



ELSEVIER

Discrete Applied Mathematics 93 (1999) 33–50

DISCRETE  
APPLIED  
MATHEMATICS

## A transfinite form of Sibson's interpolant

L. Gross<sup>a</sup>, G. Farin<sup>b,\*</sup>

<sup>a</sup> IBM Corporation, 11400 Burnet Rd., Austin, TX 78758, USA

<sup>b</sup> Department of Computer Science and Engineering, Arizona State University, Tempe, AZ 85287, USA

Received 28 October 1997; accepted 22 October 1998

---

### Abstract

Sibson's interpolant uses Voronoi diagrams in the plane to interpolate a set of scattered data points. This paper presents an extension of this method to handle the interpolation of a set of functional curves (transfinite surface interpolation). We derive a simple formula for this new surface type which can interpolate to any number of boundary curves. In addition, a unique surface may be created from a set of discontinuous curves. Finally, we present a form of the interpolant which uses convex or concave polygonal domains. © 1999 Published by Elsevier Science B.V. All rights reserved.

---

### 1. Introduction

Transfinite surface interpolation is the process of constructing a surface between a series or network of input curves. This process can currently be performed with a number of well-established methods, resulting in, for example, Coons' or Gordon surfaces (see [2]). These methods, however, are limited to triangular- or rectangular-shaped surfaces, and are not easily applicable to solving the problem of interpolation across the interior of a circle,  $n$ -sided closed polygon, or general closed curve.

The purpose of this paper is to describe a new class of surfaces for performing general transfinite surface interpolation between functions defined over a plane. The method for generating these surfaces is a natural extension of Sibson's scattered data point interpolant [7] which interpolates a surface by computing the ratios of areas of Voronoi subtiles. By observation, this method produces surfaces which appear to have nearly minimal surface area, and may provide a reasonable approximation of minimal surfaces in certain circumstances.

Theoretically, functions defined over any combination of points, lines, and curves in the plane may be interpolated with these new surfaces. It is also possible to include holes within a surface where each hole has boundary functions to which a surface can be interpolated.

---

\* Corresponding author.

The remainder of this paper is organized as follows. Section 2 discusses the foundation for this method – Sibson’s scattered data point interpolant, and its underlying geometric construction – Voronoi diagrams. Section 3 presents Sibson’s interpolant in its new transfinite form, and how it is derived from the discrete form using the continuous Voronoi diagram of a circle as a domain. Section 4 extends the interpolant to polygonal domains, both convex and concave, along with examples of surfaces of this type. In addition, an example of a polygonal domain with a hole and a corresponding interpolated surface is presented. Finally, Section 5 contains conclusions and future work for this new interpolation scheme.

## 2. Sibson’s interpolant

Sibson’s interpolant is based on the Voronoi diagram of discrete (point) data. The new transfinite form of the interpolant (discussed in Section 3) is based on the Voronoi diagram of continuous (line and curve) data. For the rest of this paper, we will use the term “discrete Voronoi diagram” to refer to the Voronoi diagram of discrete data, and “continuous Voronoi diagram” for that of continuous data.

We shall briefly describe how to construct discrete Voronoi diagrams, and provide a definition for Sibson’s interpolant.

### 2.1. Discrete Voronoi diagrams

Given a set of scattered data points in the plane, a unique tessellation of the plane into tiles can be computed where each tile contains exactly one data point, and every tile edge is the perpendicular bisector between the points contained in the tiles which share that edge. This tessellation, the Voronoi diagram, divides the plane according to the *nearest-neighbor rule*: Each point is associated with the region of the plane closest to it (see pp. 345–346 of [1]).

An example of a discrete Voronoi diagram is shown in Fig. 1. The tessellation will always consist of convex polygonal tiles because the process entails the intersection of half-planes – a process which is guaranteed to produce convex regions.

When a new point is added to the data set, we can form a tile around it by modifying the boundaries of its neighbors as in Fig. 2. The new point will have a finite tile area only if it is inserted strictly *inside* the convex hull of the existing data sites. Note that a point inserted on or outside the existing convex hull will have a tile of infinite area.

### 2.2. Sibson’s interpolant

Sibson’s interpolant [7] provides a solution to the following problem. Given:

- a set of  $N$  scattered data points  $p_n$  in the plane, and
- a function value  $z_n$  at each point;

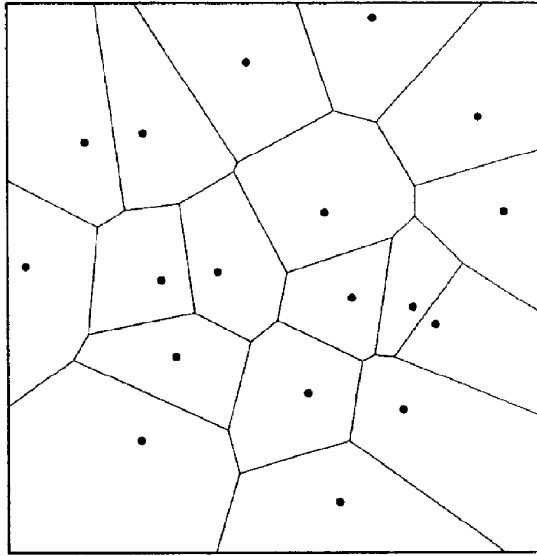


Fig. 1. Discrete Voronoi diagram.

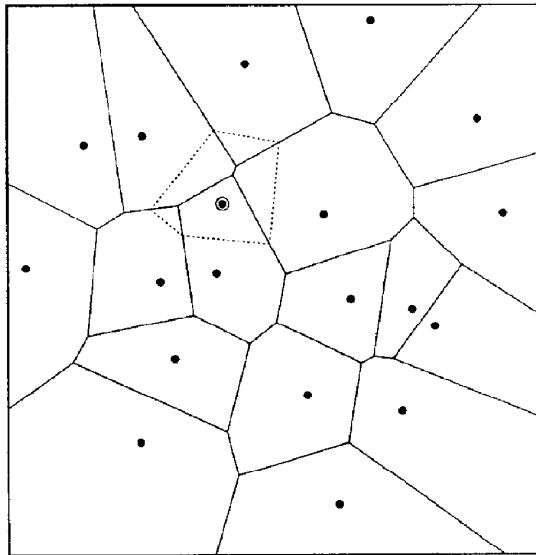


Fig. 2. Discrete Voronoi diagram with inserted point.

Find: a reasonably smooth function  $f$ , defined over the convex hull of the  $p_n$ , which is an exact interpolant, such that  $f(p_n) = z_n$  for  $n = 1, \dots, N$ , and which reflects the shape of the input data.

Sibson approached this problem by computing the discrete Voronoi diagram of the data points, and then using an identity called the *local coordinates property* to construct a new interpolant. The local coordinates property states that a point which is inserted

within the convex hull of the data points can be expressed as a convex combination of its neighboring points' positions and "contributing" tile areas [7]. Specifically, a point  $\mathbf{p}$  will have  $m$  neighbors  $\mathbf{p}_1, \dots, \mathbf{p}_m$  after being inserted into the set of data points. Each neighbor would contribute a portion of its own tile area to create  $\mathbf{p}$ 's tile. Each of these areas would then be a fraction  $u_i$  of  $\mathbf{p}$ 's total tile area, where

$$\sum_{i=1}^m u_i = 1.$$

We can call  $\mathbf{u} = (u_1, \dots, u_m)$  the *local* or *Sibson coordinates* of  $\mathbf{p}$ , where

$$\mathbf{p} = \sum_{i=1}^m u_i \mathbf{p}_i$$

is the local coordinates property [3,7].

Fig. 2 shows an example of a point being inserted into a Voronoi diagram. The inserted point there has five neighbors ( $m = 5$ ) which each donate a part of their own tile area to form the new point's tile. If we associate a function value  $z_n$  with each data site  $\mathbf{p}_n$ , we can interpolate a value at any point  $\mathbf{p}$  within the convex hull of the  $\mathbf{p}_n$ . This can be accomplished by determining  $\mathbf{p}$ 's Sibson coordinates  $\mathbf{u}$ , the function values of  $\mathbf{p}$ 's neighbors, and using *Sibson's interpolant*

$$s(\mathbf{u}) = \sum_{i=1}^m u_i z_i.$$

### 3. Transfinite form of Sibson's interpolant

We now present a new form of Sibson's interpolant which can interpolate surfaces across one or more boundary curves. This *transfinite* form of the interpolant relies on Voronoi diagrams of continuous data, e.g. lines and curves rather than points.

#### 3.1. Continuous Voronoi diagrams

The Voronoi diagram of a circle is perhaps the simplest example of a continuous Voronoi diagram. It consists of a single (infinite area) tile with a point at the circle's centroid. To understand why this occurs, we can approximate the circle with a set of points and generate their discrete Voronoi diagram as shown in Fig. 3. No matter how many points we insert into the diagram along the circumference of the circle, the only structure that is formed is the centroid point. Inserting a point inside the circle produces an elliptical tile with foci at the circle's center and the inserted point as in Fig. 4.

We will address the continuous Voronoi diagrams of polygons in Section 4.

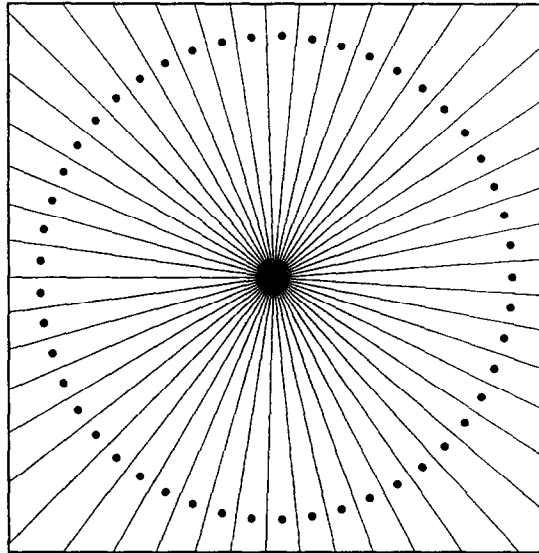
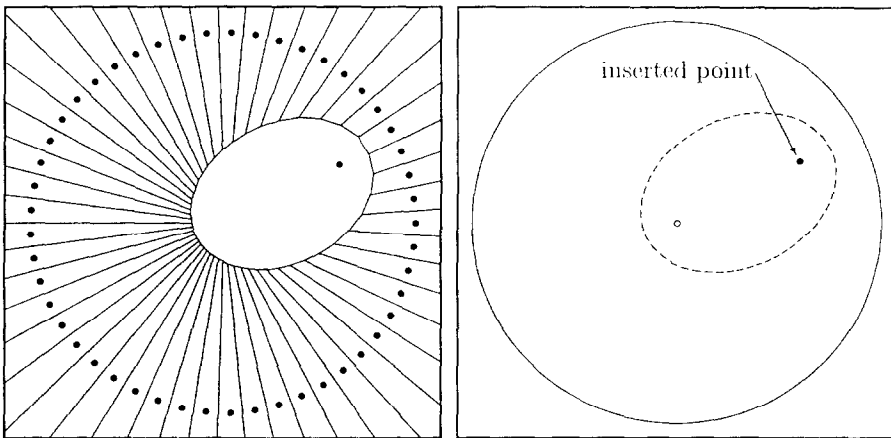


Fig. 3. Discrete Voronoi diagram of a circle.



(a) Discrete

(b) Continuous

Fig. 4. Voronoi diagrams of a circle with an inserted point.

### 3.2. Transfinite Sibson's interpolant: circular domain

We can now extend Sibson's original interpolant so that it performs transfinite surface interpolation between functions rather than scattered data point interpolation. The transfinite Sibson's interpolant can be calculated anywhere within the convex hull of the input data as in the discrete case.

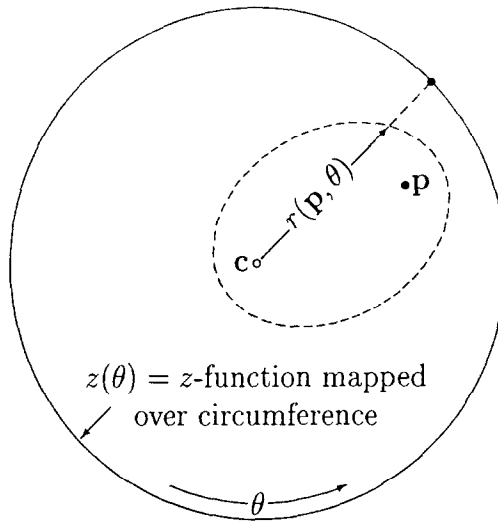


Fig. 5. Integration method over the interior of a circle.

The transfinite form of Sibson's interpolant solves the following problem. Given:

- a circle  $c(\Theta)$  in the plane, and
- a boundary function  $z(\Theta)$  over the circumference of the circle, where  $\Theta$  is a polar angle based parameter;

Find: a reasonably smooth function  $f$ , defined over the circle's interior, which is an exact interpolant, such that  $f(c(\Theta)) = z(\Theta)$  for all  $\Theta$ , and which reflects the shape of the input data.

If we refer to Fig. 4a we can see that no matter where we insert a point inside the circle, the new elliptical tile will receive contributions to its area from every existing tile. In addition, each tile area contribution will be sector-shaped.

Sibson's interpolant requires us to compute the sum of the fractions of tile area contributed by a given set of neighboring tiles multiplied by their function values. When a point  $p = (\rho, \theta)$  is inserted inside the circle as in Fig. 5, we can define the distance from the circle's centroid to the edge of the new elliptical tile as a function  $r(p, \Theta)$ . Each sector area contributed to the new tile is now  $(r^2(p, \Theta)/2) d\Theta$ , and the transfinite form of Sibson's interpolant becomes

$$s(p) = \frac{\int_0^{2\pi} (r^2(p, \Theta)/2) z(\Theta) d\Theta}{\int_0^{2\pi} (r^2(p, \Theta)/2) d\Theta}. \quad (1)$$

We derive function  $r$  as

$$r(p, \Theta) = \frac{\rho^2 - 1}{2(\rho \cos(\theta - \Theta) - 1)}.$$

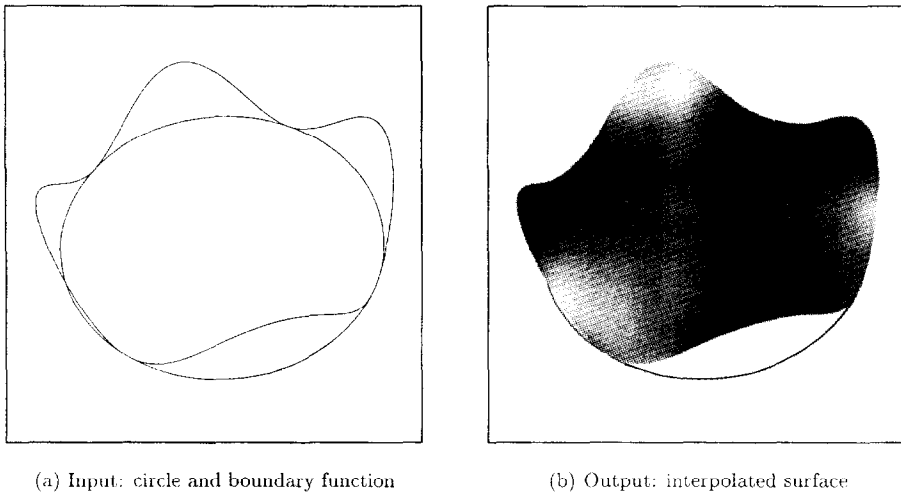


Fig. 6. Example of transfinite Sibson's interpolant over a circle.

The denominator of Eq. (1) is simply the area of the elliptical tile which is  $(\pi\sqrt{1-\rho^2})/4$ . Eq. (1) may now be rewritten as

$$s(\mathbf{p}) = s(\rho, \theta) = \frac{(1-\rho^2)^{3/2}}{2\pi} \int_0^{2\pi} \frac{z(\Theta)}{(\rho \cos(\theta - \Theta) - 1)^2} d\Theta \quad \begin{cases} 0 \leq \rho < 1, \\ 0 \leq \theta \leq 2\pi. \end{cases} \quad (2)$$

Fig. 6 shows an example of a surface created with the above equation where  $z(\Theta) = \sin(4\Theta)$ .

#### 4. Polygonal domain case

In this section we shall discuss the implementation of the transfinite form of Sibson's interpolant using a polygonal rather than a circular domain. It is possible, of course, to create a mapping from a circular domain to a set of boundary functions which form a polygon. However, the use of a polygonal domain for this interpolant will produce a noticeably different surface than will a circular domain for the same set of boundary functions (see [4]).

##### 4.1. Voronoi diagrams of polygons

A continuous Voronoi diagram in the plane can be computed from any set of line and/or curve segments in that plane. To simplify the discussion of geometric properties, however, we will consider two specific configurations of line segments: *convex and concave polygons*. The tessellation of curves is, in general, a very complex undertaking.

##### 4.1.1. Convex polygons

In an analogous manner to Section 3.1, let us consider the simple example of the continuous Voronoi diagram of an equilateral triangle as shown in Fig. 8. This figure

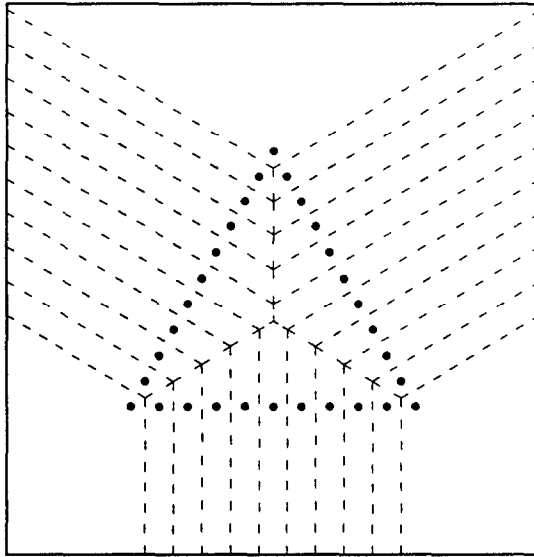


Fig. 7. Discrete Voronoi diagram of an equilateral triangle.

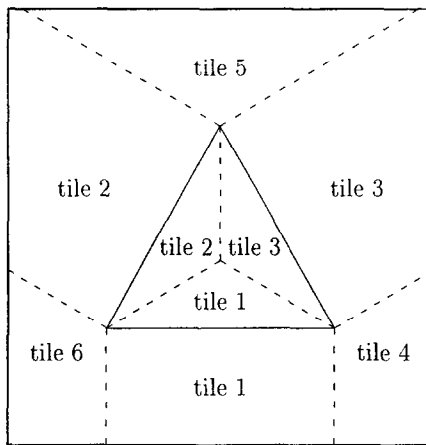


Fig. 8. Continuous Voronoi diagram of an equilateral triangle.

can be seen as the extension of its discrete counterpart in Fig. 7 where we see angle bisectors formed inside the triangle, and tile edges created on the triangle's exterior which are perpendicular to its legs and near its vertices. The continuous Voronoi diagram creates a tessellation of the triangle such that each vertex and edge lies in a separate tile. In Fig. 8, the vertices are in tiles 4, 5, and 6 while the edges are in tiles 1, 2, and 3.

For the simple case of the equilateral triangle we merely need to compute the angle bisectors to produce its continuous Voronoi diagram. However, for polygons lacking



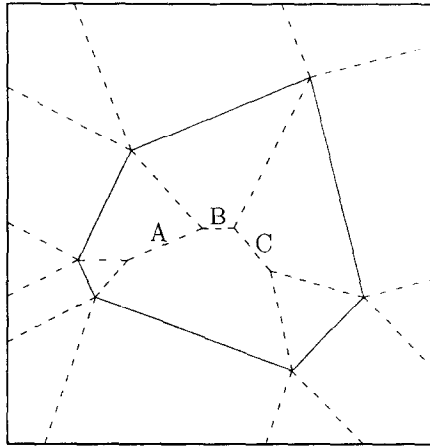


Fig. 9. Continuous Voronoi diagram of a general convex polygon.

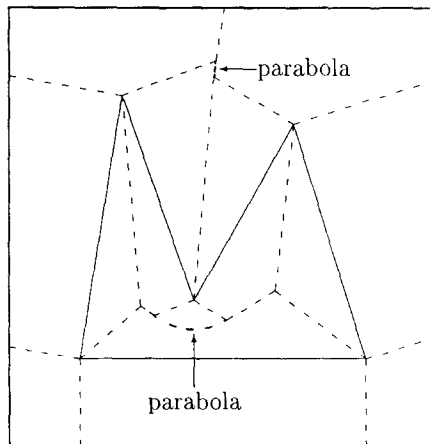


Fig. 10. Continuous Voronoi diagram of a simple concave polygon.

symmetry as in Fig. 9, the Voronoi diagram includes tile edges labeled *A*, *B*, and *C* which are bisectors between non-intersecting line segments. Computing such diagrams is not a trivial process, and algorithms of various complexities and efficiencies have been written to solve the problem [5,6,8].

#### 4.1.2. Concave polygons

An example of a continuous Voronoi diagram for a concave polygon is shown in Fig. 10. Here we can see that in addition to every polygon leg, every vertex which creates a concavity, i.e. reflex vertex, also has its own tile. Parabolic arcs will appear in the Voronoi diagrams of concave polygons at the locations where a reflex vertex and a polygon leg approach one another. The tile edges within the Voronoi diagram

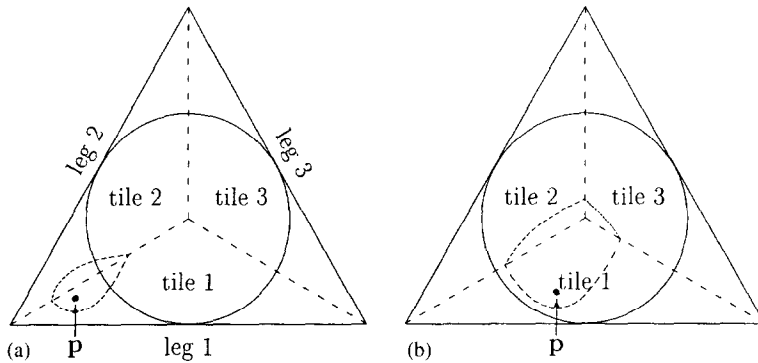


Fig. 11. Inserting a point into the Voronoi diagram of an equilateral triangle.

are defined as bisectors between data elements, and therefore will become parabolic arcs when bisecting a point and a line segment.

#### 4.1.3. Inserting a point into the Voronoi diagram of a polygon

When a point is inserted into the Voronoi diagram of a convex polygon, its tile edges will all be parabolic arcs. The inserted point's tile boundary will always be convex because the focus of each parabolic arc is the point.

The number of arcs which compose the inserted point's tile boundary is directly related to the number of *maximal circles* in which the point lies. The maximal circles (MCs) for a convex polygon are defined as circles interior to the polygon which are centred at tile vertices of the Voronoi diagram and are tangent to at least three polygon legs.

Fig. 11 shows the two general outcomes when a point  $p$  is inserted into the Voronoi diagram of the interior of an equilateral triangle. In Fig. 11a,  $p$  is inserted outside of the triangle's maximal circle, and its new tile receives contributions to its area from tiles 1 and 2 only. When  $p$  falls inside the MC, its new tile will receive contributions from all three tiles. These two cases must be distinguished when calculating the interpolant so that we know how many area integrals are to be evaluated. For more detailed information about calculating the number of MCs for a general convex polygon, see [4].

The situation is similar for concave polygons. The only difference from the convex polygon case occurs when the point  $p$  is inserted near a reflex vertex of the polygon. In this region,  $p$ 's tile boundary will contain a linear edge which will define the bisector of  $p$  and the reflex vertex (see Fig. 12). Tiles surrounding an inserted point in this case are guaranteed to be convex for the same reason as was stated for the convex polygon case.

The MCs for a concave polygon are similar to those for a convex polygon. They are centred at tile vertices of the Voronoi diagram and as a result may be tangent to only two polygon legs and pass through a reflex vertex. Fig. 13 shows the MCs

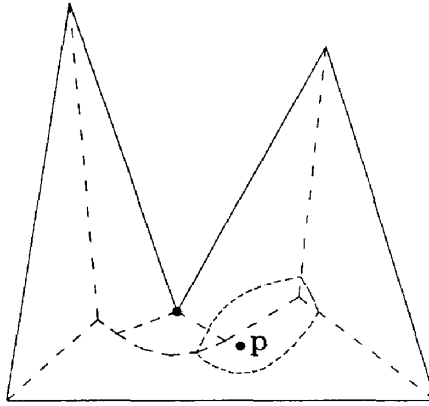


Fig. 12. Point  $p$ 's tile is formed by parabolic arcs and a linear edge.

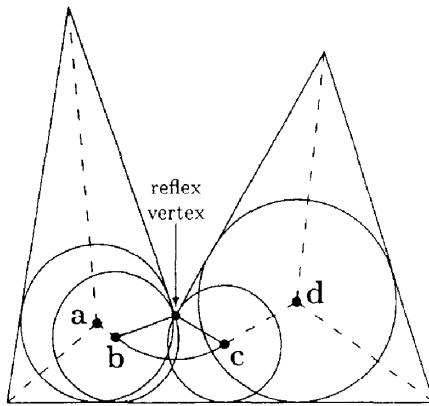


Fig. 13. Maximal circles of a concave polygon.

corresponding to the polygon of Fig. 12. The MCs centred at branch points  $a$  and  $d$  are each tangent to three polygons legs, while the MCs centred at  $b$  and  $c$  are each tangent to two legs and pass through the reflex vertex.

#### 4.2. Transfinite Sibon's interpolant: polygonal domain

The problem we will now solve can be stated as follows. Given:

- a closed  $N$ -sided polygon in the plane, and
- a boundary function  $z_n(t)$  over each polygon leg  $\mathbf{l}_n$ ;

Find: a reasonably smooth function  $f$ , defined over the polygon's interior, which is an exact interpolant, such that  $f(\mathbf{l}_n) = z_n(t)$  for all  $t$  and  $n = 1, \dots, N$ , and which reflects the shape of the input data.

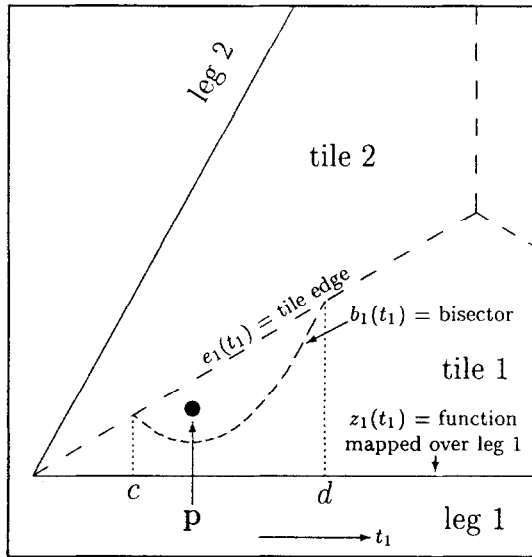


Fig. 14. Integration method for transfinite Sibson's interpolant for subtile 1.

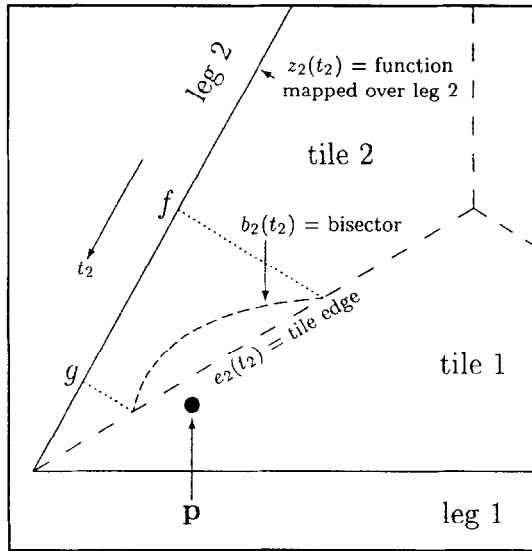


Fig. 15. Integration method for transfinite Sibson's interpolant for subtile 2.

4.2.1. Convex polygonal domain

To be able to evaluate  $f$ , we must insert a point into the Voronoi diagram so that it falls inside the given polygonal domain. As an example of how the interpolant is then calculated, let us look at Figs. 14 and 15 which contain enlargements of subtiles 1 and 2, respectively, from Fig. 11a. By assigning a parameter  $t_1$  along the length of leg 1, we can define functions  $e_1(t_1)$ ,  $b_1(t_1)$ , and  $z_1(t_1)$  to facilitate the computation of

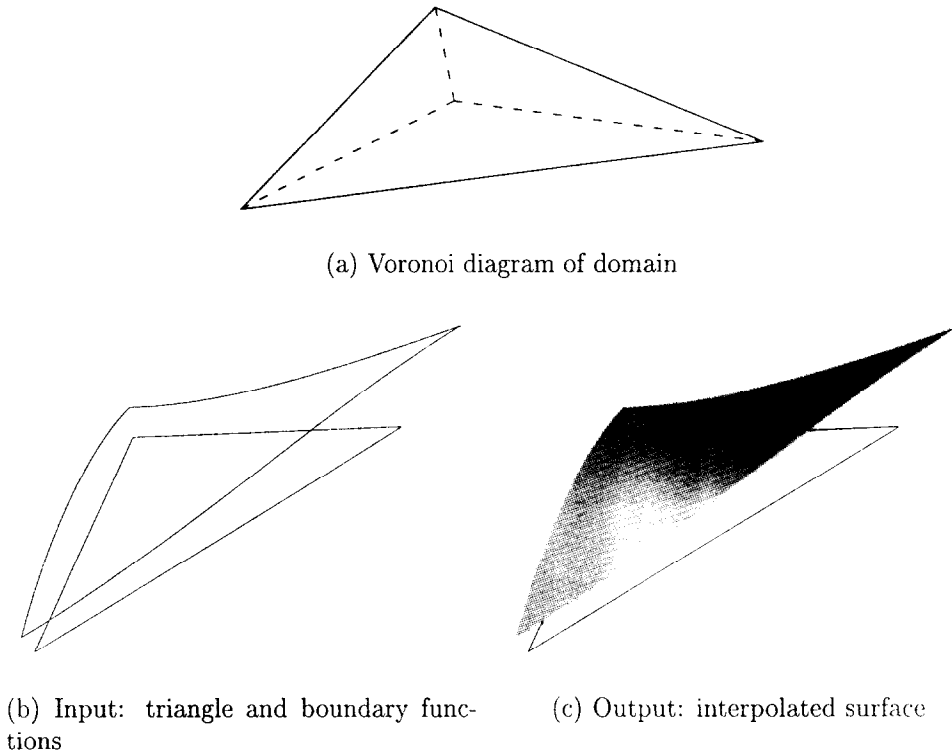


Fig. 16. Example of transfinite Sibson's interpolant over a triangle.

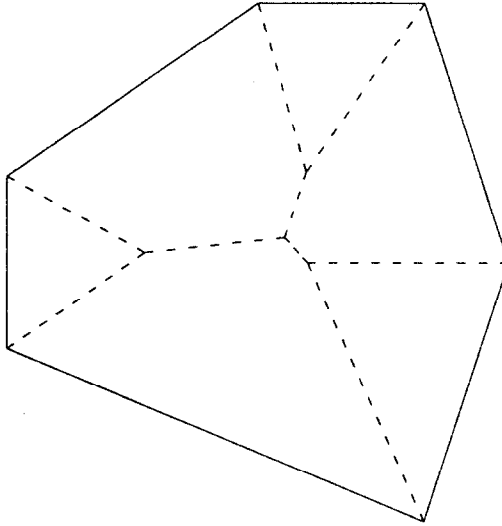
the interpolant. The function  $e_1$  corresponds to the edge of tile 1,  $b_1$  corresponds to the parabolic bisector between  $\mathbf{p}$  and leg 1, and  $z_1$  corresponds to the boundary function defined over leg 1. The limits of integration are defined by the intersection points of  $e_1$  and  $b_1$ , and are marked as  $c$  and  $d$ . The area contributed by tile 1 to create one of  $\mathbf{p}$ 's subtiles can be written as  $\int_c^d [e_1(t_1) - b_1(t_1)] dt_1$ . Similarly, as shown in Fig. 15 we can assign a parameter  $t_2$  along the length of leg 2 and determine the area contributed by tile 2 toward  $\mathbf{p}$ 's tile as  $\int_f^g [e_2(t_2) - b_2(t_2)] dt_2$ . The transfinite Sibson's interpolant is then expressed as

$$s(\mathbf{p}) = \frac{\int_c^d [e_1(t_1) - b_1(t_1)]z_1(t_1) dt_1 + \int_f^g [e_2(t_2) - b_2(t_2)]z_2(t_2) dt_2}{\int_c^d [e_1(t_1) - b_1(t_1)] dt_1 + \int_f^g [e_2(t_2) - b_2(t_2)] dt_2}.$$

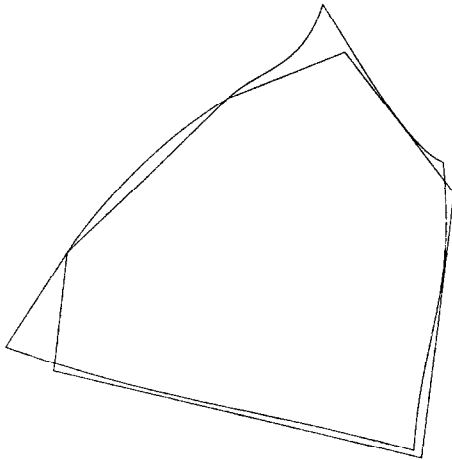
As stated earlier, when  $\mathbf{p}$  lies inside the MC, all three tiles will donate a portion of their area toward  $\mathbf{p}$ 's tile. In this case, the transfinite Sibson's interpolant is

$$s(\mathbf{p}) = \frac{\sum_{i=1}^3 \int_{t_{\text{lower}}}^{t_{\text{upper}}} [e_i(t_i) - b_i(t_i)]z_i(t_i) dt_i}{\sum_{i=1}^3 \int_{t_{\text{lower}}}^{t_{\text{upper}}} [e_i(t_i) - b_i(t_i)] dt_i}. \tag{3}$$

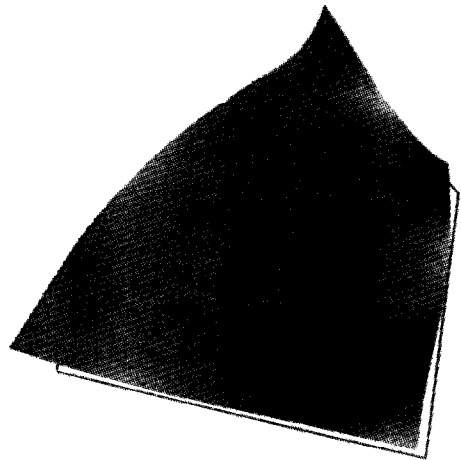
An example of the transfinite version of Sibson's interpolant for a triangular domain is shown in Fig. 16, and for a hexagonal domain in Fig. 17.



(a) Voronoi diagram of domain



(b) Input: hexagon and boundary functions

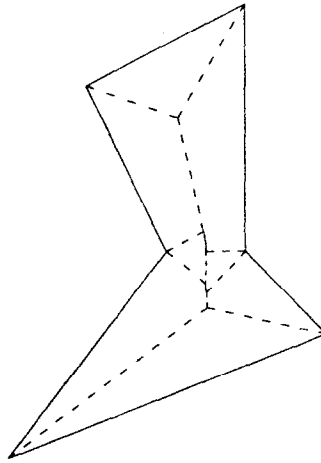


(c) Output: interpolated surface

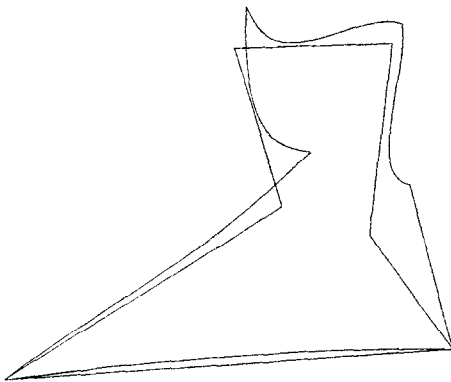
Fig. 17. Example of transfinite Sibson's interpolant over a hexagon.

#### 4.2.2. Concave polygonal domain

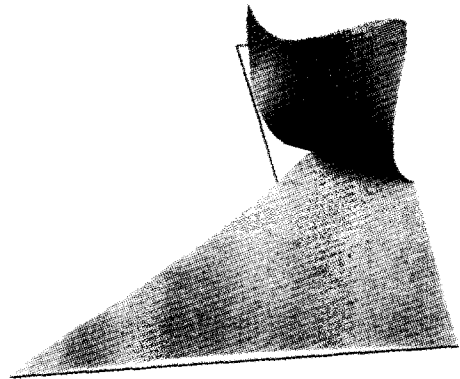
A reflex vertex's tile acts exactly like the tile of a point in the discrete case – bisectors with inserted points are straight lines. In Fig. 13, any point which is inserted within the union of the areas enclosed by the MCs centred at  $b$  and  $c$ , or within the triangular region between those circles and the bottom polygon leg, will receive an area contribution to its tile from the reflex vertex's tile.



(a) Voronoi diagram of domain



(b) Input: concave hexagon and boundary functions



(c) Output: interpolated surface

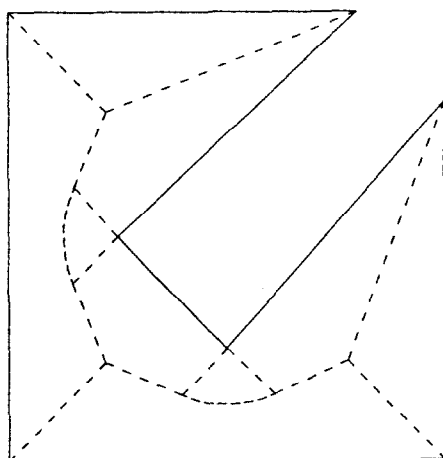
Fig. 18. Example of transfinite Sibson's interpolant over a concave hexagon.

We can now generalize Eq. (3) to represent the transfinite Sibson's interpolant for a point inserted within the Voronoi diagram of any convex or concave polygon as

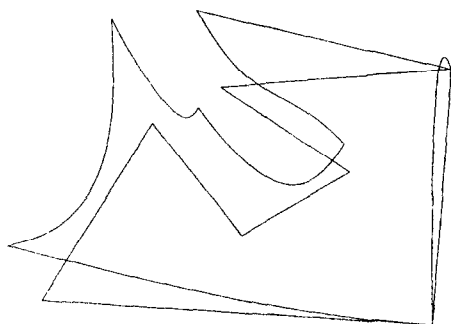
$$s(\mathbf{p}) = \frac{\sum_{i=1}^m \int_{t_{\text{lower}}}^{t_{\text{upper}}} [e_i(t_i) - b_i(t_i)] z_i(t_i) dt_i + \sum_{j=1}^n r_j z_j}{\sum_{i=1}^m \int_{t_{\text{lower}}}^{t_{\text{upper}}} [e_i(t_i) - b_i(t_i)] dt_i + \sum_{j=1}^n r_j}, \quad (4)$$

where

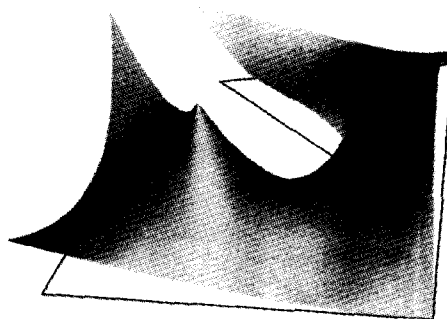
- $m$  is the number of contributing polygon leg tiles;
- $n$  is the number of contributing reflex vertex tiles;



(a) Voronoi diagram of domain



(b) Input: concave heptagon and boundary functions



(c) Output: interpolated surface

Fig. 19. Example of transfinite Sibson's interpolant over a concave heptagon.

- $r_j$  is the area of reflex vertex subtile  $j$ ; and
- $z_j$  is the function value corresponding to reflex vertex subtile  $j$ .

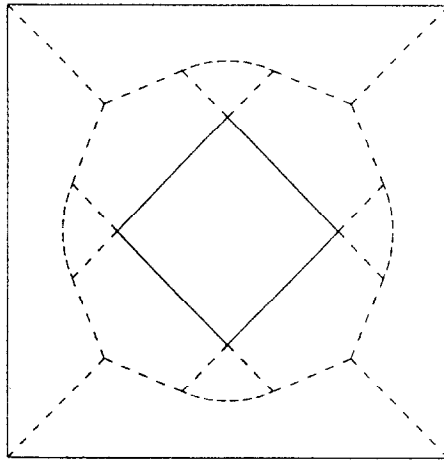
Note that the interpolant can be computed anywhere within the convex hull of the given data – a surface can be generated inside a polygon's concavity.

Examples of the interpolant for concave polygons are shown in Figs. 18 and 19.

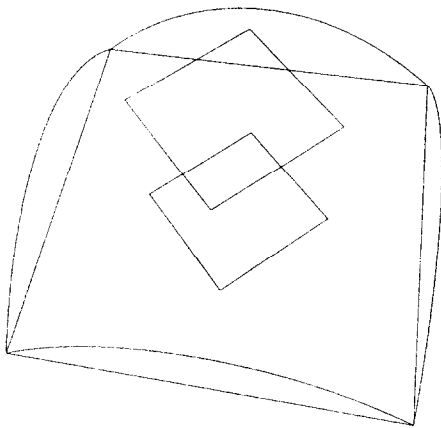
#### 4.2.3. Polygon with hole domain

The transfinite Sibson's interpolant is well defined for any two-dimensional domain polygon. In fact, the polygon need not be closed or continuous – the interpolant will be defined within the convex hull of the input domain. Domain points may be included in the interpolation process to force the surface to interpolate to specific values within the boundary functions. Holes may also be included within polygons with boundary

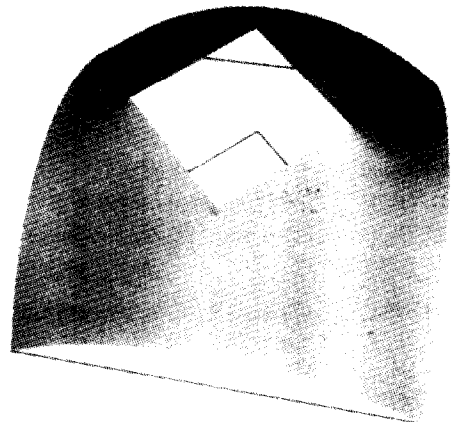




(a) Voronoi diagram of domain



(b) Input: square with hole and boundary functions



(c) Output: interpolated surface functions

Fig. 20. Example of transfinite Sibson's interpolant over a polygon with a hole.

functions of their own. See Fig. 20 for an example. Note that the Voronoi diagram for this example had to be generated manually because Lee's algorithm (see [5]) for generating Voronoi diagrams of polygons does not handle polygons with holes.

## 5. Conclusions

This paper presents a new method for performing general transfinite surface interpolation between functions defined over a plane. It is a natural extension of a discrete

surface interpolation algorithm first proposed by R. Sibson, and produces surfaces which are uniquely defined even with discontinuous input data. As seen in the figures, there are many possible geometric configurations for the boundary curves of these surfaces, as well as many possible domains for a given surface.

Additional work has been completed which provides first and second partial derivative formulas, and explores the intriguing results of the application of this interpolant to parametric data. There has also been research towards modification of the method to allow the input of tangent information along the boundary curve, and to allow interpolation between more than one set of boundary curves. This information will be available in a forthcoming paper.

Further work may concentrate on the relationship, if any, between these surfaces and minimal surfaces given their similarity in appearance for identical boundary curves. There are also possibilities for additional work toward extending the method to handle higher dimensional data (the concept should theoretically work for data of *any* dimension). Finally, it may be possible to extend Shepard's method for scattered data point interpolation to a transfinite form in a similar fashion as was done for Sibson's. However, Hardy's multiquadric and thin plate spline methods seem much more difficult to generalize this way.

## References

- [1] F. Aurenhammer, Voronoi diagrams – a survey of a fundamental geometric data structure, *ACM Comput. Surveys* 23 (3) (1991) 345–405.
- [2] G. Farin, *Curves and Surfaces for Comput. Aided Geometric Des.*, second ed., Academic Press, New York, 1990.
- [3] G. Farin, Surfaces over Dirichlet Tessellations, *Comput. Aided Geometric Des.* 7 (1–4) (1990) 281–292.
- [4] L.M. Gross, Transfinite surface interpolation over Voronoi diagrams, Ph.D. Dissertation, Arizona State University, December 1995.
- [5] D.T. Lee, Medial axis transformation of a planar shape, *IEEE Trans. Pattern Anal. Machine Intell.* PAMI-4 (4) (1982) 363–369.
- [6] D.T. Lee, R.L. Drysdale, III. Generalization of Voronoi diagrams in the plane, *SIAM J. Comput.* 10 (1) (1981) 73–87.
- [7] R. Sibson, A brief description of natural neighbour interpolation, in: V. Barnett (Ed.), *Interpolating Multivariate Data*, Wiley, New York, 1981.
- [8] C.K. Yap, An  $O(n \log n)$  algorithm for the Voronoi diagram of a set of simple curve segments, *Discrete Comput. Geom.* 2 (1987) 365–393.