Chapter 7: Alien Leaf Forms

If your life is a leaf
that the seasons tear off and condemn,
they will bind you with love
that is graceful and green as the stem.
(Leonard Cohen)

In chapter 4 we introduced non-manifold surfaces by considering a leaf, its blade and veins. In this chapter, we provide a definition for the non-manifold leaf that smoothly blends blade and veins together. The definition consists of a region function, discussed in section 4.2, and a skeleton, developed in this chapter.

This work is partly inspired by the following two photographs. These magnified views testify to the elegance of Nature, and offer a view of an often overlooked world. At this scale, the veins are significant structures, and should be modeled geometrically. In an exercise of artistic prerogative, we will blend the veins into the blade with a smoothness exceeding that shown below.

Figure 7.1 Petiole, Primary and Secondary Veins, and Blade (photograph)

Figure 7.2 Primary and Secondary Veins (photograph)

The above venation, although simple, reinforces our preference for implicit techniques. Were we to implement this geometry with patches, for example, we would require a network whose topological complexity would be on the order of that illustrated below.

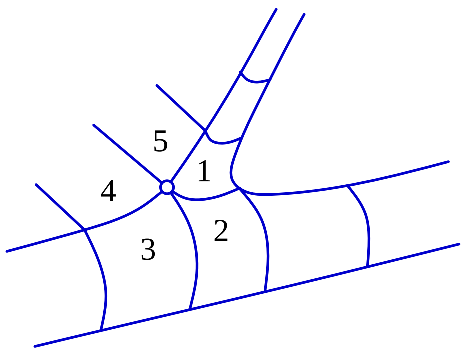


Figure 7.3 Five Patches Meet at a Point

7.1 The Leaf Skeleton

We define the leaf skeleton as those axes constituting the primary and secondary veins of the leaf. 117 points were digitized by hand, in an hour or so, from a specimen of the Northwest Maple. In the illustration below, the primary veins and petiole are drawn in green. All left descendant veins are drawn in blue, all right descendant veins are drawn in red.

The data structure for the skeleton is a tree structured graph, suggested in section 2.4. Primary veins are ordered clock-wise around the petiole; the left and right descendants of a parent vein are maintained in separate lists. The designer must distinguish descendant veins as left or right, and must order them according to distance from the base of the parent vein. This facilitates the construction of the polygonal web described in the next section. We ignored those veins wholly enclosed by the leaf blade that do not connect to the border, or *margin*, of the leaf.

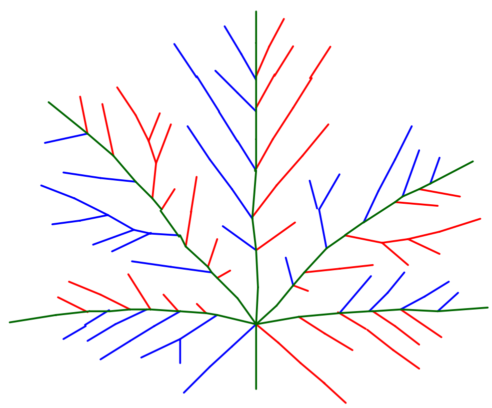


Figure 7.4 Leaf Venation

It was simpler to digitize the leaf in two, rather than three, dimensions. We associated with each point, however, a measure of curvature, defaulted to zero (flat). For a few selected points an estimated curvature was assigned; the remaining points were each assigned a curvature based on the average of the non-zero curvatures of neighboring points. The entire leaf was given a three-dimensional character by transforming each point according to its curvature. Specifically, each point was rotated so that it rested upon a sphere, centered at the base of the leaf, whose radius is the given curvature. A slightly uneven distribution of curvature produced a natural appearing, three-dimensional shape.

7.2 Blade Geometry

Because the cross-section of vein and blade is topologically equivalent to the volume and trimmed surface of figure 4.3, we provided a non-manifold definition similar to that of the sphere and patch, section 4.8.4. Specifically, one must determine whether a point is above, below, or to the side of the leaf blade. To accommodate this requirement, we represented the leaf blade as a triangle mesh.

We desire the veins to lie within the triangle mesh, even after application of the non-linear spherical transformation, described in the previous section. One method to ensure the vein remains embedded in the blade is to construct the triangle mesh so that a triangle side connects adjacent points of a vein.

The leaf margin is defined by traversing the venation graph, recursively visiting all left descendants in order and all right descendants in reverse order.[1] Two adjacent margin points are not only connected along the margin, but are also connected by veins; one need only trace the vein ancestors of the two points until a common ancestor (usually a primary vein) is reached. The circuit produced by joining two adjacent margin points via vein ancestors and via the margin forms a polygon. The collection of polygons produced by all adjacent pairs along the margin produces a polygonal web that fully tiles the leaf blade, as shown in the illustration below.

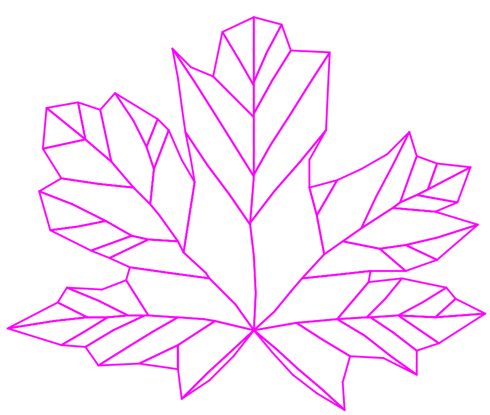


Figure 7.5 Polygonal Web

Elements of this polygonal web are, however, too large to permit a satisfactory approximation to the leaf blade. Recall that, although defined in two-dimensions, the web points are given depth by the spherical mapping. This mapping would produce noticeable creases in the larger polygons of the web. Therefore, we construct a finer polygonal mesh from the web, as illustrated below. This is performed by recursively subdividing each element of the web until the resulting polygons each have a) an area below some specified level and, b) a perimeter-to-area ratio below another specified level.

The subdivision is performed parallel to the 'narrow' axis of the polygon, and care is necessary. Because a polygon may be concave, a segment that subdivides it is not necessarily fully contained by that polygon. In the present implementation, each candidate subdividing segment is tested for inclusion within the polygon. If inclusion is not assured, the axis of subdivision is rotated until a successful segment is found.

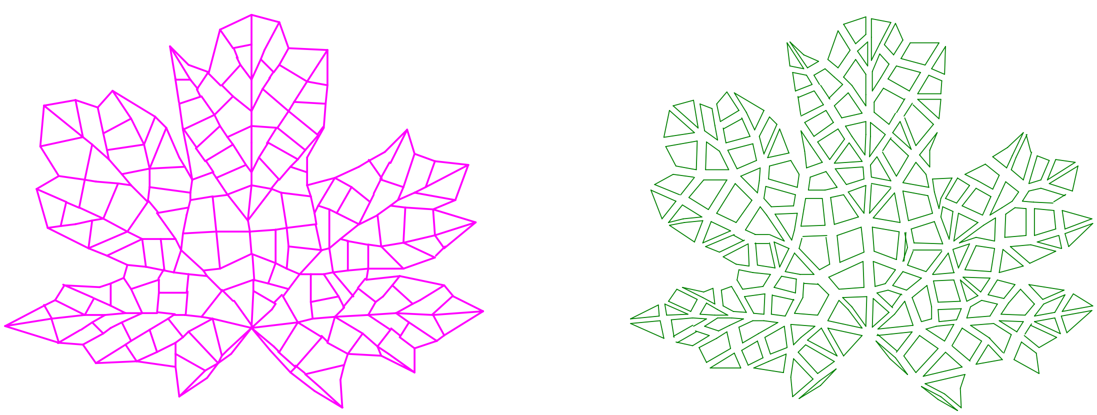


Figure 7.6 Polygonal Mesh

left: polygonal mesh, right: exploded view

Care must be taken when triangulating the polygonal mesh. A convex polygon can be safely triangulated by repeatedly removing triangles formed by connecting semi-adjacent vertices (*i.e.*, vertices separated by an intervening vertex). As illustrated below, this begins with the shortest semi-adjacent vertex connection.

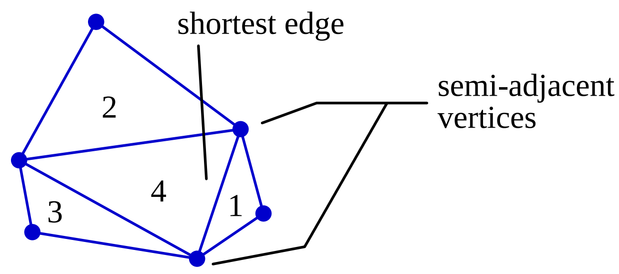


Figure 7.7 Convex Polygon Triangulation

The same procedure may be used with non-convex polygons provided that each polygon vertex be tested for inclusion within a candidate triangle; if any vertices are so included, the candidate triangle must be rejected. The figure below shows the original polygonal mesh (in green) with the triangulating edges (in red).

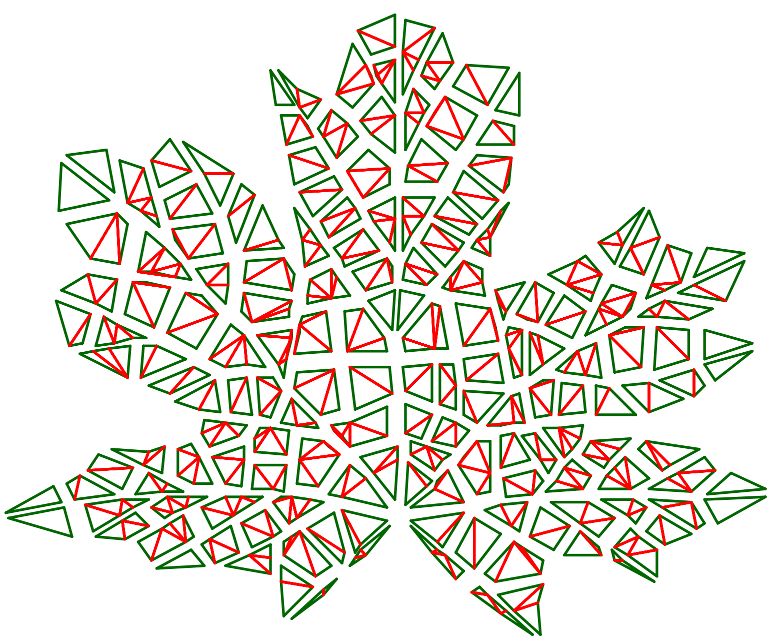


Figure 7.8 Triangulated Blade

The number of triangles in the above blade is 475, which is sufficiently large to justify an octree partitioning.[2] For each triangle tested, we must determine whether a point ***p*** is above, below, or beyond (*i.e.*, to the outside). This is done by projecting ***p*** onto the plane of the triangle. If the projection falls within the triangle, the signed distance to the plane is returned; a negative value signifying below and a positive value signifying above the plane. If the projection falls outside of the triangle, the distance to the nearest triangle edge is returned. The triangle with the minimum (unsigned) distance is used. If the projection falls outside this triangle and the nearest edge is a margin edge, ***p*** is considered beyond the triangle mesh; otherwise the sign of the distance determines whether ***p*** is above or below the triangle plane.

7.3 Of Blade and Vein

The surface of the veins in isolation are simply convolution surfaces with the skeleton defined by line segments. We employ an octree similar to that used for the triangles in order to cull segments that have no influence on a particular point in space.[3] Summation of the implicit primitives for primary veins produces the following surface.

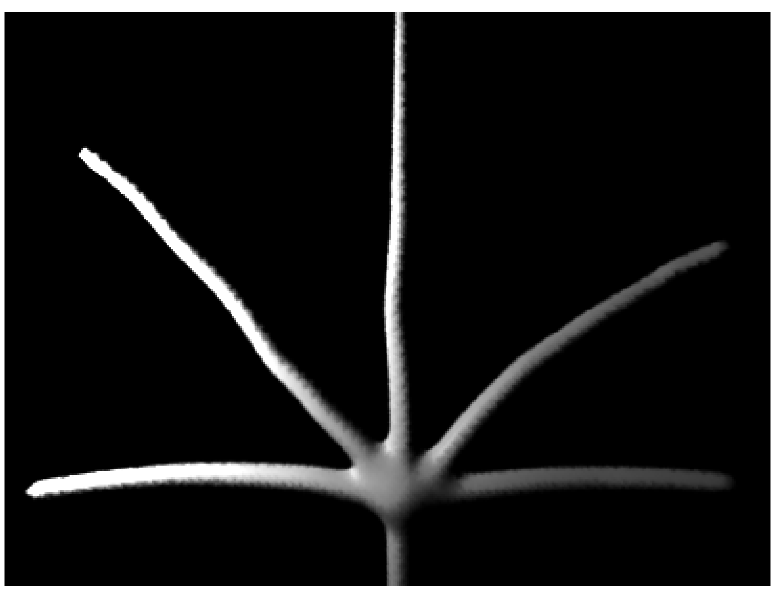


Figure 7.9 Primary Veins in Isolation

Examination of leaf veins reveals that many blend with the leaf blade in the manner illustrated below.

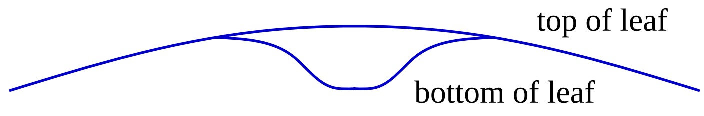
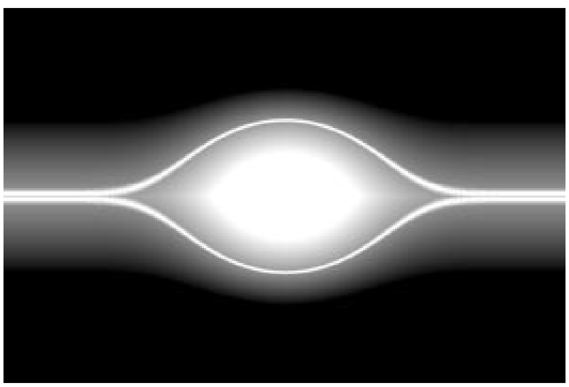


Figure 7.10 Cross-Section of Blade and a Vein

We implemented a blend in which a point is considered inside the vein if the sum of the implicit vein primitives is greater than the distance to the blade divided by the radius of the nearest vein. This blend, applied to a simple 'tee' junction embedded in a square, produces the surface shown below.



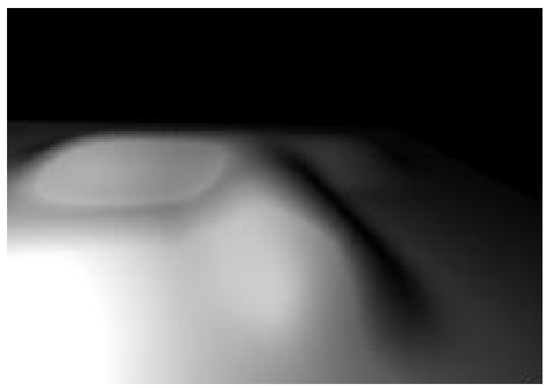


Figure 7.11 Tee Blended into Blade

left: cross-section of vein and blade, right: surface

A more localized blend improves the definition of the 'tee' shape while allowing for its blend into the square, as shown below. To mimic the cross-section of figure 7.10, a point would be considered outside the vein if it were above the blade.

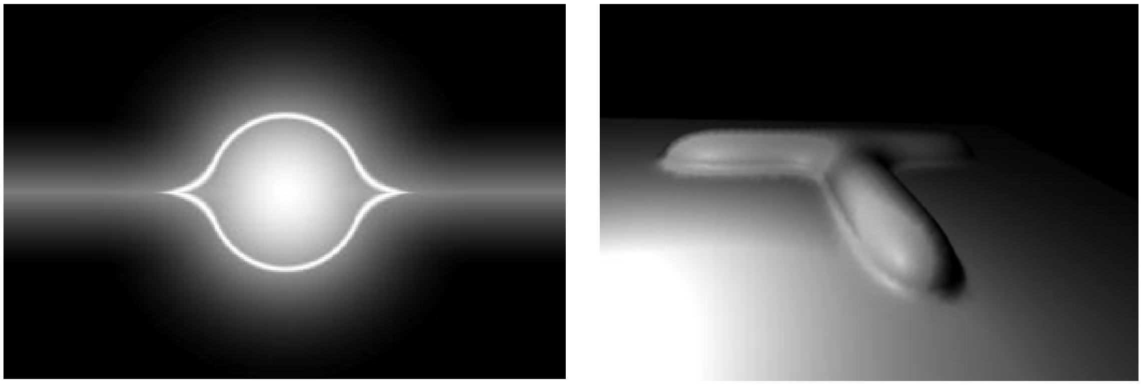


Figure 7.12 Tee Filleted into Blade

By applying this blend to the set of segments that constitute the primary leaf veins and the set of triangles that constitute the leaf blade, we obtain the surface shown below. As with the hand of chapter 6, this object is not bump or displacement mapped; the relief shown by the veins is surface geometry. Some tapering has been applied to the veins in the surface below. The veins are, however, too large overall, which detracts from the realism of the object.

Figure 7.13 Alien Leaf Form

7.4 Texture

In addition to the undue prominence of the veins, the previous image lacks realism in other ways. First, only primary veins were used to define the vein volume, and second, no texture was applied to the surface. We consider the following method to create surface texture. If, when converting the polygonal web to the polygonal mesh, the subdividing lines are themselves recursively subdivided with slight perturbations applied to their midpoints, the following pattern emerges. While this is limited in its fidelity to micro-vein patterns in a leaf, it holds some promise for improved leaf likenesses.

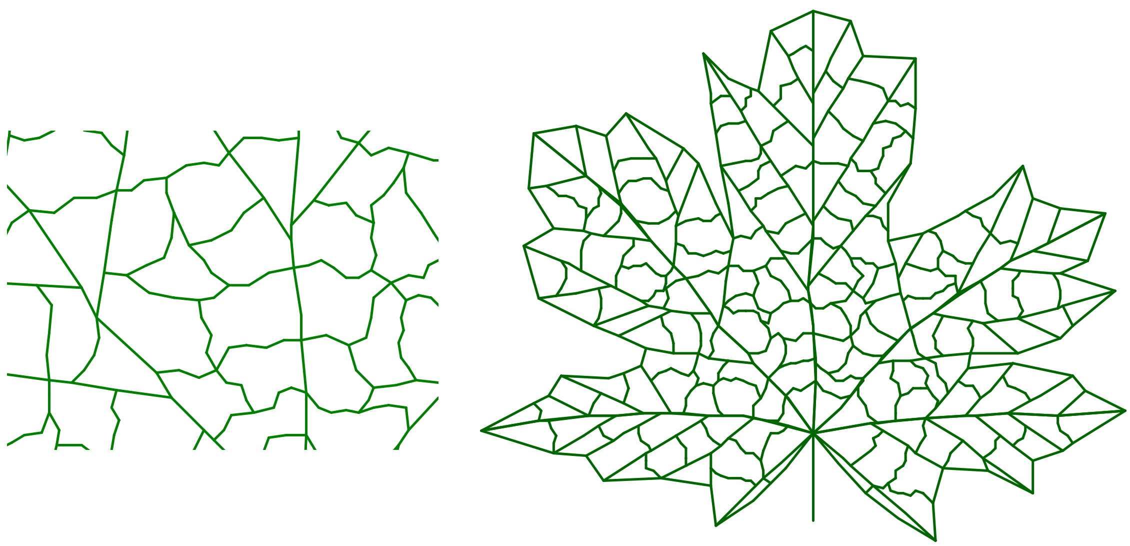


Figure 7.14 Tertiary Venation

left: closeup, right: full leaf

7.5 Conclusions

A leaf is more than a flat, common part of a plant; it is an interesting object that challenges modeling methods used in computer graphics. These challenges include the smooth blends of primitives and the representation of a non-manifold.

This chapter has applied methods of non-manifold representation and surface-volume blend, described in chapter 4, to the natural leaf form. In developing the surface from a digitized set of points, we have again demonstrated the usefulness of the skeleton in the design of natural forms.

The computational cost of distance to a patch or distance to a triangle mesh is high, however. Indeed, we may wish to minimize the use of an implicit representation of what essentially is a parametric surface. Not only can this reduce computation, but it can afford greater detail of boundary edges such as the leaf margin. Therefore, a promising area for future work is the use of a parametric surface in its ‘native’ form wherever it is fully beyond the implicitly defined volume. This may entail a non-manifold scheme similar to that described in section 4.1 in which the parametric surface is broken into triangles or sub-patches, and clipped against the implicit volume. Those triangles or sub-patches fully outside the volume can be retained in parametric form.

7.6 Notes

1. By comparing a vein tip to the preceding and following vein tips, it is possible to determine whether a margin point belongs to a concave or convex portion of the margin. In keeping with the observed characteristics of the maple leaf margin, we smooth the margin by fitting natural splines (*i.e.*, splines with zero curvature at their endpoints) to two adjacent convex margin points (along with any intervening concave margin points). As shown below, the affect of these splines is slight.

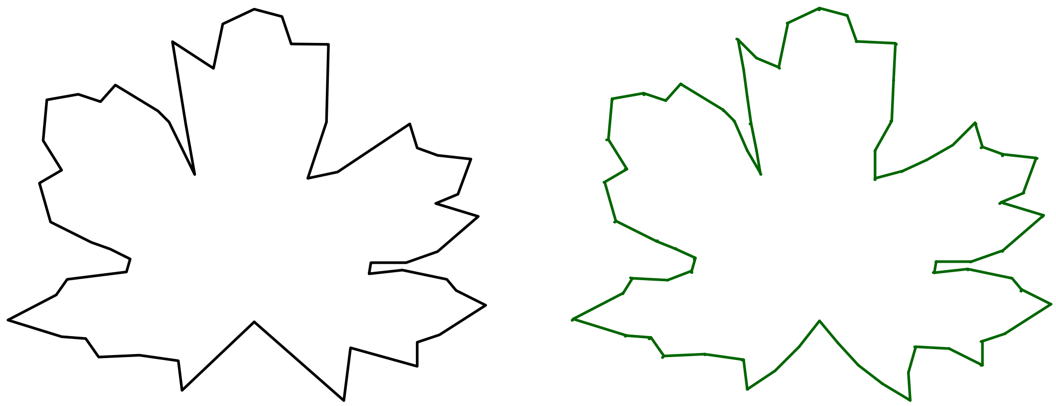


Figure 7.15 Leaf Margins

left: vein tips connected by straight lines

right: vein tips connected by natural splines

2. In the case of triangles, each octree node is associated with a cube, shown below, and a list of those mesh triangles within $s(1+\sqrt{3})/2$ of the cube center, where s is the side length of the cube. Any triangle not in the list is at least $s/2$ distant from any point within the cube or on its surface. For a given point $\boldsymbol{p}$ within the cube, if the closest triangle of those listed is less than $s/2$ distant from $\boldsymbol{p}$, then it is guaranteed to be the closest of the entire mesh. If there is no such triangle, all triangles must be tested.

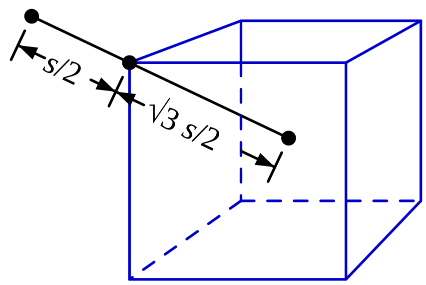


Figure 7.16 An Octree Node

3. The two octrees, one for culling veins, the other for finding close triangles, are not fully comparable. In the case of line segments, we associate with an octree

node all segments that are within $s(\sqrt{3})/2+r$ of the node center, where r is the radius of influence of the segment. Those segments not associated with the node must be at least r distant from any point within the node and, therefore, cannot influence the point.

I think I shall never see
A billboard lovely as a tree.
Indeed unless the billboards fall,
I'll never see a tree at all.
(Ogden Nash)