60−65.

Modeling lobed leaves

Lars Mündermann, Peter MacMurchy, Juraj Pivovarov and Przemyslaw Prusinkiewicz
Department of Computer Science, University of Calgary
Calgary, Alberta, T2N 1N4 Canada
lars|peterm|juraj|pwp@cpsc.ucalgary.ca

Abstract

In contrast to the extensively researched modeling of plant architecture, the modeling of plant organs largely remains an open problem. In this paper, we propose a method for modeling lobed leaves. This method extends the concept of sweeps to branched skeletons. The input of the model is a 2D leaf silhouette, which can be defined interactively or derived from a scanned leaf image. The algorithm computes the skeleton (medial axis) of the leaf and approximates it using spline curves interconnected into a branching structure (sticky splines). The leaf surface is then constructed by sweeping a generating curve along these splines. The orientation of the generating curve is adjusted to properly capture the shape of the leaf blade near the extremities and branching points of the skeleton, and to avoid self-intersections of the surface.

The leaf model can be interactively modified by editing the shape of the silhouette and the skeleton. It can be further manipulated in 3D using functions that control turning, bending, and twisting of each lobe.

(a) (b)

Figure 1. Modeling a white oak leaf. (a) Scanned leaf. (b) Deformed leaf surface compared to outline of the scanned leaf.

1 Introduction

One of the challenges in computer graphics is to create the geometry of objects in an intuitive and direct way, while allowing for interactive manipulation of the resulting shapes. This is particularly important in the synthesis of realistic images of plants, where a detailed representation of a plant component's shape is "vital in capturing the character of a species" [16]. While the modeling of plant structures is well-researched, the modeling of individual plant organs has so far remained relatively unexplored.

We present an interactive method for modeling lobed leaves that extends the concept of sweeps to branching structures. The modeling process begins with a 2D silhouette of a leaf, which can be scanned or defined interactively using a curve editor. The silhouette's skeleton is then defined interactively or computed automatically using a 2D Voronoi diagram [6], and is represented as interconnected "sticky" splines [15] that preserve topological relations during subsequent manipulation. The leaf surface is constructed by sweeping a cross-section between the skeleton and the silhouette. The planar leaf model obtained in the previous steps can subsequently be deformed in 3D space using functions that control its turning, bending and twisting along the axes of the skeleton (Figure 1).

In Section 2, we review previous work on the modeling of leaves. We then describe our proposed method in Section 3. In Section 4, we discuss the surface representation in turtle geometry and its free-form deformation. We conclude the paper with a presentation of examples in Section 5, followed by a discussion of the results and open problems for further research in Section 6.

2 Previous work

Frequently, plant components such as stems and leaves are modeled using *sweeps*. In the simplest case, a 2D generating curve is swept along a linear path normal to the plane of the curve, producing an *extrusion*. Alternatively, a 2D curve can be swept around an axis, producing a *surface of revolution*. The generator is not limited to 2D and, in the case of a general sweep, can change size, orientation, or shape.

Generalized cylinders [3] were originally applied to tree branch modeling by Bloomenthal [4], and have often been used to model plant organs since then. A generalized cylinder is formed by sweeping a (not necessarily planar) generating curve, which determines the organ's *cross-section*, along a trajectory (*carrier curve*) that defines the organ's axis [14]. The generating curve may be closed, as is typically the case for stems, or open, as for thin leaves. Furthermore, it can be scaled according to a profile curve and may change shape while being swept.

Several researchers have used generalized cylinders to model leaves supported by a single axis [8, 12]. More complex leaves have been modeled by constructing a leaf surface using spline patches [4, 11], or by calculating an implicit contour around a branching structure [7].

3 Shape generation

Our approach to shape generation is similar to Snyder's *rail product* [14]. A rail product is formed by sweeping a planar generating curve between two *rail curves*. Intuitively, one can imagine a railroad track: the rail curves are analogous to the track's rails, and the cross-sections are analogous to the railroad ties.

We extend Snyder's approach to a situation in which one of the curves may have an arbitrary branching topology: it is the branching skeleton of a given silhouette. The surface is constructed by sweeping a generating curve between the silhouette and the branching skeleton. Each element of the skeleton is traversed twice: once in the distal and once in the basal direction. In the simplest case, the generating curve is a straight line segment. Other curves may also be used, as long as they do not self-intersect during the sweep. At the end of the modeling process, the surface can be extended to 3D by deforming (turning, bending, and twisting) the skeleton.

3.1 Silhouette

The minimal input to the model is a 2D leaf silhouette. It can be specified interactively, using a B-spline curve editor, or derived from a scanned image of the leaf surface. In the latter case, a B-spline representation of the silhouette is obtained algorithmically, using the process described below.

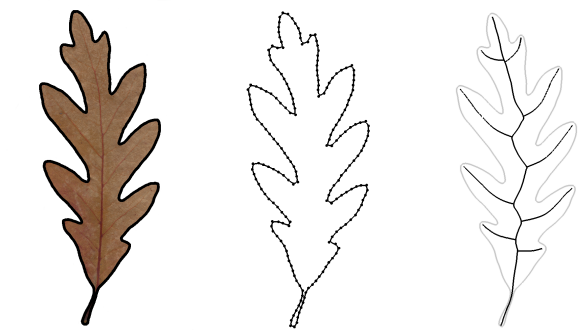

Figure 2. Modeling a white oak leaf. (a) Scanned leaf with emphasized silhouette. (b) Silhouette with a reduced number of points. (c) Skeleton.

First, the leaf is scanned using a standard flat-bed scanner (Figure 1(a)). The resulting image is converted to black and white using a threshold chosen so that the leaf shape stands out from the background. The leaf silhouette is then computed using a boundary-following algorithm [9]. The output of this algorithm is a silhouette representation in the form of a sequence of pixels, which may be thought of as vertices of pixel-size linear segments (Figure 2(a)). In order to create a more compact representation, we reduce the number of vertices using inverse curve subdivision [13]. Curve subdivision algorithms refine a coarse set of points that approximates a smooth curve and produce a finer approximation with a larger number of points. Inverse subdivision proceeds in the opposite direction: given a finer approximation it creates a coarser one, with a smaller number of points. The resulting vertices are then used as control points of a B-spline curve, which we use as a representation of the leaf silhouette (Figure 2(b)).

3.2 Skeleton

The object's skeleton is a branching structure (Figure 2(c)). To represent it, we use *sticky splines* [15], which extend the concept of splines to branching structures. Sticky splines maintain topological relations between different interconnected spline curves when the shape of individual curves changes, for example as a result of editing.

The skeleton can be constructed interactively, for example, using the the main veins of a leaf as a guide. However, a user must make sure that the skeleton meets a set of somewhat non-intuitive criteria to prevent holes and self-intersections in the generated surface. For this reason, the

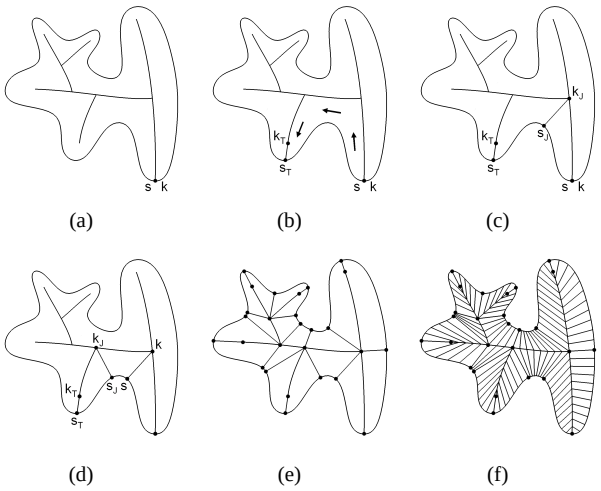

Figure 3. Segmentation process during sweeping.

user may prefer to have the skeleton generated automatically. In this case, the holes and self-intersections are prevented automatically, but, in general, the branches no longer correspond to leaf veins.

We implemented a skeleton generator based on the concept of the medial axis. The medial axis was introduced by Blum [5] as a technique of object representation. For a planar figure, it defines the loci of inscribed discs of maximal diameter that maintain contact with at least two points on the curve. The medial axis is closely related to Voronoi diagrams and there are various ways to construct it [2]. In our work, we have used Fortune's [6] sweepline algorithm to calculate an approximation to the medial axis. Typically, this algorithm generates a large number of points. We reduce the number of points using inverse subdivision [13], as outlined in Section 3.1.

3.3 Sweeping to Construct the Mesh

The leaf surface is constructed by sweeping a cross-section using the skeleton and the silhouette as the rails. Positions of the cross-section are determined by points on each of the two curves. We split the underlying topology into segments defined using *branching points* of the skeleton and *anchor points* along the silhouette curve, to minimize the impact of the shape of one lobe on the segmentation of other lobes.

The silhouette is represented as a 2D closed B-spline and the skeleton as interconnected B-splines. Let k and s define starting points on the skeleton and silhouette, respectively (Figure 3(a)). We search the skeleton in a clockwise, distal direction for the leftmost terminal point k_T (Figure 3(b)). We then calculate a corresponding anchor point s_T on the silhouette, determined by intersecting the tangent to the skeleton at point k_T with the silhouette.

Next, starting from point k, we traverse the skeleton one more time in the same direction until we reach either a branching point or the terminal point k_T. For every visited branching point k_J, we calculate a corresponding anchor point s_J on the silhouette (Figure 3(c)). The anchor point s_J is the point of the segment $[s, s_T]$ of the silhouette curve that is closest to k_J. The points (k, k_J, s_J, s) form a *segment*. We now reset the variables k to k_J and s to s_J and repeat previous steps, finding new pairs of branching and anchor points (Figure 3(d)), until we reach the terminal point k_T. The set of points (k, k_T, s_T, s) defines the terminal segment.

The process continues in a clockwise, basal direction along the current branch of the skeleton. We reassign k to k_T and s to s_T, and traverse the current branch towards its base until we reach either the first junction point that supports a right branch or the base of the branch. The segmentation process continues in a clockwise, distal direction along the skeleton until the base of the main branch of the skeleton is reached (Figure 3(e)).

Finally, we calculate the equally spaced railroad ties within each segment using arc-length parametrization of the skeleton and silhouette curves (Figure 3(f)). Each railroad tie T_i connects a point $k_i = (k_{i,x}, k_{i,y})$ on the skeleton with an endpoint $s_i = (s_{i,x}, s_{i,y})$ on the silhouette. The set of railroad ties is denoted $G = \{T_0, T_1, \ldots, T_n\}$, where $T_i = (k_i, s_i) = (k_{i,x}, k_{i,y}, s_{i,x}, s_{i,y})$. The railroad ties, together with the corresponding segments of the skeleton and silhouette curves, constitute the leaf's geometry when connected into a strip of quadrilaterals.

3.4 Texture mapping

The scanned image of the leaf also provides texture information (Figure 4(c)). In order to make use of it, we store the texture coordinates $k_{i,u}$, $k_{i,v}$, $s_{i,u}$, and $s_{i,v}$ with the railroad ties. The texture coordinates are calculated as:

$$\begin{aligned} k_{i,u} &= k_{i,x}/I_w, & k_{i,v} &= k_{i,y}/I_h, \\ s_{i,u} &= s_{i,x}/I_w, & s_{i,v} &= s_{i,y}/I_h, \end{aligned} \tag{1}$$

where I_h and I_w represent the height and width of the texture image.

We also take advantage of the texture's transparency channel to make all pixels outside the leaf surface transparent. This prevents the background from showing up near the boundaries of the leaf in case of inaccuracies in the texture mapping.

(a) (b) (c)

Figure 4. Modeling a white oak leaf.
(a) Silhouette and skeleton.
(b) Leaf mesh.
(c) Textured surface.

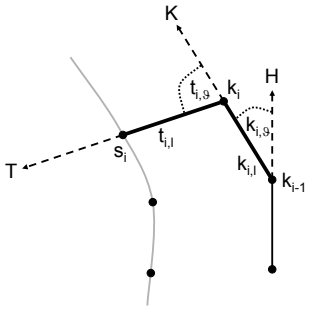

Figure 5. Turtle parameters.

4 Free-form deformation

The mesh description is transformed into a turtle geometry representation [1, 11] for the purpose of interactive free-form deformation of the models. The *turtle's state* is characterized by a position vector $\vec{P}$ and three orientation vectors $\hat{H}$, $\hat{L}$, and $\hat{U}$ that indicate the turtle's *heading*, the direction to the *left*, and the *up* direction. These vectors have unit length, are perpendicular to each other, and satisfy the equation $\hat{H} \times \hat{L} = \hat{U}$. Rotations of the turtle are expressed by the equation $[\hat{H}' \quad \hat{L}' \quad \hat{U}'] = [\hat{H} \quad \hat{L} \quad \hat{U}] \cdot R$, where R is a 3×3 rotation matrix. Specifically, rotations by angle ϑ about vectors $\hat{H}$, $\hat{L}$, and $\hat{U}$ are represented by the matrices $R_H(\vartheta)$, $R_L(\vartheta)$, and $R_U(\vartheta)$ [11].

We now associate additional parameters $k_{i,\vartheta}$, $k_{i,l}$, $t_{i,\vartheta}$ and $t_{i,l}$, which are pertinent to the turtle's state, with the railroad ties T_i. The parameters $k_{i,\vartheta}$ and $k_{i,l}$ specify a rotation around the turtle's up direction and translation in the turtle's heading direction between consecutive points along the skeleton. Parameters $t_{i,\vartheta}$ and $t_{i,l}$ specify a rotation around the turtle's up direction and translation in the resulting heading direction, as needed to trace the railroad tie T_i as if it was a small branch off the skeleton (Figure 5).

We sort the leaf's railroad ties T_i by their first appearance along the skeleton. The turtle parameters for consecutive railroad ties T_i, $i > 0$ are then determined as follows. First, the parameters $k_{i,l}$ and $k_{i,\vartheta}$ are calculated using Equation 2.

$$\begin{aligned} \vec{K} &= \begin{pmatrix} k_{i,x} - k_{i-1,x} \\ k_{i,y} - k_{i-1,y} \\ 0 \end{pmatrix}, \\ k_{i,l} &= |\vec{K}|, \\ k_{i,\vartheta} &= sign(\hat{U} \cdot (\vec{K} \times \hat{H}))\ \ acos(\hat{H} \cdot \tfrac{\vec{K}}{|\vec{K}|}). \end{aligned} \tag{2}$$

The turtle is then translated by $k_{i,l}$ in the heading direction and rotated by angle $k_{i,\vartheta}$ around the $\hat{U}$ axis. The parameters $t_{i,l}$ and $t_{i,\vartheta}$ defining the railroad tie T_i are calculated in a similar manner:

$$\begin{aligned} \vec{T} &= \begin{pmatrix} s_{i,x} - k_{i,x} \\ s_{i,y} - k_{i,y} \\ 0 \end{pmatrix}, \\ t_{i,l} &= |\vec{T}|, \\ t_{i,\vartheta} &= sign(\hat{U} \cdot (\vec{K} \times \vec{H}))\ \ acos(\hat{H} \cdot \tfrac{\vec{T}}{|\vec{T}|}). \end{aligned} \tag{3}$$

These calculations are performed repeatedly for all railroad ties. The turtle parameters specify the leaf surface in terms of rotations around the turtle's up direction $\hat{U}$ and movements in the turtle's heading direction $\hat{H}$. The ties lie in the plane formed by $\hat{H}$ and $\hat{L}$.

The planar leaf model obtained in the previous steps can be deformed in 3D using functions that characterize additional rotations of the turtle around the axes $\hat{H}$, $\hat{L}$, and $\hat{U}$. These functions can be specified for every branch of the skeleton, controlling its turning, bending, and twisting [12]. We define them graphically, and manipulate them using an *interactive function editor* (Figure 6).

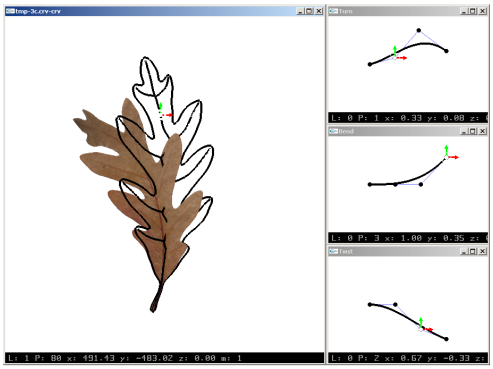

Figure 6. A screen shot of the modeling environment. The leaf model can be deformed in 3D space using graphically defined functions that control additional turning, bending, and twisting of the skeleton.

5 Examples

To illustrate our approach, we recreated surfaces of several lobed leaves. In particular, we modeled, texture mapped, and deformed leaves of *Quercus alba* (white oak), *Passiflora caerulea* (white passion flower), *Hedera helix* (needlepoint ivy), and *Bifurcatum* (staghorn) (Figure 7). Both the main axes and individual lobes were interactively deformed in 3D. In Figure 8, the scanned and deformed leaves were composed into a scene.

The silhouette and skeleton are specified as planar curves. In addition, the third dimension (z component) can be used to model crinkly patterns. A user may specify these z deformations explicitly. Alternatively, they may be generated using some noise function. Figure 9 shows oak models with 1D Perlin noise [10] applied to the silhouette curves in order to create wavy margins.

6 Conclusions

We have extended the concept of sweeps to branching structures. Our method generates a free-form deformable model from an arbitrary silhouette curve. In our experience, this method is intuitive and well suited to the interactive modeling of plants organs such as lobed leaves. Visually important aspects of appearance, such as posture, can easily be captured and controlled. The process is sufficiently fast to support interactive modeling on current personal computers.

We demonstrated the flexibility of our approach by deforming surfaces of several lobed leaves. This flexibility results from the use of turtle geometry to specify leaf geometry. The shape of leaves can be directly and intuitively manipulated using interactive modeling techniques. Images of close-up scenes and scenes containing numerous different leaves can easily be created.

An open research problem is the use of closed cross-

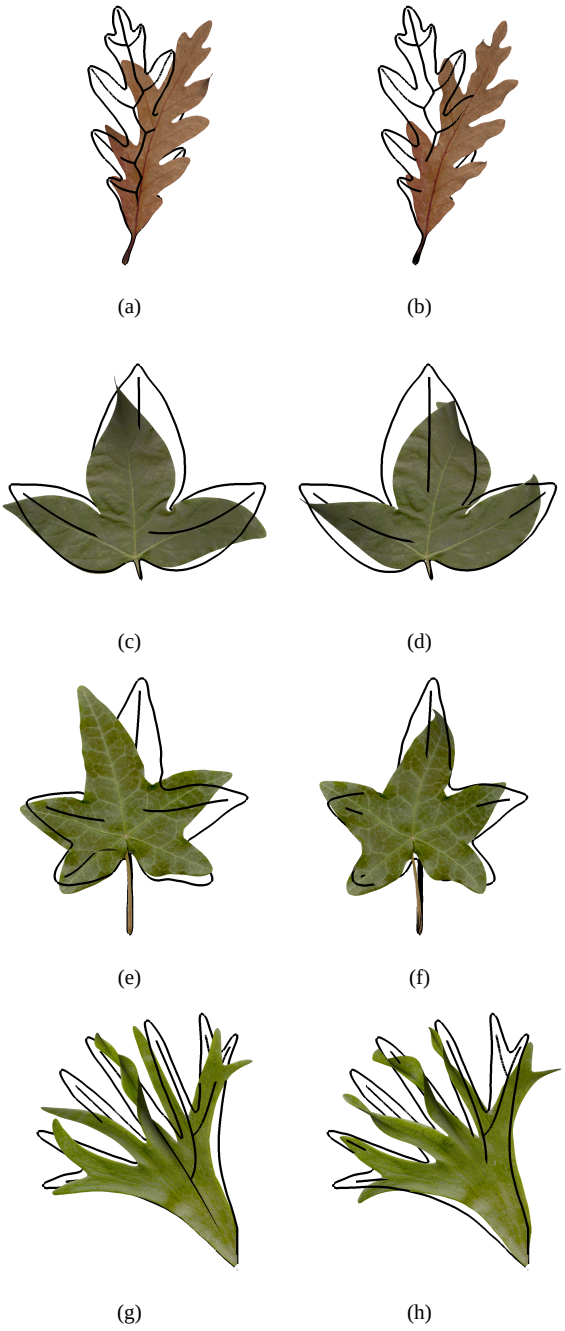

Figure 7. Deformed models of scanned leaf images: (a,b) white oak. (c,d) white passion flower. (e,f) needlepoint ivy. (g,h) staghorn.

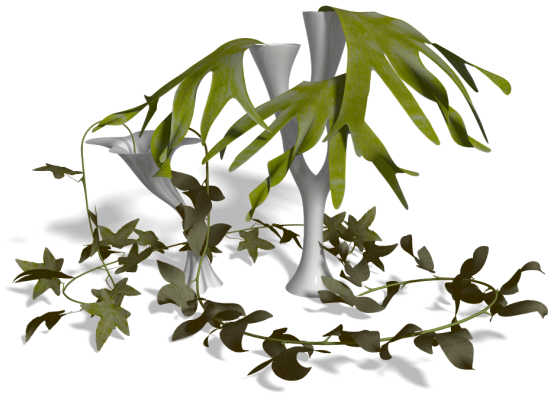

Figure 8. Rendering of a close-up scene of leaf models individually deformed from scanned images.

Figure 9. Close-up rendering of oak leaves. The leaf models are individually deformed from scanned images.

sections with this method. We think that this could lead to a unified design approach for many plant organs, including stems.

Acknowledgements

This research was supported in part by a scholarship from the Department of Foreign Affairs and International Trade (Canada) to L. Mündermann, and a Natural Sciences and Engineering Research Council of Canada Discovery Grant to P. Prusinkiewicz. The authors would like to thank Faramarz Samavati for his assistance with inverse subdivision, David Kroeker for making available to us leaves from the Devonian Gardens (City of Calgary), and Mark Fox and members of the GraphicsJungle for helpful discussions.

References

[1] H. Abelson and A. diSessa. *Turtle Geometry*. MIT Press, Cambridge, 1981.

[2] F. Aurenhammer. Voronoi diagrams - a survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23(3):345–405, 1991.

[3] T. Binford. Visual perception by computer. In *Proceedings of the IEEE System Science and Cybernetic Conference*, 1971.

[4] J. Bloomenthal. Modeling the mighty maple. In *Proceedings of SIGGRAPH 1985*, Computer Graphics, pages 305–311, 1985.

[5] H. Blum. A transformation for extracting new descriptors of shape. *Models for the Perception of Speech and Visual Form*, pages 362–380, 1967.

[6] S. Fortune. A sweepline algorithm for voronoi diagrams. *Algorithmica*, 2:153–174, 1987.

[7] M. S. Hammel, P. Prusinkiewicz, and B. Wyvill. Modelling compound leaves using implicit contours. *Visual Computing*, pages 119–131, 1992.

[8] B. Lintermann and O. Deussen. Interactive modeling of plants. *IEEE Computer Graphics and Applications*, 19(1):56–65, 1999.

[9] J. R. Parker. *Practical Computer Vision Using C*. John Wiley & Sons, Inc., 1994.

[10] K. Perlin. An image synthesizer. In *Proceedings of SIGGRAPH 1985*, Computer Graphics, pages 287–293, 1985.

[11] P. Prusinkiewicz and A. Lindenmayer. *The Algorithmic Beauty of Plants*. Springer-Verlag, New York, 1990.

[12] P. Prusinkiewicz, L. Mündermann, R. Karwowski, and B. Lane. The use of positonal information in the modeling of plants. In *SIGGRAPH 2001 Conference Proceedings*, Computer Graphics, pages 289–300, 2001.

[13] F. F. Samavati and R. H. Bartels. Multiresolution curve and surface representation: Reversing subdivision rules by least-squares data fitting. *Computer Graphics Forum*, 18(2):97–119, 1999.

[14] J. M. Snyder. *Generative Modeling for Computer Graphics and CAD*. Academic Press, 1992.

[15] C. W. A. M. van Overveld. Sticky splines: Definition and manipulation of spline structures with maintained topological relations. *ACM Transactions on Graphics*, 15(1):72–98, 1996.

[16] K. West. *How to Draw Plants. The Techniques of Botanical Illustration*. Timber Press, Portland, OR, 1997.