Spatial Tessellations: Concepts and Applications of Voronoi Diagrams
Edition by ATSUYUKI OKABE & BARRY BOOTS and SUNG NOK CHIU
Copyright © 1992,2000 by John Wdey & Sons Ltd

CHAPTER 6

Spatial Interpolation

In Chapter 2 the ordinary Voronoi diagram was defined with respect to a generator set of n distinct points, $P = \{p_1, \ldots, p_n\}$, located in m-dimensional Euclidean space, $\mathbb{R}^m$. As with the bounded ordinary Voronoi diagram of Section 2.1, our concern here will be with a finite region S in $\mathbb{R}^m$. The location of each point p_i in S is indicated by $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{im})^{\mathrm{T}}$ and for the purposes of this chapter we label these points as *data sites*. Unlike Chapter 5 we make no assumption regarding how the data sites are located in S, although in most of the applications discussed below the data sites are irregularly spaced in S. In addition, for each data site we have a numerical observation z_i which we call the *data value*. If we assume that the data values represent observations from a surface defined by $(\boldsymbol{x}, z)$, *spatial interpolation* involves finding a function $f(\boldsymbol{x})$ which best represents the entire surface and which predicts values of z for locations other than P in S. Such a function is referred to as an *interpolant*. The particular form we select for $f(\boldsymbol{x})$ depends on the type of measurement scale of z and the use to which the interpolated surface is put. However, we can make an immediate distinction between *exact* and *approximate interpolants* in terms of whether or not they reproduce the data values; i.e. for an exact interpolant, $f(\boldsymbol{x}_i) = z_i$. Approximate interpolants may be considered a form of data smoothing which are more appropriate than exact interpolants if the data values are subject to inaccurate measurement or errors. This distinction is not of primary concern in this chapter although most of the interpolants discussed specifically are exact ones.

Various other characteristics may also be used to distinguish between interpolants. One which is relevant here is the difference between *global* and *local interpolants*. With the former the value of $f(\boldsymbol{x})$ at any given location in S depends on all the data values while the latter involves the use of only those values at 'nearby' points. Various definitions of nearby are provided below. In general, local interpolants involve partitioning S into s subdomains, S_i; $S = \cup_{i=1}^{s} S_i$ and then selecting a function of the form

$$f(\boldsymbol{x}) = f_i(\boldsymbol{x}), \quad \boldsymbol{x} \in S_i, \quad i = 1, \ldots, s. \tag{6.0.1}$$

Thus, global interpolants are affected by the addition or deletion of data values or by changing the location of data sites. For local interpolants such

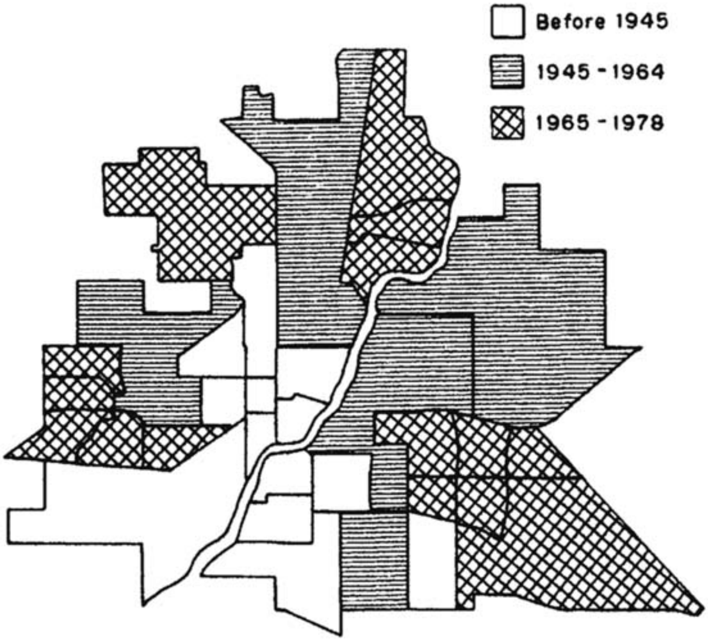


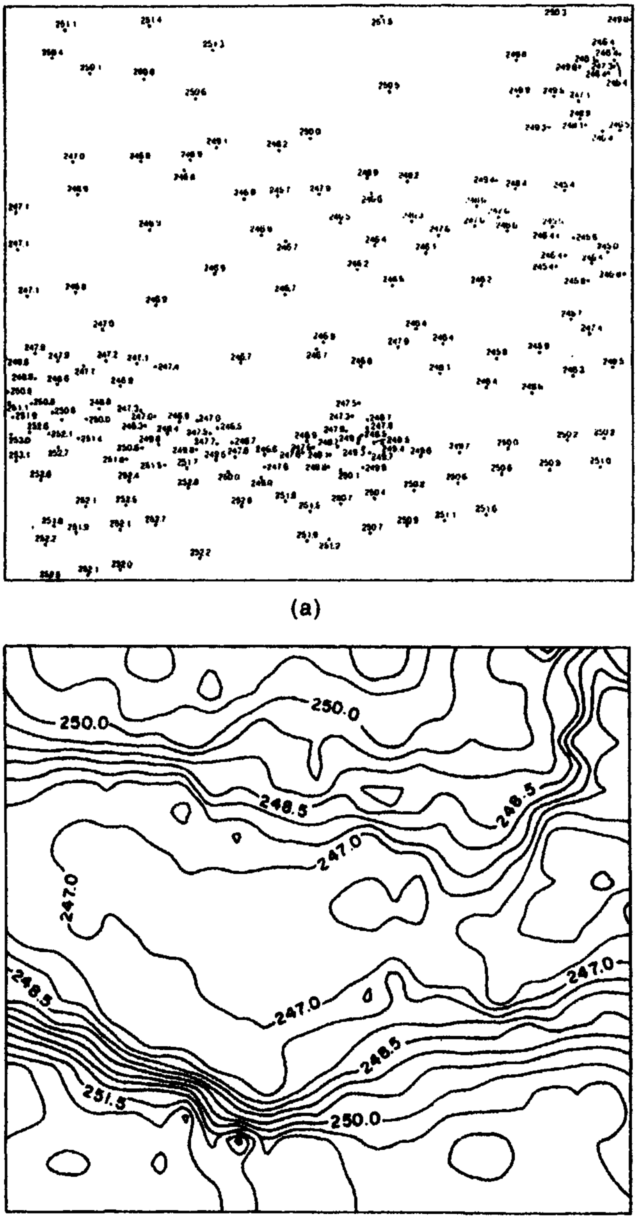
Figure 6.0.1 Choropleth map: Saskatoon Census Metropolitan Area, Canada. Time period of immigration of the majority of the immigrants by census tract, 1981.

changes only produce effects at locations in the vicinity of the changes. Consequently, fairly large systems of equations are needed to evaluate $f(x)$ for global interpolants, which means that exact versions are not feasible when n is large, say in the order of 100–200 data sites. In contrast, local interpolants typically involve only small systems of equations although there may be a large number of these. In this chapter we consider local interpolants exclusively since global interpolants do not involve Voronoi diagrams.

We focus our discussion on $\mathbb{R}^2$. Although the methods described can be extended in theory to $\mathbb{R}^3$, complications created by the bounded nature of S quickly dominate the interpolation. Also the majority of applications relate to $\mathbb{R}^2$. Typical examples of data values include soil and rock types, elevations of a geological strata, barometric pressure and thickness of an ore body or of a manufactured part. In such cases our choice of an interpolant (surface) will be influenced by the nature of the data values. Values measured on a nominal scale, such as rock and soil types, require a piecewise continuous surface which may be represented in two dimensions as a k-phase mosaic (or in cartographic terms, a choropleth map) (see Figure 6.0.1 where $k = 3$). Other values, such as elevation, which are continuous random variables, require at least a continuous surface. These can be represented in a number of ways including contour (isarithmic) maps, perspective views, hill shading and slope maps (see Figures 6.0.2(b), (c), (d) and (e), respectively). The nature of the continuity of the interpolant (surface) is designated C^k which indicates that the kth or lower order derivatives of the interpolant are all continuous. Thus, for a C^0 interpolant the derivatives are in general discontinuous at the data sites so that a contour map produced from it will appear angular (see Figure 6.0.3). In contrast, a C^1 interpolant such as that used in Figure 6.0.2(b) produces smooth contours. For most practical applications, such as those listed below, C^1 interpolants are sufficient, particularly since, with the exception of special cases, higher order smoothness does not appear to be detectable by the human eye (Sibson, 1981, p. 22).

In terms of the uses of the interpolant, by far the most common is simply to produce a contour map or other visual representation of the surface. Other uses involve calculating some property of the surface at a specified location, calculating the volume under the surface, as in the estimation of ore bodies, and calculating global statistics for the surface, such as estimating the mean precipitation over a region.

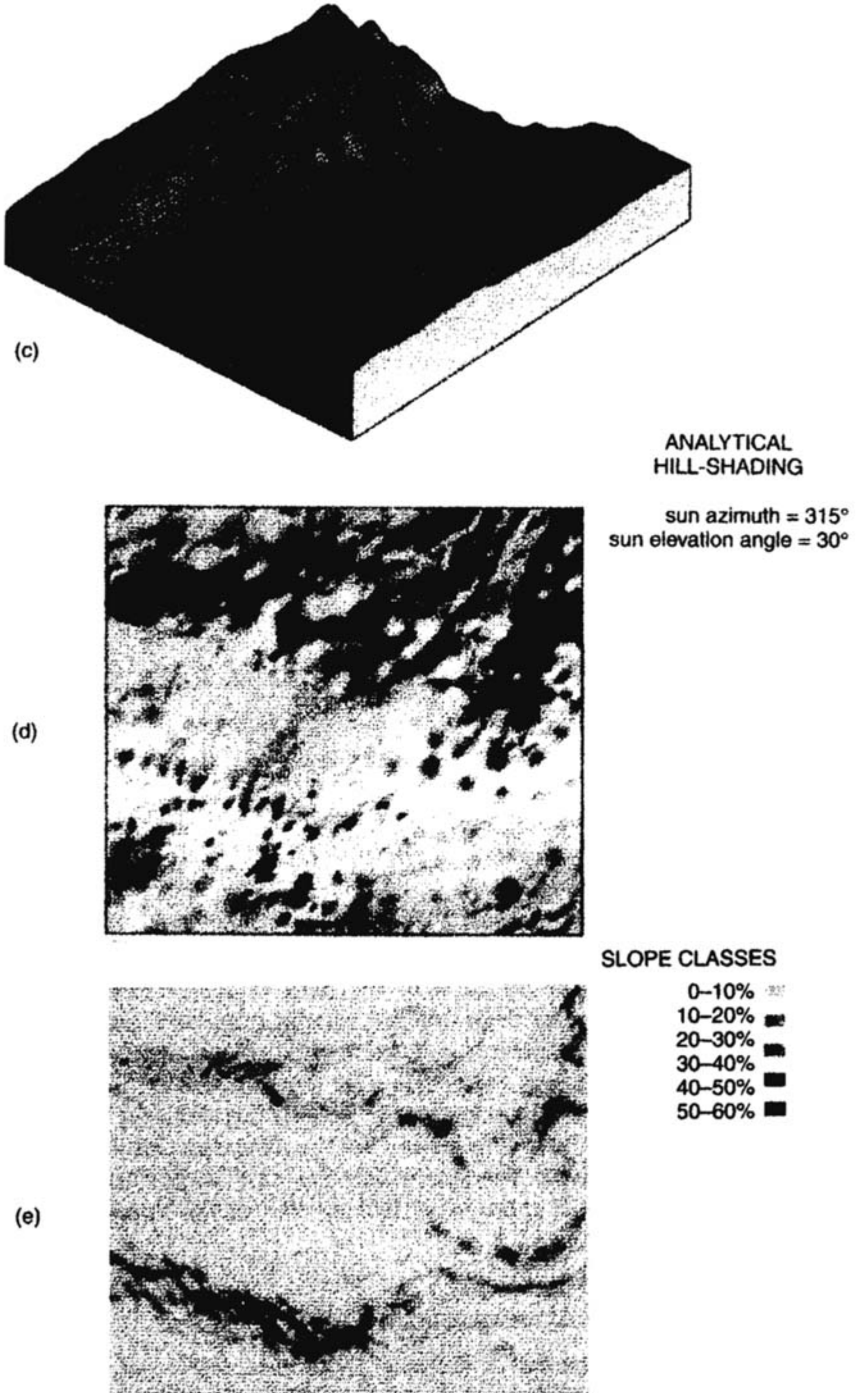
This chapter considers only those interpolation procedures that involve Voronoi diagrams or Delaunay triangulations. We refer to these as *polygonal* and *triangular methods*, respectively, although in view of the duality between Voronoi diagrams and Delaunay triangulations, this distinction is one adopted primarily for the convenience of presentation. General reviews of spatial interpolants can be found in Rhind (1975), Schumaker (1976), Schut (1976), Barnhill (1977), Ripley (1981, Chapter 4), Franke (1982), Sabin (1982), Lam (1983), Gold (1989a), Franke and Nielson (1991) and Watson (1992). Section 6.1 deals with polygonal methods while triangular methods are the subject of Section 6.2. The methods described in Section 6.1 include both the use of the ordinary Voronoi diagram and the ordered order-k Voronoi diagram. The main thrust of Section 6.2 is a consideration of the merits of using triangular domains and the characteristics of different triangulations, including the Delaunay triangulation. Initially, this discussion assumes that S is convex, usually rectangular. Section 6.3 considers the modifications necessary to extend the Delaunay triangulation to situations in which S is neither convex nor simply connected. This section also considers ways of incorporating specific data features into the triangulation. In Section 6.4 we consider how the Delaunay triangulation can be used in approximating surfaces with a predefined level of accuracy. In Section 6.5 we concentrate on triangular and tetrahedral meshes that are useful for solving partial differential equations. Finally, in Section 6.6 we note that a tessellation may be considered an ordering of the data sites and we examine how the Voronoi and Delaunay tessellations can be used in this regard for multivariate data.



(a)

(b)

Figure 6.0.2 Different representations of a continuous surface. (a) Original data sites and data values. (b) Contour (isarithmic) map. [*See facing page*] (c) Perspective view. View direction (azimuth) = 45°; viewing angle = 30°. (d) Hill shading. Sun azimuth = 35°; sun elevation angle = 145°. (e) Slope. Classes: 1. 0–10%; 2, 10–20%; 3, 20–30%; 4, 30–40%; 5, 40–50%; 6, 50–60%.


(c)
ANALYTICAL HILL-SHADING
sun azimuth = 315°
sun elevation angle = 30°
(d)
SLOPE CLASSES
0–10%
10–20%
20–30%
30–40%
40–50%
50–60%
(e)

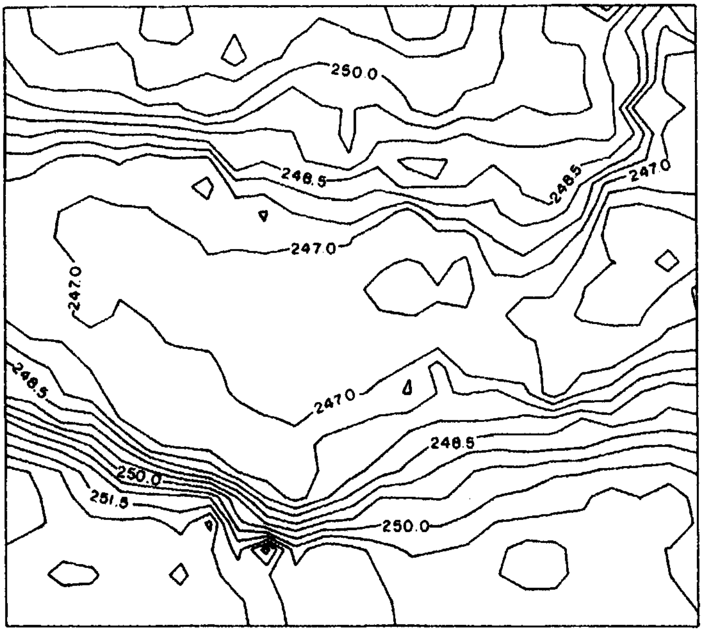


Figure 6.0.3 Contour map of data in Figure 6.0.2 (a) using a C^0 interpolant.

6.1 POLYGONAL METHODS

In general all local interpolants represent the value of the surface $f(\boldsymbol{x})$ at an arbitrary point p located at $\boldsymbol{x}$ in S as a weighted, usually linear function of the values at a set of 'nearby' data sites, $D(p)$ $(D(p) \subset P)$, so that

$$f(\boldsymbol{x}) = \sum_{i=1}^{n_D} w_i z_i, \quad p_i \in D(p), \tag{6.1.1}$$

where n_D is the number of nearby sites and w_i is the weight attached to p_i. Local interpolants differ in terms of how the nearby data sites are selected and how the weights are attached to the data values at those sites. The extent to which these two decisions are related may vary considerably in that different weighting schemes may be applied to the same set of nearby data sites. Since, with one exception (see Section 6.1.2), the weighting procedures do not involve concepts of Voronoi diagrams, our presentation will emphasize the methods of selecting nearby data sites. As Tobler (1975) indicated, there are many ways in which this can be performed. In this chapter we consider only those ways that involve either Voronoi diagrams or Delaunay triangulations.

6.1.1 Nearest neighbour interpolation

Perhaps the simplest means of defining $D(p)$ is to select only the nearest data site to p and assign the associated data value to z. Thus,

$$f(\boldsymbol{x}) = z_i, \quad d(p,p_i) < d(p,p_j), \quad p_i \in P, \ j \neq i, \tag{6.1.2}$$

where $d(p, p_i)$ and $d(p, p_j)$ are the Euclidean distances between p and p_i and p and p_j, respectively. Note that this definition is essentially equivalent to that of an open Voronoi polygon given in Section 2.1. If $\boldsymbol{x}$ is equidistant from two or more members of P, $f(\boldsymbol{x})$ is undefined. Such locations correspond to the boundaries of the Voronoi polygons of $\mathcal{V}(P)$ so that the resulting surface is discontinuous. In this particular application the individual polygons are usually referred to as *proximal polygons* (Peucker and Chrisman, 1975).

Such an interpolant is most appropriate when z is measured on a nominal scale as with land use, soil, vegetation or rock type whose individual values form patches or phases over S. In such cases once the initial proximal polygons are formed, adjacent polygons with equal values can be merged to form larger patches (see Figure 6.1.1). Arnold and Milne (1984) used this approach to derive soil maps for Greece based on data values collected at data sites spaced every 50–200 m over the study area.

This application may be considered the equivalent of the nearest-point rule for pattern reconstruction in image analysis (Ahuja and Schachter, 1983, Section 1.4.3.3). Here the problem is one of reconstructing a continuous image from the colours observed at a number of fixed sample points. Under the nearest-point rule the colour assigned to each image element is that of the nearest sample point. A parallel procedure is also used in quantization which involves analog to digital conversion (Conway and Sloane, 1993, pp. 56–62) and has applications in digital measuring instruments or recording devices and in digital communications systems. Here the value at any location $\boldsymbol{x}$ is 'rounded off' to that of the closest data site.

Proximal polygon interpolation can also be used when z is not nominally scaled although it will not reproduce the underlying continuous surface. The application which gives rise to the term 'Thiessen polygon' is just such an example (Thiessen, 1911; Horton, 1917). Thiessen's concern was to estimate the mean value $\bar{z}$ of a surface (in his case, precipitation) over a bounded region for which data values z_i only existed for an irregularly spaced set of n data sites (precipitation gauges). Given the spatial distributions of gauges Thiessen felt it inappropriate to estimate $\bar{z}$ simply as

$$\bar{z} = \sum_{i=1}^{n} z_i / n. \tag{6.1.3}$$

Instead he used

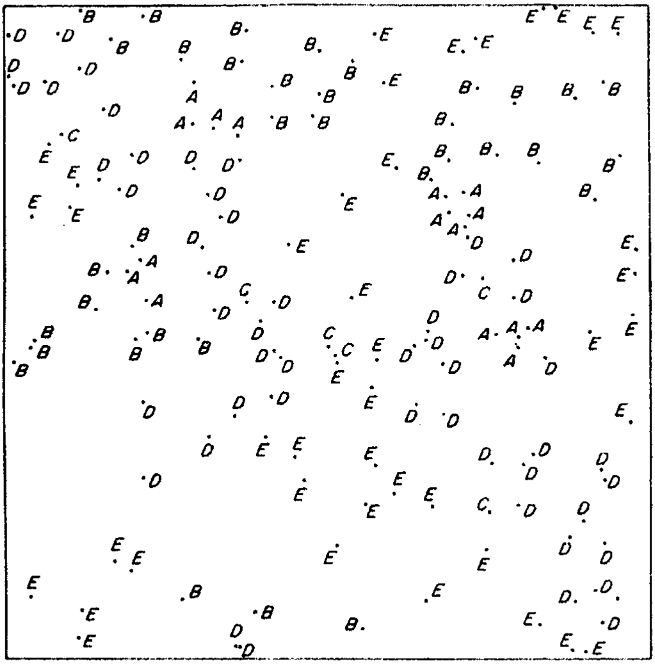
$$\bar{z} = \sum_{i=1}^{n} w_i z_i \Big/ \sum_{i=1}^{n} w_i, \tag{6.1.4}$$

where w_i is the area of the proximal polygon associated with p_i. This approach is very old, but it is still used in meteorology (Chakravarti and Archibold, 1993). This example is also discussed in Chapters 8 and 9.

The same approach is sometimes used in estimating the average grade of an ore deposit, although as befits a different application with different origins, the proximal polygons are usually termed *area of influence polygons* (Popoff, 1966; Reedman, 1979; Hayes and Koch, 1984). The approach involves representing the surface as a set of polygonal prisms whose heights reflect the thicknesses of the ore body (see Figure 6.1.2). The average grade $\bar{z}$ is calculated using expression (6.1.4) with w_i equal to the volume of the prism associated with p_i and z_i equal to the ore grade (see Section 7.1 for more details and additional references).

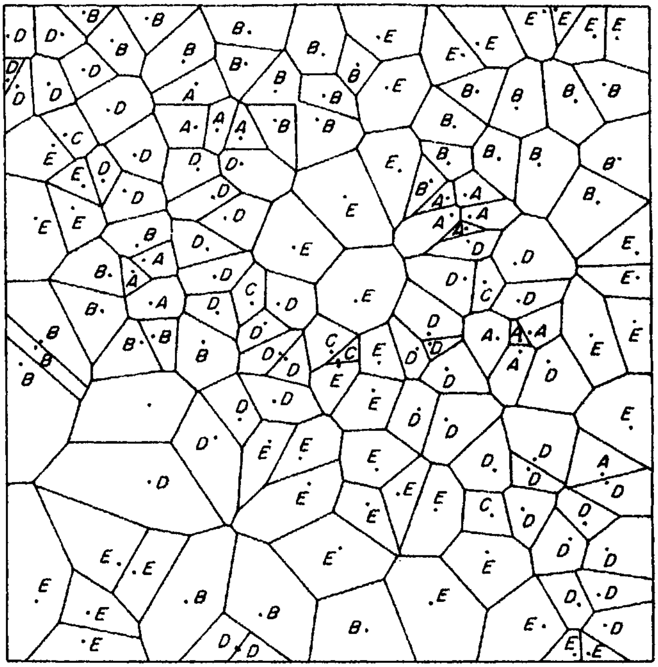
6.1.2 Natural neighbour interpolation

Another way of defining $D(p)$ for an arbitrary point p at location $\boldsymbol{x}$ in S is to identify those members of P which are Voronoi or, to use Sibson's term, 'natural' neighbours of p (Sibson, 1981; Watson and Philip, 1987). This involves generating the Voronoi diagram of $P \cup \{p\}$. Like all the members of P, p has its associated Voronoi region, $V(p)$ (the shaded polygon in Figure 6.1.3). Those points of P whose Voronoi regions are contiguous to $V(p)$ form

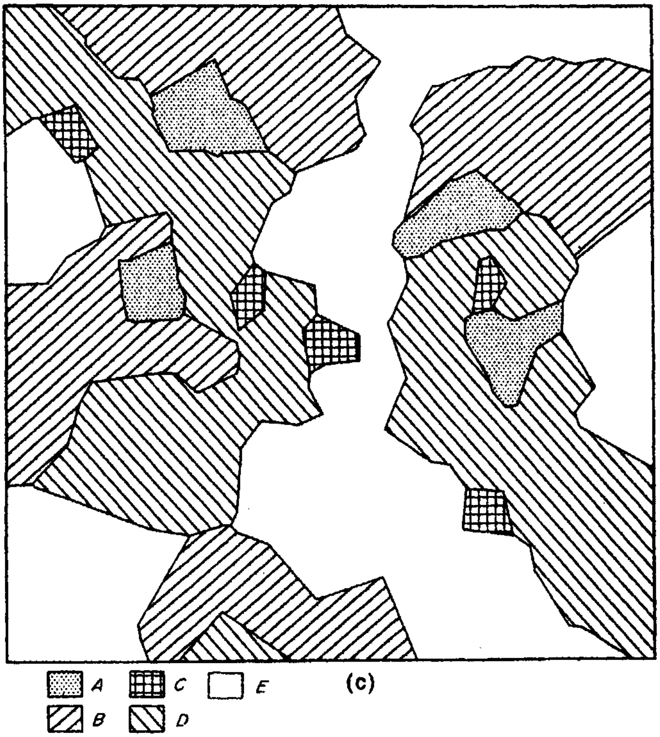


(a)

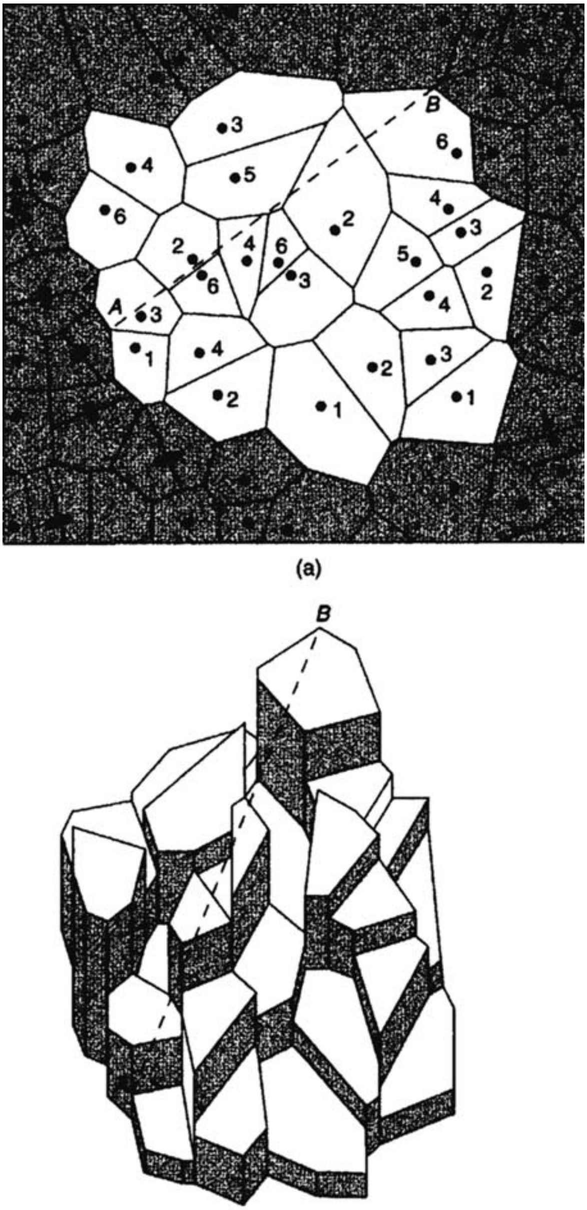
Figure 6.1.1 Creation of a discontinuous surface by nearest neighbour interpolation. (a) Data sites and data values. [*See facing page*] (b) Voronoi diagram (proximal polygons). (c) Final surface.



(b)



(c)



(a)

(b)

Figure 6.1.2 Representation of a surface as a set of polygonal prisms. (a) Voronoi diagram of data sites (value at data site represents thickness of ore body; shaded polygons are not represented in (b)). (b) Polygonal prisms. (Common dashed line *AB* is shown to facilitate comparison.)

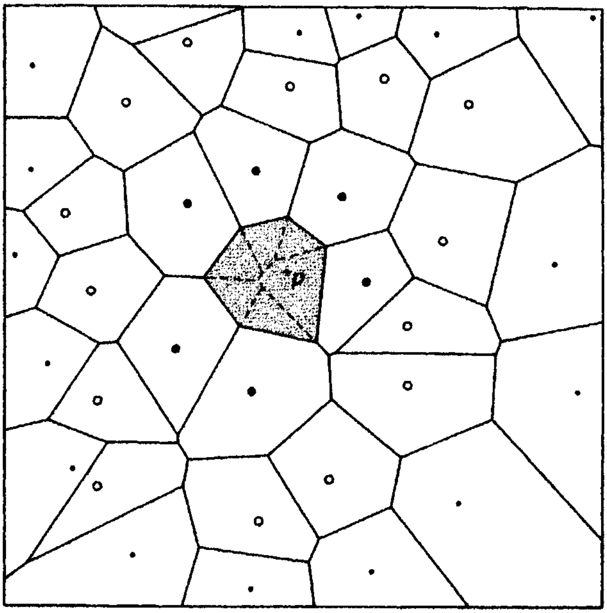


Figure 6.1.3 Voronoi diagram of $P \cup \{p\}$. *Key*: ●, natural neighbours of p; points contributing to interpolated value at p using equation (6.1.9); ○, points contributing to interpolated values at p using equation (6.1.13); ·, other points of p.

the natural neighbours of p (the filled circles in Figure 6.1.3). The data values at the natural neighbours are then used in equation (6.1.1).

A special case of spatial interpolation is the missing value problem where the unknown value of a variable at a given location must be estimated from the known values of the variable at other locations. In their study of rainfall data for Kansas and Nebraska, U.S.A., Haining *et al.* (1984) used natural neighbours to identify which weather stations to use in estimating missing values at other stations.

There are also other situations besides interpolation where it may be necessary to identify nearby points and in which natural neighbours have been proposed as the solution (Besag, 1974, 1975; Ord, 1975). A particular instance is the study of spatial autocorrelation (Goodchild, 1986; Griffith, 1987; Odland, 1988). This involves measuring the extent of dependence in the values of a variable observed at nearby locations. Spatial autocorrelation measures ρ often have the general form

$$\rho = \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} z_i z_j, \qquad (6.1.5)$$

where z_i and z_j are the values of the variable at data sites p_i and p_j, respectively, and w_{ij} is a weight reflecting the spatial relationship between p_i and p_j. In a study of the grain handling system of the province of Manitoba in

Canada from 1943 to 1975, Griffith (1982) suggested setting $w_{ij} = 1$ in equation (6.1.5) if p_i and p_j are natural neighbours and $w_{ij} = 0$, otherwise. The same procedure was taken in a study of loblolly pine stands by Reed and Burkhart (1985).

It is worth noting that the natural neighbour definition has also been frequently proposed as a way of identifying neighbours in a variety of other circumstances. In $\mathbb{R}^3$ these include identifying neighbouring atoms, molecules or particles in both crystalline and amorphous structures including monatomic liquids, solutions, metallic glasses, metals and alloys and for identifying neighbouring cracks in percolation structures. The definition of neighbouring capillaries in muscle tissue is an example involving two-dimensional sections of material in $\mathbb{R}^3$, while the recognition of neighbours for plants and animals and for stores in competitive situations in ecology and retailing, respectively, are examples of the use for phenomena represented in $\mathbb{R}^2$. Discussion of these uses together with the appropriate references is contained in Chapter 7.

If the members of P have associated weights (see Section 3.1) these may be combined with natural neighbour concepts to define other types of neighbour relationships. One example occurs in the field of spatial interaction modelling (Haynes and Fotheringham, 1984) which attempts to predict the movement of people, goods, information, etc. between places located in $\mathbb{R}^2$. Typically, the amount of interaction I_{ij} between two places p_i and p_j with weights w_i and w_j, respectively (where the weights usually reflect some measure of the sizes of the places such as their populations), is considered to be directly proportional to the product of their weights and inversely proportional to the distance separating them so that

$$I_{ij} = k w_i w_j d(p_i, p_j)^{-\beta}, \tag{6.1.6}$$

where $d(p_i, p_j)$ is the distance between p_i and p_j, k is a constant, and β is the distance exponent ($\beta > 0$). A variant of this general interaction formulation, known as the intervening opportunity model, was proposed by Stouffer (1940). He replaced the direct measure of distance used in equation (6.1.6) by an indirect one and argued that

$$I_{ij} = k w_i w_j O_{ij}^{-1}, \tag{6.1.7}$$

where O_{ij} is the number of opportunities intervening in $\mathbb{R}^2$ between p_i and p_j which can be measured by

$$O_{ij} = \sum_{\substack{k=1 \\ k \neq i,j}}^{l} w_k \tag{6.1.8}$$

where l is the number of points in P which intervene between p_i and p_j. The problem with this modification is the definition of the intervening places. One way is to generate the Voronoi diagram $\mathcal{V}(P)$ of P and to consider the places intervening between p_i and p_j as those whose Voronoi polygons are intersected by the line segment, $\overline{p_i p_j}$. For example, in Figure 6.1.4(a) the places intervening between p_{18} and p_{35} are p_6, p_{13} and p_{30}. In those circumstances where w_i is recorded as on ordinal rank r_i (the larger the rank the more important the place), it is often suggested that interaction between p_i and p_j will only occur if no points with higher ranks intervene between the two. Thus, to determine which places interact, we can use the same procedure as above. In this case p_i and p_j will interact if the line segment $\overline{p_ip_j}$ does not intersect a Voronoi polygon of a point with a higher rank than either p_i or p_j. Thus, in Figure 6.1.4(a) if the numbers associated with the points represent their ranks, the points which interact with p_{18} are shown by the line segments linking them to p_{18}.

Elliott (1981, 1982, 1983, 1985) has proposed a further variant of this method. He labelled the set of places for which interaction occurs with p_i as the *surrounding larger neighbours* (SLNs) of p_i. The SLNs of p_i may be defined by generating $\mathcal{V}(P)$ (see Figure 6.1.4(a)) which in turn may be represented by the dual Delaunay triangulation, $\mathcal{D}(P)$ (see Figure 6.1.4(b)). If we consider $\mathcal{D}(P)$ as a graph (see Section 1.3.6) we can measure how far each point p_j is from p_i in terms of the number of edges e_j of $\mathcal{D}(P)$ (see Figure 6.1.4(b)). Point p_j is an SLN of p_i if it has a higher rank than p_i and a path of length e_j can be traced from p_j to p_i which does not pass through any other vertex of $\mathcal{D}(P)$ with a higher rank than p_j. Figure 6.1.4(c) shows the SLNs for p_{18}. These may be compared with the higher ranked places defined by the previous method (the solid lines in Figure 6.1.4(a)). Elliott (1983, 1985) made use of SLNs derived in this way in modelling inter-city airline passenger traffic in the United States.

Sibson (1981) has also proposed using the natural neighbour concept in deriving the appropriate weights used in equation (6.1.1). We know from our discussion of ordered order-k Voronoi diagrams in Section 3.2 (Property OOK2) that in the Voronoi diagram of $P \cup \{p\}$ the Voronoi region of p, $V(p)$, can be exhaustively subdivided into ordered order-2 regions $V((p,p_j))$, such that the first and second nearest points of $P \cup \{p\}$ from any location in $V((p,p_j))$ are p and p_j, respectively (see Figure 6.1.3). Let $\kappa(\boldsymbol{x},\boldsymbol{x}_j)$ be the area of $V((p,p_j))$. In the special case $p = p_j$, we set $\kappa(\boldsymbol{x},\boldsymbol{x}_j) > 0$ and $\kappa(\boldsymbol{x}, \boldsymbol{x}_i) = 0$ for $i \neq j$. Sibson used the normalized values $\lambda(\boldsymbol{x},\boldsymbol{x}_j) = \kappa(\boldsymbol{x},\boldsymbol{x}_j)/\sum_i \kappa(\boldsymbol{x}, \boldsymbol{x}_i)$ in deriving the weights for z_j in expression (6.1.1). These values have two fundamental properties

Property LCP1 $\lambda(\boldsymbol{x},\boldsymbol{x}_j)$ is a continuous function of $\boldsymbol{x}$ which is continuously differentiable everywhere except for the data sites.

Property LCP2 Provided that $V(p)$ does not meet the boundary of S,

$$\sum_i \lambda(\boldsymbol{x},\boldsymbol{x}_i)\boldsymbol{x}_i = \boldsymbol{x}.$$

A proof of LCP2 is given by Sibson (1980a) (see Section 3.2.2, Property OOK7). In view of this representation of $\boldsymbol{x}$, Sibson labelled the $\lambda(\boldsymbol{x},\boldsymbol{x}_j)$'s the *local coordinates* of point p. If $V(p)$ meets the boundary of S, $\lambda(\boldsymbol{x},\boldsymbol{x}_j)$ does not have any physically useful meaning because it is not independent of the

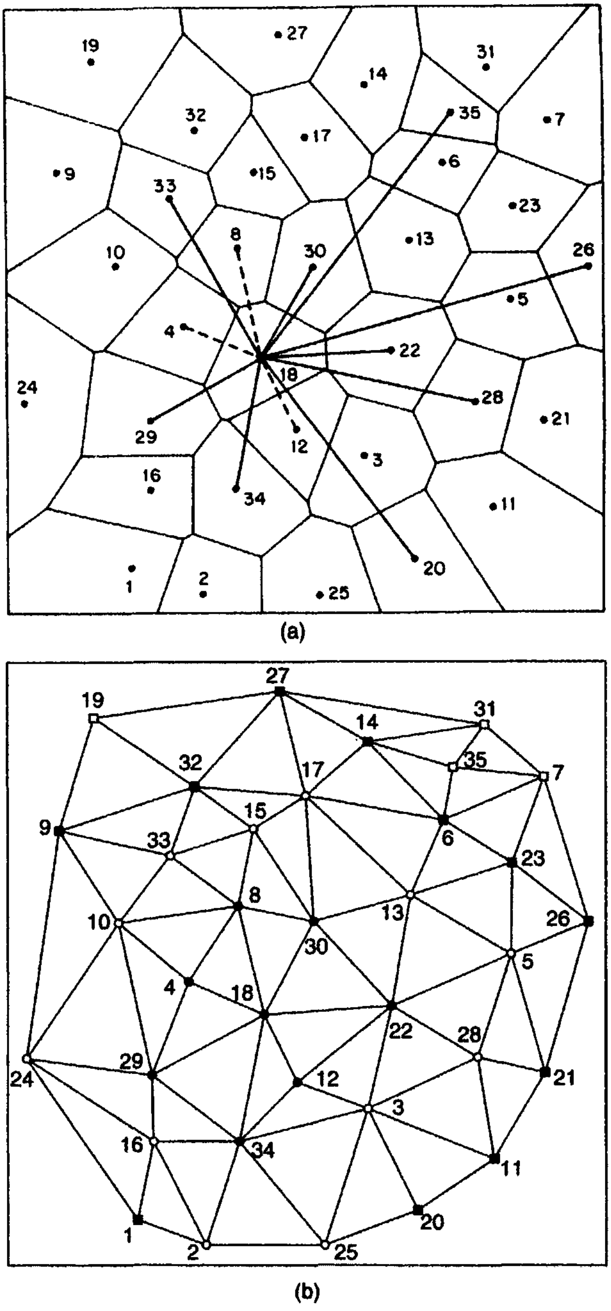


Figure 6.1.4 Definition of intervening opportunities. (a) Places interacting with p_{18}: ——— higher ranked places; - - - - lower ranked places. (b) Distance from p_{18} in terms of number of Delaunay edges; ●, 1; ○, 2; ■, 3; □, 4.

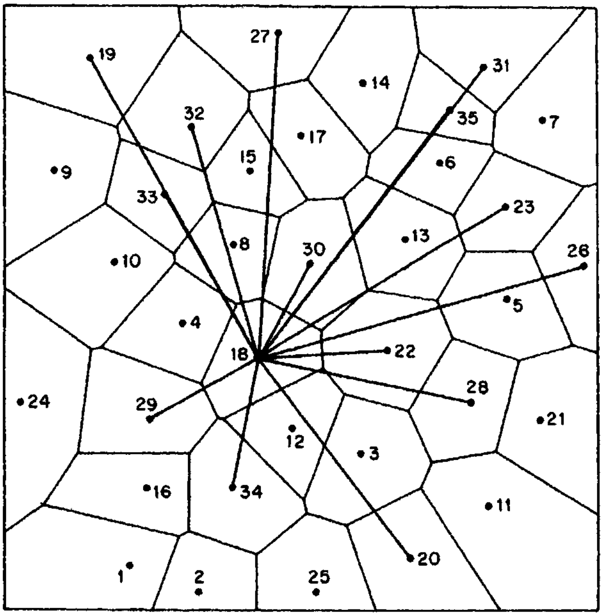


Figure 6.1.4 (c) Surrounding larger neighbours of p_{18}.

shape of region S. Indeed, in what follows we use the value of $\lambda(\boldsymbol{x}, \boldsymbol{x}_j)$ only for points p such that $V(p)$ does not meet the boundary of S.

A C^0 interpolant, $f_0(\boldsymbol{x})$, can be obtained by using the local coordinates directly so that

$$f_0(\boldsymbol{x}) = \sum_j \lambda(\boldsymbol{x}, \boldsymbol{x}_j) z_j. \tag{6.1.9}$$

Thus the data values that contribute to the interpolated values at p in Figure 6.1.3 are those for the points shown by filled circles. If the data values come from a linear function of the form

$$z = \alpha + \beta^{\mathrm{T}} \boldsymbol{x},$$

$f_0(\boldsymbol{x})$ correctly reproduces it, which can be shown as follows. Substituting

$$z_j = \alpha + \beta^{\mathrm{T}} \boldsymbol{x}_j$$

in equation (6.1.9) gives

$$\begin{aligned} f_0(\boldsymbol{x}) &= \sum_j \lambda(\boldsymbol{x}, \boldsymbol{x}_j)\,(\alpha + \beta^{\mathrm{T}} \boldsymbol{x}_j) \\ &= \alpha + \beta^{\mathrm{T}} \sum_j \lambda(\boldsymbol{x}, \boldsymbol{x}_j) \boldsymbol{x}_j. \end{aligned}$$

Using Property LCP2,

$$z = \alpha + \beta^{\mathrm{T}} \sum_j \lambda(\boldsymbol{x},\boldsymbol{x}_j)\, \boldsymbol{x}_j.$$

Thus, $f_0(\boldsymbol{x}) = z$.

Sibson also used the local coordinates in deriving a C^1 interpolant. He fitted planes $\zeta(\boldsymbol{x}_i)$ for the value at $\boldsymbol{x}_i$, using a weighted least squares fit with weights

$$w(\boldsymbol{x}_i,\boldsymbol{x}_j) = \lambda(\boldsymbol{x}_i,\boldsymbol{x}_j) / \, d(\boldsymbol{x}_i,\boldsymbol{x}_j)^2, \tag{6.1.10}$$

where

$$d(\boldsymbol{x}_i,\boldsymbol{x}_j)^2 = (\boldsymbol{x}_j - \boldsymbol{x}_i)^{\mathrm{T}} (\boldsymbol{x}_j - \boldsymbol{x}_i).$$

Explicitly, $\boldsymbol{b}_i$, the gradient at $\boldsymbol{x}_i$, is given by

$$\boldsymbol{b}_i = H_i^{-1} \boldsymbol{q}_i, \tag{6.1.11}$$

where

$$H_i = \sum_j w(\boldsymbol{x}_i,\boldsymbol{x}_j)(\boldsymbol{x}_j - \boldsymbol{x}_i)^{\mathrm{T}} (\boldsymbol{x}_j - \boldsymbol{x}_i),$$

$$\boldsymbol{q}_i = \sum_j w(\boldsymbol{x}_i,\boldsymbol{x}_j)(\boldsymbol{x}_j - \boldsymbol{x}_i)(z_j - z_i).$$

Thus,

$$\zeta_i(\boldsymbol{x}) = z_i + \boldsymbol{b}_i^{\mathrm{T}} (\boldsymbol{x} - \boldsymbol{x}_i)$$

is the linear function through $(\boldsymbol{x}_i, z_i)$ with slope $\boldsymbol{b}_i$. Combining these functions with weights $W(\boldsymbol{x}, \boldsymbol{x}_i) = \lambda(\boldsymbol{x}, \boldsymbol{x}_i)/d(\boldsymbol{x}, \boldsymbol{x}_i)$, gives the interpolant

$$\begin{aligned} f^*(\boldsymbol{x}) &= z_i = \boldsymbol{x}_i, && \text{if } \boldsymbol{x} = \boldsymbol{x}_i \\ f^*(\boldsymbol{x}) &= \sum_j \left[W(\boldsymbol{x},\boldsymbol{x}_j)\zeta_j(\boldsymbol{x})\right] \Big/ \sum_j W(\boldsymbol{x},\boldsymbol{x}_j) && \text{if } \boldsymbol{x} \neq \boldsymbol{x}_i. \end{aligned} \tag{6.1.12}$$

The final C^1 interpolant, $f_1(\boldsymbol{x})$, is a linear combination of $f_0(\boldsymbol{x})$ and $f^*(\boldsymbol{x})$ chosen to ensure that it fits exactly restricted quadratic surfaces of the form

$$z = \alpha + \beta^{\mathrm{T}} \boldsymbol{x} + \gamma \boldsymbol{x}^{\mathrm{T}} \boldsymbol{x}.$$

Specifically,

$$f_1(\boldsymbol{x}) = \frac{\left\{\left[\sum_i \lambda(\boldsymbol{x},\boldsymbol{x}_i)\, d(\boldsymbol{x},\boldsymbol{x}_i)\right] \Big/ \sum_i W(\boldsymbol{x},\boldsymbol{x}_i)\right\} f_0(\boldsymbol{x}) + \sum_i \left[\lambda(\boldsymbol{x},\boldsymbol{x}_i)\, d(\boldsymbol{x},\boldsymbol{x}_i)^2\right] f^*(\boldsymbol{x})}{\left[\sum_i \lambda(\boldsymbol{x},\boldsymbol{x}_i)\, d(\boldsymbol{x},\boldsymbol{x}_i)\right] \Big/ \sum_i W(\boldsymbol{x},\boldsymbol{x}_i) + \sum_i \lambda(\boldsymbol{x},\boldsymbol{x}_i)\, d(\boldsymbol{x},\boldsymbol{x}_i)^2}. \tag{6.1.13}$$

Note that while equation (6.1.9) is valid for all locations which satisfy the condition imposed in Property LCP2 above, equation (6.1.13) holds only for a more limited set of locations. This set consists of those locations p for which neither $V(p)$ nor any Voronoi polygon adjacent to $V(p)$ in the Voronoi diagram $\mathcal{V}(P \cup \{p\})$ meets the boundary of S. The set of data values which contribute to the value of $f^*(x)$ at p in Figure 6.1.3 are those for the points shown by both filled and unfilled circles.

Sibson's basic interpolant, equation (6.1.9), has been studied further from several points of view. Farin (1990) considered its role in surface modelling. He notes that, if the number of the natural neighbours is four, Sibson's interpolant agrees with the classical bilinear interpolant. He also points out that Sibson's interpolant is not 'idempotent' in the following sense. Suppose that we generate Sibson's interpolant $f_0(x)$ using the height data z_i at n sites $x_i, i = 1, 2, \dots, n$, and next we add a new site x_{n+1} and the height $z_{n+1} = f_0(x_{n+1})$. If we generate Sibson's interpolant $f_0(x)'$ using the $n+1$ height data, we have $f_0(x) \neq f_0(x)'$ in general. This is a somewhat strange property for the interpolation system. Hence, we have to be careful in applications.

Piper (1993) gave an explicit expression of the gradient of Sibson's local coordinate together with its geometrical interpretation, while Sambridge *et al.* (1995) produce expressions for the derivatives of the interpolated function. Gross (1995) generalized Sibson's interpolant so that it can be used for the interpolation of continuously distributed height data along closed curves, where the Voronoi diagram for polygons plays an important role.

Finally, we note that Lowell (1994) uses order-2 Voronoi diagrams to interpolate nominal scale data, while Edwards (1993) and Edwards and Moulin (1995) use them to operationalize a wide range of linguistic spatial concepts such as 'between', 'behind' and 'close'. Watson (1985, 1988) extends the concept of natural neighbour order to n-component directional data, as normalized onto an n-dimensional sphere.

6.2 TRIANGULAR METHODS

In this approach interpolation of z at an arbitrary location p in S involves three steps. First, we construct a triangulation $\mathcal{T}(P)$ of the data sites in P using the data sites as vertices of the triangles. Next we identify the triangle $T(p)$ of $\mathcal{T}(P)$ which contains p and define the set of nearby points $D(p)$ as the vertices of $T(p)$. Finally, we compute z using the values of z_i $(i = 1, 2, 3)$ for the members of $D(p)$.

The same procedure is also used in two other applications. One is in finite element methods, which will be discussed in Section 6.5. The other is the estimation of ore reserves where triangular prisms may be used in place of polygonal ones (see Section 6.1.1) (Harding, 1920–21, 1923; Popoff, 1966, pp. 78–84; Reedman, 1979, pp. 438–439).

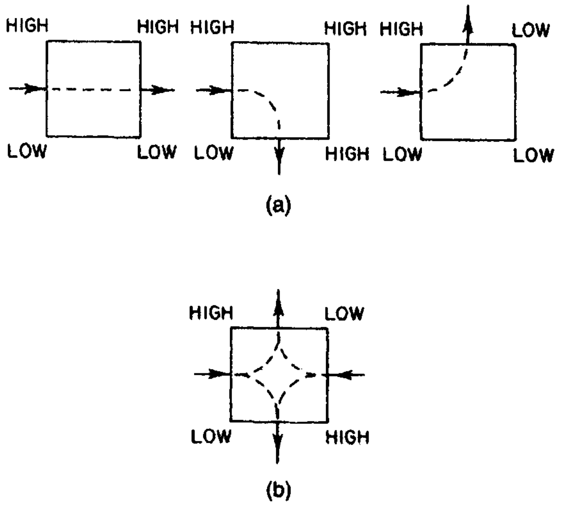
In all of these application areas the methods used in the last two steps are independent of the nature of $\mathcal{T}(P)$ and since they do not involve Voronoi or Delaunay concepts, we confine our attention in this chapter to the construction of $\mathcal{T}(P)$. Reviews of different triangle based interpolants can be found in Barnhill (1977), Gold (1980), McCullagh (1981), Watson and Philip (1984a) and Watson (1992).

When the main purpose of interpolation is to create contours from data values at a set of data sites irregularly spaced over S, the triangulation approach possesses several advantages over the other most frequently used approach, the regular grid approach. An instance of this is a *digital terrain model* (DTM) where the data values are elevations of some region of a planetary surface. In this context $\mathcal{T}(P)$ is referred to as a *triangulated irregular network* (TIN). In the regular grid approach a regular, usually square grid is superimposed on S and values of z are estimated at the grid intersections on the basis of their spatial relationships to the data values for the members of P. There are a variety of ways that such an estimation can be performed (see Monmonier, 1982, pp. 58–65; Watson and Philip, 1985; Jones *et al.*, 1986, pp. 44–57 for reviews). The main advantage of the regular grid approach is that the set of grid intersections may be stored as a matrix in which both their locations and topological relationships are defined implicitly so that it is necessary only to store the z values assigned to them. In contrast, the topological relationships of the facets of a TIN must be stored explicitly, although this is not necessary if they are to be used only for hill shading. If the facets of the TIN are used as the units of storage, each facet requires six pointers, three for the vertices and three to identify the neighbouring triangles. Since the total number of triangles in a TIN is $2n - n_c - 2$, where n_c is the number of data sites on the boundary of the convex hull of P (see Property D11, Section 2.4), this scheme requires a total of $12n - 6n_c - 12$ pointers. Alternatively, we can list in a consistent order (e.g. clockwise from the north), the neighbouring data sites of each data site. This requires a total of $6n - 2n_c - 6$ pointers, equal to twice the number of edges in the TIN (see Property D11, Section 2.4) for about a 50% savings in storage.

Topographical surfaces are not stationary (see Section 1.3.3) since the variation in terrain height changes from one landform to another (compare a jagged mountain range with a coastal plain). Thus, the spacing of points of any regular grid superimposed on S must be adjusted to capture height changes in areas of greatest variability. This means that many of the grid points will be redundant in areas of constant relief. TINs with their natural variable resolution can easily adapt to such variations. Mark (1975) suggests that in a typical DTM 14 regular data points are required for each irregular data point to produce the same level of representation. Also it is much easier to incorporate linear and other required features (such as ridge and valley lines) in a TIN structure (see Section 6.3). In such circumstances Peucker *et al.* (1976) suggest that the ratio of regular to irregular data points rises to about 287:1. Furthermore, since the data sites are the vertices of the TIN, the data values will be interpolated exactly. It is unlikely that this will happen in the regular grid approach which, as a result, should more properly be considered a smoothing procedure rather than an interpolation one.

The facets of a TIN also avoid the 'saddle point' problem which can arise with the regular grid. If the data values associated with four neighbouring grid intersections take any of the forms in Figure 6.2.1(a), then the path of the contour through the grid square defined by the intersections is unambiguous. However, if the data values are configured as in Figure 6.2.1(b), then several different paths for the contour are possible. Finally, TINs also appear to have advantages in terms of the computational effort and speed involved in calculating the final contours. Although there is little systematic information on this, at least one study (McCullagh and Ross, 1980) suggests that the identification of the TIN structure takes on average only one-tenth of the time required to compute the values of z at the grid intersections. The computing time for C^0 contours is approximately equal for the regular grid and the TIN while a C^1 interpolation on the former takes about two-thirds of the time taken on the latter. Nevertheless, this suggests that overall the TIN method is at least five times faster than the regular grid in this context. Computational speed for an actual contouring situation reported by Philip and Watson (1982) supports this suggestion. On the other hand, Kumler (1994) reaches an opposite conclusion when analysing twenty-five different physiographic maps of areas in the United States. He examines different ways of selecting points to construct TINs from regular grid map data, compares TINs and regular grids, and concludes that for the data he examined, TINs are never more efficient.

For n data sites in P, let E be the set of $n(n-1)/2$ line segments (edges) joining all pairs of data sites. Then $\mathscr{T}(P)$ is a maximal subset of E in which no two edges intersect except at their end points. $\mathscr{T}(P)$ is not unique. The number of possible triangulations, $N[\mathscr{T}(P)]$, of P depends in part on the location of the members of P in S and no general expression exists for evaluating $N[\mathscr{T}(P)]$, although enumeration methods (Masada *et al.*, 1996b) and the upper bounds (Dey, 1993; Dey and Shah, 1995) are studied. However, each triangulation has $3(n-1) - n_c$ edges and $2(n-1) - n_c$ triangles (see Section 2.4).



(a)

(b)

Figure 6.2.1 The saddle-point problem. (a) Unambiguous paths for the contour. (b) Ambiguous paths for the contour.

In the three areas of application mentioned above it is desirable that the triangulation selected avoids the occurrence of triangles with two highly acute interior vertex angles and that the triangles be as equiangular as possible. This is because elongated triangles involve greater separation of data sites and reduce the ability of the triangulation to reflect local variation in Z. Whitney (1929) appears to be the first to have recognized the importance of this consideration to spatial interpolation. In Section 2.4 (Property D15) we saw that the Delaunay triangulation, $\mathscr{D}(P)$, of P satisfies a local max-min angle criterion (also referred to as an equiangular criterion) in that the diagonal of every strictly convex quadrilateral, Q, in $\mathscr{D}(P)$ is chosen so that it maximizes the minimum interior angle of the two resulting triangles (see Figure 2.4.13 in Section 2.4). Sibson (1978) demonstrated that $\mathscr{D}(P)$ is the only triangulation with this property. Furthermore, as reported in Property D16 in Section 2.4.2, $\mathscr{D}(P)$ is also globally equiangular in that it maximizes the minimum angle occurring in $\mathscr{T}(P)$.

Lawson (1977, pp. 178–181) demonstrated that the equiangular criterion is equivalent to another, namely the circle criterion. If k is a circle passing through any three vertices of Q, and the fourth vertex is interior to k, we select the diagonal linking this vertex to the opposite vertex (see Figure 6.2.2(a)). If the fourth vertex is exterior to k we select the other diagonal (Figure 6.2.2(b)). If the fourth vertex is on k we select either diagonal (Figure 6.2.2(c)). This criterion avoids the selection of triangles with relatively large circumcircles and Watson and Philip (1984b) conjecture that $\mathscr{D}(P)$ is the $\mathscr{T}(P)$ which minimizes the mean circumcircle diameter. It is also known that from any triangulation $\mathscr{T}(P)$ we can reach $\mathscr{D}(P)$ by repeated application of local reconnections of diagonals (Cherfils and Hermeline, 1990; see also Property D16 in Section 2.4).

It has also been suggested (Lawson, 1977; De Floriani *et al.*, 1982, 1983, 1985) that the equiangular criterion of $\mathscr{D}(P)$ is equivalent to the 'Pitteway' criterion proposed by McLain (1976), which requires that any location within any triangle of $\mathscr{D}(P)$ should have one of the vertices of that triangle as its closest data point. However, Sabin (1976) and Ripley (1981, p. 40) show that there are some patterns of data sites for which no $\mathscr{T}(P)$ satisfies the Pitteway criterion. In Section 2.4 (Property D3) we give a necessary and sufficient condition for $\mathscr{D}(P)$ to satisfy the Pitteway criterion.

Additional speculation concerns the relationship of $\mathscr{D}(P)$ to other triangulations, in particular the *minimum length triangulation* (also known as the *minimum weight triangulation* and the *optimal triangulation*), MLT(P), and the '*greedy*' *triangulation*, GT(P). Let $|\mathscr{T}(P)|$ denote the sum of the lengths of the edges in $\mathscr{T}(P)$. Then MLT(P) is the $\mathscr{T}(P)$ which minimizes $|\mathscr{T}(P)|$. MLT(P) is unique for general sites. This triangulation was originally proposed by Bengtsson and Nordbeck (1964) and Düppe and Gottschalk (1970). Steele (1982) showed that for n points ($1 \leq n \leq \infty$) located in a unit square according to a homogeneous planar Poisson point process (see Section 1.3.3), $|\mathrm{MLT}(P)|/n^{1/2}$ tends to a constant as n tends to infinity.

The greedy triangulation GT(P) is the triangulation obtained by consid-

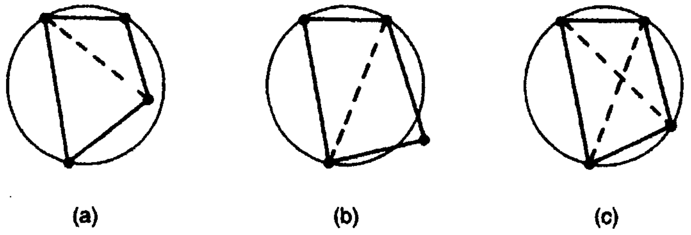


Figure 6.2.2 The circle criterion.

ering all possible edges in order of increasing length and inserting an edge provided that no edge already exists which properly intersects it. If no pair of edges has the same length GT(P) is unique.

Often $\mathcal{D}(P)$, MLT(P) and GT(P) are identical (see Figure 6.2.3(a)). However they need not be (see Figure 6.2.3(b)) as was first noted by Lawson (1977) and Lloyd (1977). In most situations where $\mathcal{D}(P)$, MLT(P) and GT(P) are not identical, GT(P) is a better approximator of MLT(P) than $\mathcal{D}(P)$. For arbitrarily large n, Manacher and Zobrist (1979) constructed sets of n data sites P_1 and P_2 for which

$$|\mathcal{D}(P_1)|/|\mathrm{MLT}(P_1)| \geq \mathrm{O}(n/\log n) \qquad (6.2.1)$$

and

$$|\mathrm{GT}(P_2)|/|\ \mathrm{MLT}(P_2)| > \mathrm{O}(n^{1/3}) \qquad (6.2.2)$$

where $|\cdot|$ is the length of the triangulation as defined above and O(k) is the order of k (see Section 1.3.4). P_2 involves non-convex polygons. However, Lingas (1986a) constructed a positive real $\varepsilon < 1$ for which

$$|\mathrm{GT}(P)|/|\ \mathrm{MLT}(P)| < \mathrm{O}(n^{\varepsilon}) \qquad (6.2.3)$$

holds for any convex, planar point set P with n data sites.

Kirkpatrick (1980) identified arbitrarily large convex sets of n data sites P_3 for which

$$|\mathcal{D}(P_3)|/|\mathrm{MLT}(P_3)| > \mathrm{O}(n), \qquad (6.2.4)$$

and so $\mathcal{D}(P_3)$ is asymptotically no better than an arbitrary triangulation in approximating MLT(P_3). However, Lingas (1986b) proved that when P consists of data sites located in a unit square according to a homogeneous planar Poisson point process (see Section 1.3.3),

$$|\mathcal{D}(P)|/|\ \mathrm{MLT}(P)| < \mathrm{O}(\log n) \qquad (6.2.5)$$

with probability of at least $1 - cn^{\alpha}$, where c and α are constants satisfying $c > 0$ and $\alpha > 1$. Furthermore, Chang and Lee (1984) showed that when the data points are located in this way the average length of $\mathcal{D}$(P), E($|\mathcal{D}(P)|$), is of the same order as that of MLT(P), E($|$MLT(P)$|$); more precisely they proved that

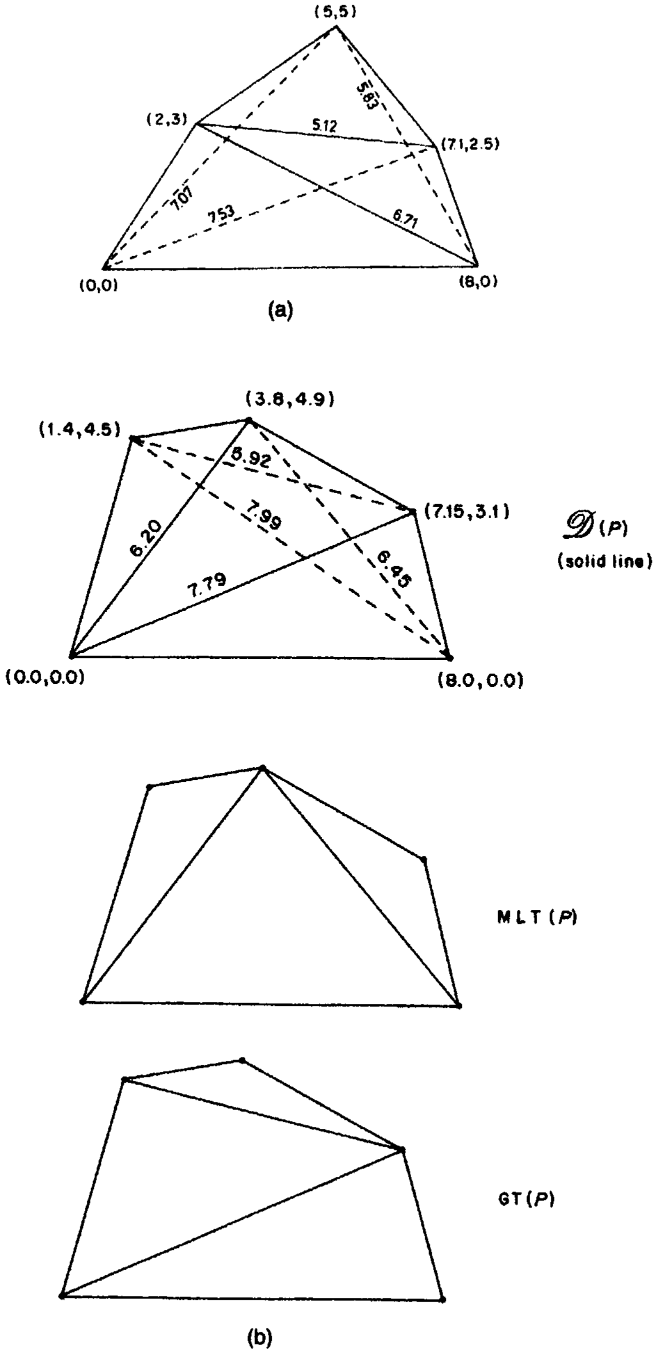


Figure 6.2.3 (a) A set of data sites in $\mathbb{R}^2$ for which $\mathcal{D}(P) = \text{MLT}(P) = \text{GT}(P)$ (solid line). (b) A set of data sites in $\mathbb{R}^2$ for which $\mathcal{D}(P) \neq \text{MLT}(P) \neq \text{GT}(P)$ (redrawn from material in Watson and Philip, 1984b).

$$E(|\mathcal{D}(P)|)/E(|\mathrm{MLT}(P)|) < 64/9\pi. \qquad (6.2.6)$$

$\mathcal{D}(P)$, MLT(P) and GT(P) are also proved to be good approximations of the complete graph spanning P in the sense that the length of the shortest path between any two vertices approximates the distance between the two points (Das and Joseph, 1989; Levcopoulos and Lingas, 1989; Keil and Gutwin, 1989; Dobkin *et al.*, 1990; see also Property D18 in Section 2.4).

Other triangulations have been suggested, but unlike $\mathcal{D}(P)$, MLT(P) and GT(P) these may be prone to problems such as lack of uniqueness and the creation of unnecessarily elongated triangles. Typically, these triangulations involve an iterative procedure in which first an initial triangulation is constructed and then this is 'optimized' according to specific criteria. Examples of such optimization criteria include (a) requiring that any location in a triangle must be closer to one of the vertices of that triangle than to any other vertex of the triangulation (i.e. the 'Pitteway' criterion discussed above) (McLain, 1976; Kleinstreuer and Holderman, 1980); (b) minimizing the length of the diagonal of the convex quadrilateral Q (Zienkiewicz and Phillips, 1971; Fraser and Van den Driessche, 1972; Fraser, 1977; Mirante and Weingarten, 1982); and (c) maximizing the minimum triangle height perpendicular to the proposed diagonal of Q (Gold *et al.*, 1977; Gold, 1980).

Another criterion for a good triangulation is related to the gradient of the interpolated surface. Let $\mathcal{T}$ be a triangulation for a given set P of sites, and for $p \in \mathrm{CH}(P)$ let $f(p)$ be the piecewise linear interpolant based on the triangulation $\mathcal{T}$, i.e. $f(p)$ be the height at p of the triangle determined by the given heights at the sites. The integration over CH(P) of the square of the gradient of the interpolant is called the *roughness* of the interpolant. Rippa (1990) showed that the Delaunay triangulation attains the minimum roughness for any set of height data. In this sense, the Delaunay triangulation gives the best interpolant. Wilson *et al.* (1990) observed a similar property.

However, from a perceptual point of view, the interpolant based on the Delaunay triangulation is not always the best one. Sometimes we get a better interpolation by swapping some of the edges of the Delaunay triangulation. Suppose that we are given a continuous surface over CH(P) and want to find the triangulation $\mathcal{T}$ for P such that the piecewise linear interpolating surface based on $\mathcal{T}$ is the closest to the given surface. One strategy to get this triangulation is to start with the Delaunay triangulation and to swap the edges in such a way that the resulting interpolating surface is closer to the given surface. Dyn *et al.* (1990), Brown (1991) and Schumaker (1993) showed that this strategy sometimes gives a better triangulation than the Delaunay triangulation.

Uniqueness and direct construction are particularly important to automated triangulation procedures. These characteristics of $\mathcal{D}(P)$ together with its equiangular property has led to it being proposed most frequently as a good triangulation for both spatial interpolation (Lewis and Robinson, 1977; Peucker *et al.*, 1978; Fowler and Little, 1979; McCullagh and Ross, 1980; De Floriani *et al.*, 1982, 1983, 1985; Philip and Watson, 1982; Watson and Philip, 1984b; Parker *et al.*, 1987; Jones 1989) and finite element methods (Bramble and Zlamal, 1970; Cavendish, 1974; Lawson, 1977; Akima, 1978; Joe, 1986), although in the latter area of application Babuska and Aziz (1976) argued that what is essential is not a minimum angle condition but a maximum angle (min-max) condition in which no angle is too close to 180°. Nielson and Franke (1983) showed that the min-max triangulation differs only slightly from the Delaunay triangulation, and Edelsbrunner *et al.* (1990b) have proposed an algorithm for generating the min-max triangulation.

6.3 MODIFYING DELAUNAY TRIANGULATIONS

One problem in using $\mathcal{T}(P)$ as the basis for interpolation over S is that $\mathcal{T}(P)$ is only defined in the interior of the convex hull CH(P) of P and therefore $\mathcal{T}(P)$ does not usually cover S. Furthermore, particularly for irregularly spaced data sites, $\mathcal{T}(P)$ invariably contains some undesirable long, thin triangles associated with points on the boundary of CH(P) regardless of the criterion used to define $\mathcal{T}(P)$. One way to extend a function defined over $\mathcal{T}(P)$ is to divide the area of S outside CH(P), S\CH(P), into trapezoids and triangles by inserting perpendiculars to the exterior edges of CH(P) at each of the vertices, as shown in Figure 6.3.1 (Akima, 1978; Ripley, 1981, pp. 42–43; Franke, 1982, p. 788). The value of the interpolant for any location in S\CH(P) is then obtained from the data values at one (triangular area) or two (trapezoidal area) nearest data sites. Strictly speaking such values should be considered extrapolations rather than interpolations.

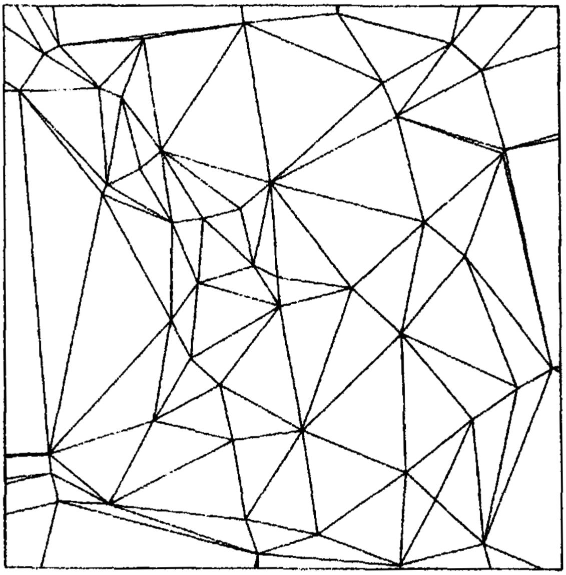


Figure 6.3.1 Extension of $\mathcal{D}(P)$ to cover S by constructing perpendiculars.

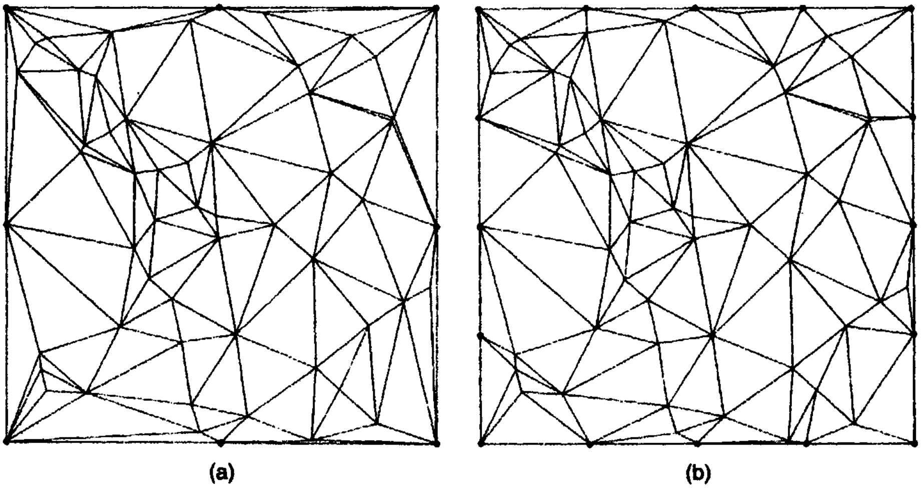


Figure 6.3.2 Extension of $\mathcal{D}(P)$ to cover S by introducing imaginary points (filled circles): (a) 8 imaginary points; (b) 16 imaginary points.

Another procedure which completes coverage of S and reduces the incidence of elongated triangles is to introduce imaginary data sites on the boundary of S (McCullagh and Ross, 1980; Franke, 1982; Lasser and Stuttgen, 1996) as indicated by the filled circles in Figure 6.3.2. The impact of this procedure on $\mathcal{T}(P)$ depends on the number and spacing of such imaginary data sites (compare Figures 6.3.2(a) and 6.3.2(b)).

So far in this chapter we have assumed that S is a simply connected, convex region. We now consider the construction of $\mathcal{D}(P)$ when S does not possess these properties as may be the case in some applications in digital terrain modelling (Yoeli, 1977; De Floriani *et al.*, 1982; Mirante and Weingarten, 1982) and finite element methods (Lewis and Robinson, 1977; Kleinstreuer and Holderman, 1980; Sadek, 1980) where non-convex and multiply connected regions may be encountered (for finite element methods, see Section 6.5). In such cases it is necessary to include edges of S as part of $\mathcal{T}(P)$. Another instance where specific edges must be incorporated into $\mathcal{T}(P)$ arises in digital terrain modelling when the data values refer to elevations. Although the data sites are usually irregular in terms of their location in the $x-y$ plane, often they are chosen to preserve both point features, such as pits and peaks, and linear features, such as ridges, valleys and fault lines, of special topographical significance (Peucker and Douglas, 1975; Yoeli, 1977; Fowler and Little, 1979; McCullagh, 1981; De Floriani *et al.*, 1985; Christensen, 1987; Buys *et al.*, 1991).

In all of the above instances we can consider the linear features as a set of non-intersecting edges L with end points in P, which together with P can be considered as a planar, straight line graph $G(P, L)$. Thus, the problem

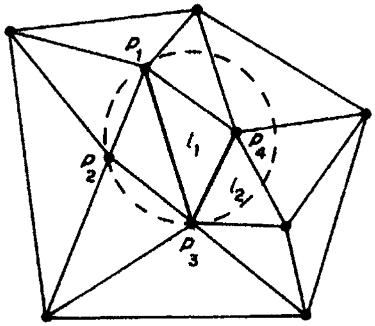


Figure 6.3.3 Illustration of criterion for defining edges in a constrained Delaunay triangulation.

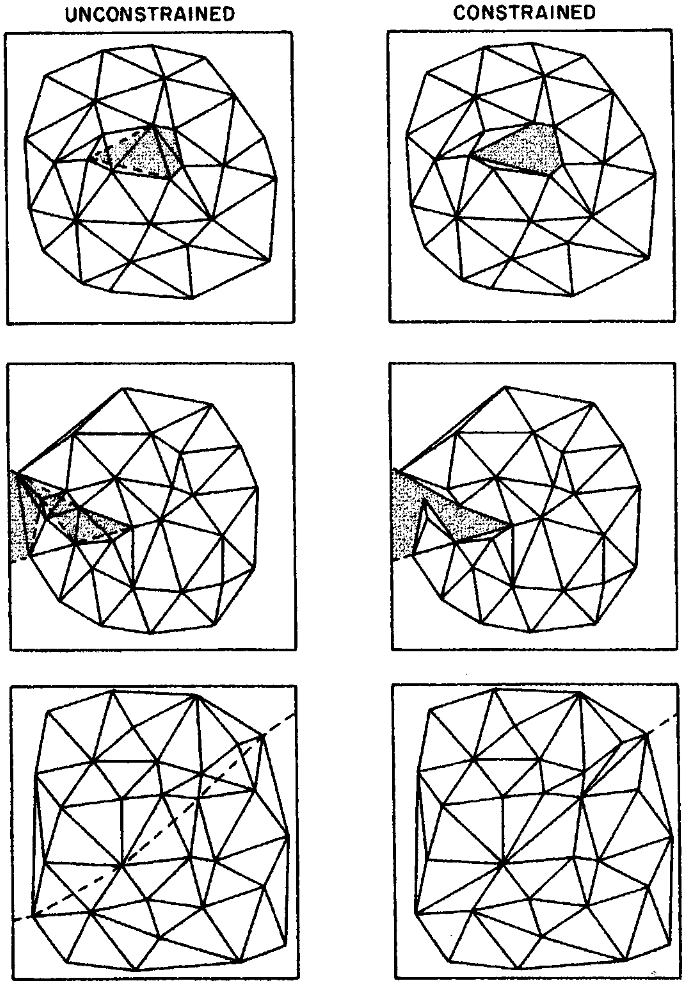


Figure 6.3.4 Examples of unconstrained and constrained triangulations for given sets of data sites. Bottom: - - - -, ridge line.

becomes one of constructing a triangulation $\mathcal{T}(G)$ for $G(P, L)$. Our concern is extending the Delaunay criteria (see Section 6.2) to $\mathcal{T}(G)$ and we refer to the resulting triangulation as the *constrained Delaunay triangulation*, CDT(G), of $G(P, L)$ (recall Section 3.4.3; see also Section 6.5).

The modifications of $\mathcal{D}(P)$ which must be undertaken to create CDT(G) arise because the presence of L in S affects the mutual visibility of members of P. Two sites in P, p_i and p_j, are said to be mutually visible if the line segment $\overline{p_i p_j}$ does not properly intersect any member of L. For example, in Figure 6.3.3 points p_1 and p_2 are mutually visible while points p_2 and p_4 are not. Let V be the set of edges joining those data sites which are mutually visible. Then CDT(G) consists of L together with a subset of V in which the circumcentre of each triangle with vertices p_i, p_j and p_k does not contain in its interior any other data site which is visible from p_i, p_j and p_k. Thus, points p_1, p_2 and p_3 in Figure 6.3.3 form a triangle of CDT(G) even though the circle passing through them contains p_4 because p_2 and p_4 are not mutually visible. If no four points of P are cocircular, CDT(G) is unique. CDT(G) is a subgraph of the visibility graph of $G(P, L)$ (see Section 3.4.1). For graphs $G(P, \varnothing)$, CDT(G) is equal to $\mathcal{D}(P)$. Figure 6.3.4 shows examples of CDT(G) together with their unconstrained counterparts $\mathcal{D}(P)$.

Like $\mathcal{D}$(P),CDT(G) satisfies the local equiangular criterion since it applies to any strictly convex quadrilateral of $\mathcal{T}(G)$ and only one diagonal can be selected if a quadrilateral of $\mathcal{T}(G)$ is not convex or if one of the diagonals is a member of L. CDT(G) also satisfies global equiangularity (see Lee and Lin, 1986, for a proof). Lee and Lin (1986) provide an O(n^2) algorithm for computing CDT(G) which reduces to O($n \log n$) when $G(P, L)$ is a simple polygon.

6.4 APPROXIMATING SURFACES

There are some applications where the number of data sites P may be extremely large. One instance is digital terrain modelling where the data sites may be in the form of a dense raster (grid) structure, particularly if they have been obtained by remote sensing (Fowler and Little, 1979; De Floriani *et al.*, 1983, 1985; Chen and Guevara, 1987). To capture the fundamental features of the surface being modelled while avoiding subsequent unnecessary time-consuming search and retrieval operations during its analysis and inefficiencies in its storage, it may be undesirable to use $\mathcal{T}(P)$ in deriving the function to represent the surface. Instead, we can select a representative subset P' of data sites in P which enables the surface to be represented with less than some predetermined level of error. P' should be generated so that only those members of P necessary to reduce the error level below the specified level are selected. Typically, this is achieved by an incremental procedure in which an initial set P' is defined and $\mathcal{T}(P')$ is generated. Further data sites are added to P' until the surface generated using $\mathcal{T}(P')$ no longer exceeds the specified error level. Such an incremental procedure is facilitated

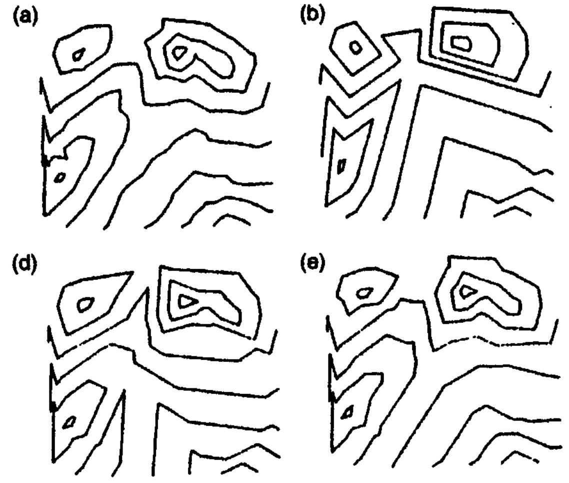


Figure 6.4.1 Estimating surfaces: (a) original surface; (b) 50% of error, obtained by selecting 20% of data sites; (c) 30% of error, obtained by selecting 30% of data sites; (d) 20% of error, obtained by selecting 50% of data sites. (*Source*: De Floriani *et al.*, 1985.)

by using the Delaunay triangulation $\mathscr{D}(P')$ to define $\mathscr{T}(P')$ since the local optimization character of $\mathscr{D}(P')$ ensures that an existing triangle is only affected if the new data site is added within the circumcircle defined by the vertices of the existing triangle.

A typical procedure is that used by De Floriani *et al.* (1983, 1985). The initial set P' is chosen so that it contains only those data sites necessary to define the boundary of the region under consideration, which may be the boundary of the convex hull CH(P) of P or a modified form obtained using the procedures defined in Section 6.3. $\mathscr{D}(P')$ is then constructed. For each data site p_i'' in the set P'' ($P'' = P\backslash P'$) of the data sites not in P', we can define an associated error value $e(p_i'')$ as

$$e(p''_i) = |z''_i - f(p''_i)| / z''_i \tag{6.4.1}$$

where z_i'' is the data value at p_i'' and $f(p_i'')$ is the interpolated value (obtained using a specified interpolation procedure) at p_i'' using $\mathscr{D}(P')$. The error associated with $\mathscr{D}(P')$, e[$\mathscr{D}(P')$], is defined as

$$e\left[\mathscr{D}(P')\right] = \max_{p''_i \in P''} e(p''_i) \tag{6.4.2}$$

and can be used as a measure of the error. If $e[\mathscr{D}(P')]$ exceeds the predetermined error level the data site p_i'' ($p_i'' \in P'$) for which $e(p_i'') = e[\mathscr{D}(P')]$ is added to P' and $\mathscr{D}(P')$ is updated. The processes are repeated until $e[\mathscr{D}(P')]$ no longer exceeds the specified error level. Figure 6.4.1 shows a given surface at three different error levels when a C^0 interpolant is used in equation (6.2.5). Note that this procedure may be suboptimal since P' need not be the minimal subset of *P* necessary to produce a surface with a specified error level.

Falcidieno and Pienovi (1990), Rippa (1992) and Fjällströn (1993) proposed similar data-selection strategies in the context of the data compression. Park and Kim (1995) also used the same strategy for the concise and fine representation of solid models using cubic triangular Bézier surfaces. A similar progressive refinement is also used to decrease the aliasing effect in computer graphics (Painter and Sloan, 1989).

De Floriani (1989a), De Floriani and Puppo (1992a) and De Floriani *et al.* (1996) proposed hierarchical data structures to represent a triangle-based surface in different resolutions, and De Floriani *et al.* (1993) constructed an algorithm for extracting the contour from these data structures.

6.5 DELAUNAY MESHES FOR FINITE ELEMENT METHODS

Finite element methods are used widely in modelling flow, diffusion, convection, heat transfer, mechanical oscillation and similar processes. Those methods can be considered as a kind of spatial interpolation since the solution of a partial differential equation is approximated by the values at discrete mesh sites or by the weights of trial functions defined locally on the mesh elements (Ho-Le, 1988; Schwarz, 1988; George, 1991; Bern and Eppstein, 1992b; Field, 1995; Babuska *et al.*, 1995; Klein and Cohen-Or, 1997).

The meshes used in the finite element methods are divided into two classes. One class consists of meshes composed of simplices (i.e. triangles in the two-dimensional space and tetrahedra in the three-dimensional space) while the other class is composed of non-simplex elements (such as quadrilaterals and hexahedra). One promising tool for the former class of meshes is the Delaunay tessellation, which we call the *Delaunay mesh*, and which we concentrate on in this section. However, we note that the medial axis and the medial surface (see Section 3.5.4) are sometimes used for the latter class (Tam and Armstrong, 1991; Srinivasan *et al.*, 1992; Price *et al.*, 1995; Price and Armstrong, 1997). For more information on this class, see Joe (1995b) for example. There is also another type of discretization of partial differential equations called finite volume methods, for which Voronoi diagrams are often used (Ghosh and Mukhopadhyay, 1993; Ghosh and Liu, 1995; Ghosh and Moorthy, 1995; Moorthy and Ghosh, 1996; Ball, 1996; Lee and Ghosh, 1996).

In this section we study mesh generation in two and three dimensions separately. While two-dimensional meshes can be generated relatively easily, there are difficult problems associated with three-dimensional meshes.

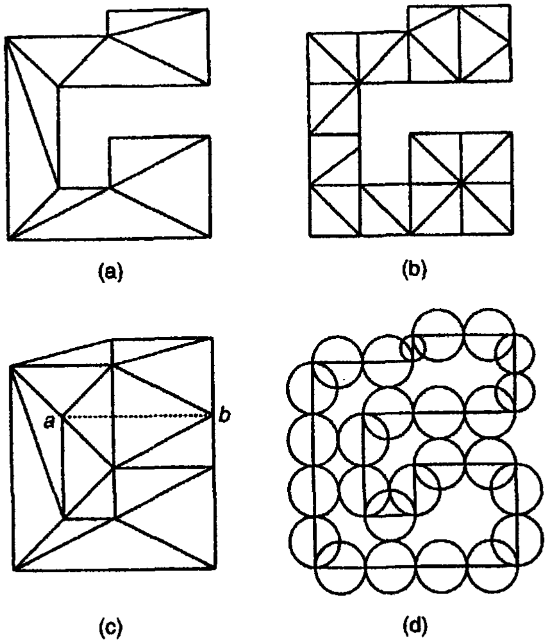


Figure 6.5.1 Triangulation inside a polygonal domain: (a) constrained Delaunay triangulation; (b) triangulation using additional points; (c) Delaunay triangulation for the same set of generators as in (a); (d) placement of additional points which guarantee the consistency of the Delaunay triangulation with the boundary of the domain.

6.5.1 Two-dimensional Delaunay meshes

Suppose that we are given a partial differential equation in a polygonal region S, and want to generate a triangular tessellation of S to solve the equation. According to the convention in finite element methods, we call this tessellation a *mesh* over S. Mesh generation differs from other Delaunay interpolations in two ways.

First, the set P of generators is not fixed. P should contain at least the vertices of the polygon S, but the number of other generators and their locations are not given a priori. We can use this freedom to get good meshes, to control mesh density, and for other purposes. For example, Figures 6.5.1(a) and (b) show two different meshes over the same polygonal region S. In (a) only the vertices of the polygon S are used, whereas in (b) many additional generators are used. Second, the mesh should be consistent with the boundary of S; that is, the whole of the boundary of S should be realized as the edges of the triangles. This constraint makes the mesh generation a little difficult.

One strategy to generate the mesh may be first to compute the Delaunay triangulation and then to remove the triangles outside S. However, this strategy does not necessarily work. For example, if we compute the Delaunay triangulation for the vertices of S in Figure 6.5.1(a), we get the triangulation as shown in (c). Note that in (c) the boundary edge *ab* is not realized as an edge of a triangle, and hence we cannot obtain a consistent mesh by this strategy.

There are two typical strategies to overcome this difficulty. One is to utilize the constrained Delaunay triangulation (Chew, 1989a) (see Section 3.4.3 and Section 6.3). This is also called the *domain Delaunay triangulation* (Sapidis and Perucchio, 1990). The constrained Delaunay triangulation can be constructed in $O(n \log n)$ time, where n is the number of vertices of S (Chew, 1989a; Moreau and Volino, 1993). Actually, the mesh in Figure 6.5.1(a) is obtained first by generating the constrained Delaunay triangulation with the boundary edges in S as the constraints, and then by removing the triangles outside S.

The other strategy is to place additional vertices so that the associated Delaunay triangulation realizes all the boundary of S. As shown in Figure 6.5.1(d), suppose that we place sufficiently many additional generators on the boundary of S in such a way that each circle having the consecutive generators as the diameter does not contain other generators. Then, the Delaunay triangulation of such generators realizes the whole boundary by the Delaunay edges (recall Property D5 in Section 2.4, i.e. the empty circle property). Such Delaunay triangulations are called *conforming Delaunay triangulations* (Edelsbrunner and Tan, 1993).

Edelsbrunner and Tan (1993) showed that this strategy always works by placing at most $O(m^2n)$ additional generators, where n is the number of vertices and m is the number of constraint edges. However, $O(m^2n)$ is for the worst case; a smaller number of additional generators is sufficient in many cases. Methods for the incremental addition of new points for this purpose were studied by Boissonnat (1988), Schroeder and Shephard (1988) and Sapidis and Perucchio (1991b).

Because of the lexicographical maximality of the small angles (recall Property D16 in Section 2.4), the Delaunay triangulation gives a good-quality mesh for the given set of generators. However, we have freedom in choosing the locations of the additional generators. This freedom enables us to get a better triangulation.

Figure 6.5.2(a) shows the mesh generated by the Delaunay triangulation. After generating this triangulation, we move the generators in the interior of the region to the centre of gravity of the adjacent generators simultaneously (this operation is called 'Laplacian smoothing' (Srinivasan *et al.*, 1992)), obtaining the triangulation shown in (b). This triangulation is better because their are fewer thin triangles than in (a).

Another strategy to improve the quality of a mesh is to insert additional generators. Chew (1989a), Dey (1990) and Dey *et al.* (1991, 1992a) recommended inserting new generators at the circumcentres of bad-quality triangles; see also Rebay (1993).

Mavriplis (1990) and Posenau (1993) pointed out that sometimes thin triangles, i.e. thin in some specific direction, are required rather than fat triangles, and Posenau (1993) showed that this requirement can be fulfilled using the

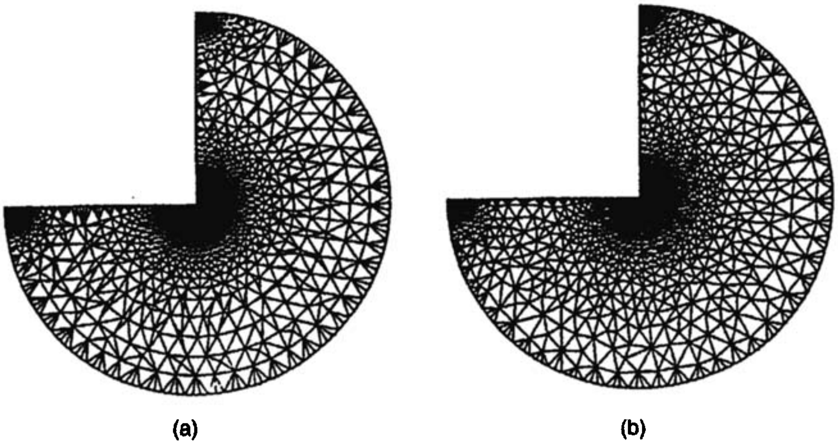


Figure 6.5.2 Mesh improvement with the displacement of points: (a) Delaunay mesh; (b) Laplacian smoothing.

Delaunay triangulation with respect to the elliptic distance. Srinivasan *et al.* (1992) used the Voronoi diagram of a polygon, particularly the medial axis, for placing vertices for both the triangular mesh and the quadrilateral mesh. Goodness of the mesh also depends on the partial differential equation to be solved. Hence, the *solution-adaptive mesh* is often used; that is, we first solve the equation using an initial mesh, and next modify the mesh adaptively in order to decrease the errors in the solution (Weatherill *et al.*, 1994, 1995; Marcum and Weatherill, 1995a,b; Medina *et al.*, 1996). For other two-dimensional, good-quality meshes, refer to Cendes *et al.* (1983), Lo (1985, 1989) and Baker *et al.* (1988), Bern *et al.* (1990), Cline and Renka (1990), Bern and Eppstein (1991, 1992a), Reichert *et al.* (1991), Melissaratos and Souvaine (1992), Weatherill (1992), Bern *et al.* (1995a,b), Borouchaki and George (1997) and Karamete *et al.* (1997).

6.5.2 Three-dimensional Delaunay meshes

Three-dimensional mesh generation is much more complicated than in two dimensions. One reason is that for a given polyhedral region S bounded by triangular faces and a given set P of generators, there might be no tetrahedral tessellation that satisfies the constraint.

For example, consider the polyhedron shown in Figure 6.5.3. This is a triangular prism whose boundary surface is partioned into triangles by three vertex-disjoint diagonal edges. Suppose that the given generators are only the vertices of this polyhedron. Then, there is no tetrahedral tessellation, because any new line segment connecting two vertices lies outside the polyhedron.

It is still an open problem to judge in polynomial time whether a given constraint (specified by the region S and the generator set P) admits a tetra-

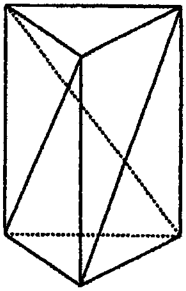


Figure 6.5.3 Polyhedral domain that cannot be partitioned into tetrahedra unless additional points are used.

hedral tessellation. Thus, no algorithm is known, for a given polyhedral region *S* bounded by triangular faces and a given set *P* of generators, to generate the constrained Delaunay tetrahedrization or to tell that it does not exist in O(n) time. Hence, what we can do is to insert new generators in such a way that the augmented set of generators hopefully admits the consistent tetrahedral tessellation.

For the region in Figure 6.5.3, let us insert one new generator in the interior in such a way that all the corners of this polyhedron are visible from the new generator. Then, the region can be partitioned into tetrahedra, i.e. into triangular cones with the new generator as the apex and the triangular faces as the base triangles.

A typical strategy is to place sufficiently many additional generators on the boundary surface of *S* so that the resulting ordinary Delaunay tessellation realizes the whole boundary surface. To guarantee this property, we search for the placement of sufficiently many new generators on the faces of the boundary so that every Delaunay triangle in the two-dimensional Delaunay triangulation generated by the generators on each face admits a circumsphere that contains no other generators. If such a placement of new generators is found, then the ordinary Delaunay tetrahedrization realizes all the boundary faces, and consequently all we have to do is to remove the tetrahedra outside *S*. Methods for placing such new generators were studied by Dey *et al.* (1991, 1992a).

Another strategy is first to generate the Delaunay mesh for the set of generators (this is in general not consistent with the boundary of *S*), and then insert new generators one by one in order to eliminate tetrahedra that penetrate the boundary surface of the region *S*. This kind of incremental insertion was studied by George *et al.* (1990, 1991).

Another difficulty in three-dimensional mesh generation is the fact that Delaunay tetrahedra do not necessarily have good shape from the viewpoint of the finite element methods (Dey *et al.*, 1991, 1992a; Escobar and Montenegro, 1996). In two dimensions the goodness of the Delaunay triangles is supported by the max-min angle property (refer to Properties D15 and D16 in Section 2.4). In some sense, similar equiangularity properties also hold in three dimensions (and in higher dimensions) as shown by Rajan

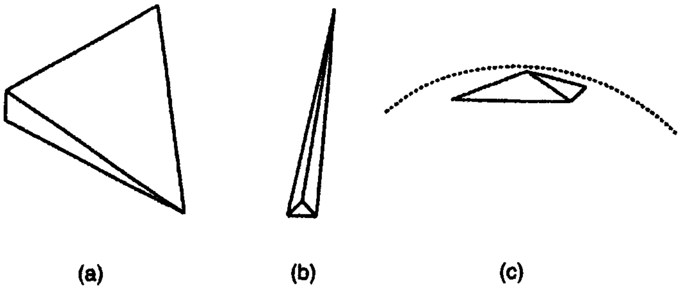


Figure 6.5.4 Bad tetrahedra that can be avoided by the Delaunay triangulation.

(1991, 1994) and Schmitt and Spehner (1993). However, there are different types of angles associated with a tetrahedron: angles of each face triangle, angles formed by two adjacent faces (which are called the *dihedral angles*), and solid angles at the vertices. In particular, the dihedral angles do not admit an equiangularity-type property. Indeed, the Delaunay tetrahedrization often contains smaller dihedral angles than other tetrahedrizations for the same set of generators.

Suppose that a tetrahedron t has the radius R of the circumscribing sphere, the longest edge length L and the shortest edge length l. According to Dey *et al.* (1992a), let us define ω and κ by $\omega = 2R/L$ and $\kappa = L/l$. Note that for any tetrahedron $1 \leq \omega$ and $1 \leq \kappa$. If κ is very large, the tetrahedron has either one short edge as in Figure 6.5.4(a) or three short edges as in Figure 6.5.4(b). They are bad tetrahedra, and hence a smaller κ is desirable. If ω is very large, the tetrahedron is very small compared with the circumsphere, as shown in Figure 6.5.4(c). This tetrahedron is also unsuitable for interpolation, and hence a smaller ω is desired. Recall that the circumsphere of every Delaunay tetrahedron is empty. A tetrahedron with large κ or large ω has a large circumsphere relative to the shortest edge length. As the sphere becomes larger, it is in general more likely to contain other generators. Therefore we can expect that the Delaunay mesh can avoid tetrahedra with large κ or large ω. However, even if both ω and κ are small, there may still be bad tetrahedra, as shown in Figure 6.5.5; this diagram consists of the pair of the top view and the front view. In this tetrahedron two dihedral angles are very large (nearly π) and the other four are very small. Such thin tetrahedra cause large numerical errors in finite element methods. However, tetrahedra of this type often appear in the Delaunay mesh because their circumspheres are not very large.

Thus, the Delaunay mesh is not always satisfactory; it may contain undesired slivers. Hence, some postprocessing is necessary to improve the quality of the mesh. Typical postprocessings proposed so far are based on heuristic local refinements, and there are three approaches.

The first approach is the local refinement using 'flips'. Suppose that, as shown in Figure 6.5.6(a), two tetrahedra share a triangular face $v_1v_2v_3$, and their union forms a convex hexahedron. Let the other vertices be v_4 and v_5,

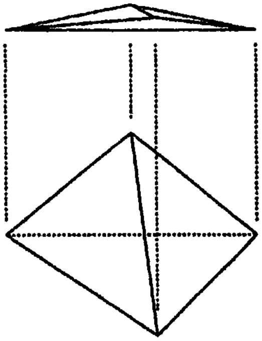


Figure 6.5.5 Bad tetrahedron with small ω and κ.

respectively. If we remove the face $v_1v_2v_3$ and add three faces $v_1v_4v_5$, $v_2v_4v_5$ and $v_3v_4v_5$, then this hexahedron is partitioned into three tetrahedra as shown in Figure 6.5.6(b). The change of the tetrahedrization from (a) to (b) or from (b) to (a) is called a *flip*. The flip operation is a three-dimensional counterpart of the swap operation for the two-dimensional triangulation.

If we have some criterion to judge whether a flip improves the quality of a mesh, then we can construct a local refinement procedure. An example of such a criterion is to maximize the minimum solid angle of the associated two or three tetrahedra; that is, we apply a flip operation if it makes the minimum solid angle larger (Joe, 1989, 1991b). Another criterion is to maximize the ratio of the radius of the inscribed sphere to that of the circumsphere (Joe, 1995a). Other criteria and a combination of two or more criteria are also proposed (Joe, 1995a). Nehl and Field (1991) and Zhu and Zienkiewicz (1997) proposed the adaptive subdivision of the mesh according to the errors in the finite element method. Kanaganathan and Goldstein (1991) showed experimentally that the Delaunay tetrahedrization still gives a better mesh than many other methods. If we use the empty-circumsphere criterion, then we can change any tetrahedral mesh to the Delaunay mesh by a sequence of flip operations (Joe, 1991a).

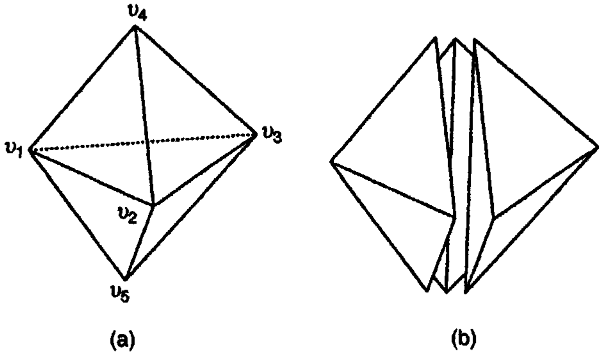


Figure 6.5.6 Flip operation for a hexahedron: (a) decomposition into two tetrahedra; (b) decomposition into three tetrahedra.

The second local refinement approach is the relocation of generators. By moving the vertices of the slivers or nearby vertices, we can improve the mesh quality (Cavendish *et al.*, 1985; George, 1997).

The third local-refinement approach is the insertion of new generators. Proposed heuristics include the insertion of the midpoints of edges (Liu and Joe, 1995), the insertion of new generators near the slivers (Baker, 1989), and the insertion of new generators at the centres of the circumspheres of the slivers (Yuen *et al.*, 1991). For other heuristics, see Morgan *et al.* (1991), Weatherill and Hassan (1992, 1994) and Yagawa *et al.* (1995).

Besides the Delaunay-mesh approaches, there are many other methods for quality-guaranteed mesh generation. For details, refer to the surveys by George (1991), Bern and Eppstein (1992b), Field (1995) and Babuska *et al.* (1995). The Delaunay mesh is also used in a domain including the time axis (Morgan *et al.*, 1994, 1996).

6.6 ORDERING MULTIVARIATE DATA

Unlike univariate data there is no unique way of ordering a set of multivariate data. However, if the individual cases are represented as a point set P in $\mathbb{R}^m$, the creation of either the Voronoi diagram, $\mathcal{V}(P)$, or the Delaunay tessellation, $\mathcal{D}(P)$, imposes a spatial structure on P. Thus, $\mathcal{V}(P)$ and $\mathcal{D}(P)$ are equivalent to a spatial ordering of the elements of P (Watson, 1985; Gold and Cormack, 1987). In this section we consider a number of situations where such ordering has been exploited.

When we are working with $\mathcal{D}(P)$, as in the digital terrain models discussed in Section 6.2, ordering the constituent triangles allows them to be processed incrementally, thus facilitating such tasks as isoline extraction, creating perspective views, and determining the visibility regions of arbitrary vertices. Gold and Maydell (1978), Gold (1987) and Gold and Cormack (1987) show that any triangulation may be represented as an ordered binary tree with respect to some viewpoint, thus permitting front-to-back or radially-outwards processing of the triangles and their associated data points. If the triangulation is a Delaunay triangulation, De Floriani (1989b) and De Floriani *et al.* (1991) show that it is possible to order the triangles with respect to any vertex in such a way that a ray from the vertex to an arbitrary target point inside any one of the triangles passes through only triangles which rank lower in the ordering than the triangle containing the target point. For example, in Figure 6.6.1 the ray from vertex v to target point t in triangle 10 passes in order through triangles 6 and 9. When the triangles are ordered in this way, each region R_k consisting of the union of triangles $1,2,\ldots,k$ is star shaped with respect to the vertex under consideration (see Figure 6.6.1). While this is a property of any Delaunay triangulation (see Property D19 in Section 2.4) it is not possessed by all planar triangulations. This ordering scheme has been generalized to $\mathbb{R}^m$ for cell complexes whose faces are

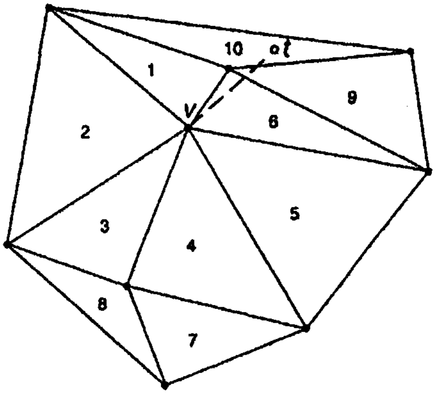


Figure 6.6.1 Ordering of Delaunay triangles with respect to vertex *v*.

obtained by orthogonal projections of the faces of a convex polytope in $m+1$ dimensions (Edelsbrunner, 1990).

The same spatial relations which give rise to the Voronoi diagram and the Delaunay tessellation can also be represented by the set of all empty (natural neighbour) spheres which define the Delaunay tessellation (see Figure 2.4.3). These natural neighbour spheres may be organized hierarchically to form a *natural tree* in $\mathbb{R}^2$ (Watson and Mees, 1996) or a *Delaunay tree* in $\mathbb{R}^3$ (Boissonnat and Teillaud, 1986) which corresponds to an incremental construction of the Delaunay tessellation (see Figure 6.6.2). Thus, the nodes of the tree are composites made up of groups of associated natural neighbour spheres and the clusters of data points which define them. As well

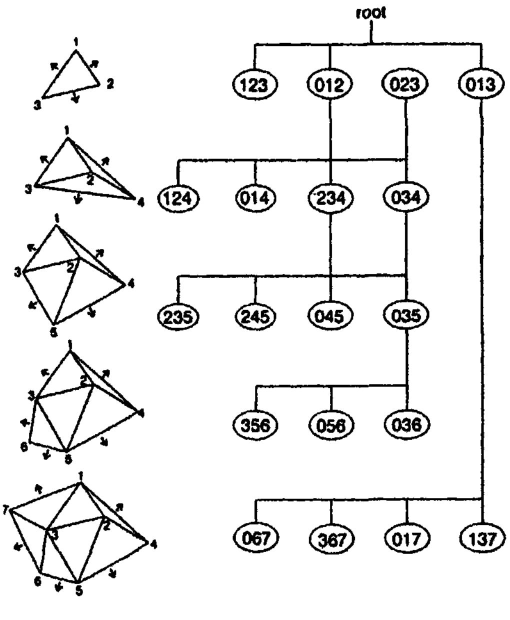


Figure 6.6.2 A Delaunay triangulation and its natural tree.

as providing an efficient means of generating Delaunay tessellations, especially when they are used for the study of dynamic systems in higher dimensions (Mees, 1991), natural trees have a wide range of applications in problems such as density estimation, interpolation, gradient estimation and classification.

Another means of organizing the data hierarchically is proposed by Okabe and Sadahiro (1996) who consider the situation in which each point p_i in P has an associated attribute value a_i measured on at least an ordinal scale (see Figure 6.6.3(a)). They begin by finding first-level local centres which are defined as those points whose attribute values are greater than or equal to

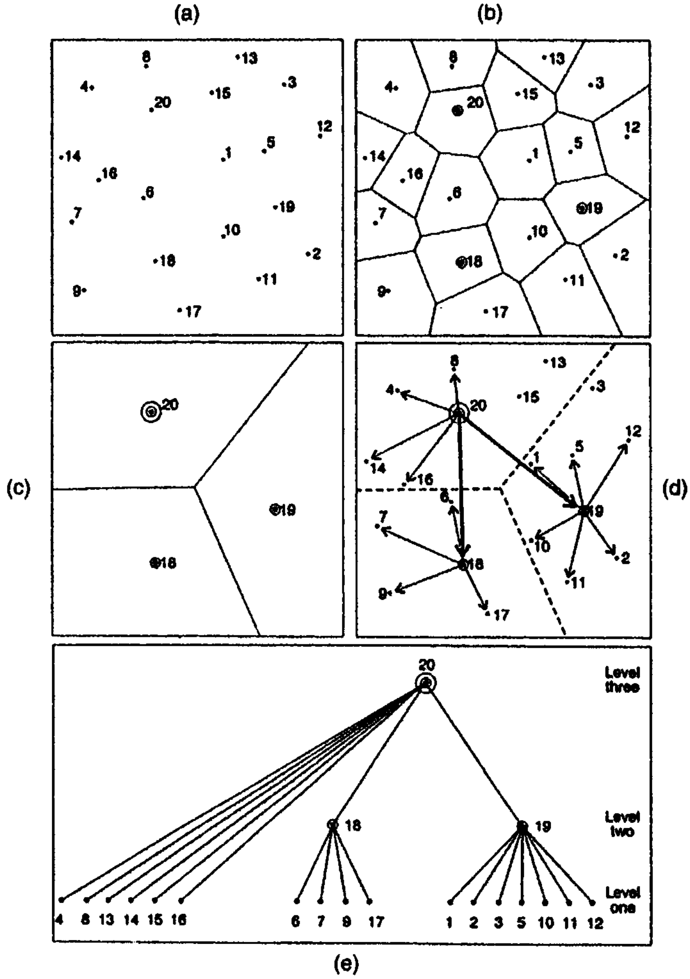


Figure 6.6.3 Hierarchical ordering of a set of points: (a) points and associated attribute values; (b) definition of first-level local centres; (c) definition of second-level local centres; (d) directed tree graph of relationships between points; (e) hierarchical order.

those of their Voronoi neighbours (see Figure 6.6.3(b)). In the same way, the Voronoi diagram of the first-level local centres is used to define second-level local centres (see Figure 6.6.3(c)). This procedure is repeated, with the Voronoi diagram of the kth-level local centres being used to define the $(k + 1)$th-level local centres, until only one point is left. Finally, the hierarchical relationships between the points can be summarized by a directed tree graph (see Figures 6.6.3(d) and 6.6.3(e)) by determining for each kth-level local centre the $(k + 1)$th level Voronoi polygon which contains it. Okabe and Sadahiro use this procedure to create a spatial hierarchy of 100 market-places in a 6 km-square residential area in Nishinomiya near Osaka.

Some problems may require considering more than one spatial structure simultaneously. For instance, the integration of remotely sensed data with cartographic data is usually achieved by transforming both the remotely sensed image and the cartographic data to a common base. Converting the remotely sensed data may require changing the origin, scale and orientation of the image, as well as correcting for any patterns of distortion in the data. This is done by collecting a set of *ground control points* (GCPs) at locations which are known to be coincident on both the image and the map (e.g. road intersections). The location of every GCP on both the image ($\boldsymbol{x}_i'$) and the map ($\boldsymbol{x}_i$) is then recorded and used to generate a polynomial which models the relationship between $\boldsymbol{x}_i'$ and $\boldsymbol{x}_i$. The polynomial can then be used to predict $\boldsymbol{x}_i'$ for any $\boldsymbol{x}_i$. As an alternative to this procedure, Devereux *et al.* (1990) suggest simultaneously constructing the Delaunay triangulation of the GCPs on both the image $\mathcal{D}(I)$ and the map $\mathcal{D}(M)$. Then, within each triangle in $\mathcal{D}(I)$ and $\mathcal{D}(M)$, generate a new GCP using the point of intersection formed by two lines which bisect any two triangle edges and their opposite angles. Add these secondary GCPs to the original GCPs and recompute $\mathcal{D}(I)$ and $\mathcal{D}(M)$. Examine the sizes of the triangles in $\mathcal{D}(I)$. If a triangle is less than some pre-specified threshold size, then it is no longer a source for new GCPs; if not, generate another new GCP in the above manner. The procedure terminates when every triangle in $\mathcal{D}(I)$ is less than the threshold size. While the final $\mathcal{D}(I)$ and $\mathcal{D}(M)$ will have the same topology, they will only have the same geometry if there is no distortion in the image. Distortion in the image can be corrected by matching the triangles of $\mathcal{D}(I)$ and $\mathcal{D}(M)$. Label the vertices of a triangle in $\mathcal{D}(M)$, and its corresponding triangle in $\mathcal{D}(I)$, ABC and $A'B'C'$, respectively. Since any two sides of the triangle ABC (say the shortest two, AB, AC) are linearly independent vectors, any pixel X inside triangle ABC can be linked to vertex A by vector $\overrightarrow{AX}$ which can be expressed as a linear combination

$$\overrightarrow{AX} = \alpha\overrightarrow{AB} + \beta\overrightarrow{AC} \quad (0 \leq \alpha \leq 1;\ 0 \leq \beta \leq 1 - \alpha). \tag{6.6.1}$$

Equation (6.6.1) can be solved by two simultaneous equations and the values of the coefficients inserted into

$$\overrightarrow{A'X'} = \alpha\overrightarrow{A'B'} + \beta\overrightarrow{A'C'} \tag{6.6.2}$$

to find the image pixel location X'. Once found, the relevant pixel can be retrieved from the image and placed in the corrected output image at X. In studying the more general problem of matching a set of data points and a set of model points in the plane, Ogawa (1986), Finch and Hancock (1995), and Finch *et al.* (1997) generate the Delaunay triangulation of one of the point sets to provide a point–edge–face structure which is then checked for consistency with three-point subsets of the other point set.

Another kind of ordering arises in some multivariate two-sample tests. Suppose that we have two m-dimensional samples $X = \{x_1, x_2, \ldots, x_n\}$ and $Y = \{y_1, y_2, \ldots, y_m\}$, in each of which the points are independent and identically distributed and, without making any distributional assumptions, we wish to determine if the populations from which the samples are drawn are the same in some way. Friedman and Rafsky (1979, 1981) suggest defining a proximity graph G (see Section 2.5) on the pooled sample and examining the number of edges L in the graph which have one endpoint in each of the samples. When L is small, points in each sample tend to have neighbours in the same sample suggesting that the two samples are drawn from different populations. If the two samples are drawn from the same population, the mean of L is

$$\mathrm{E}(L \mid T) = gT \tag{6.6.3}$$

and the variance is

$$\mathrm{Var}\,(L \mid C, T) = g\left\{T(1 - gT)T + C + \frac{2(m-1)\,(n-1)}{(N-2)\,(N-3)}\left[T(T-1) - 2C\right]\right\} \tag{6.6.4}$$

where $N = n + m$ is the number of points in the pooled sample, T is the number of edges of G, C is the number of edge pairs which share a common vertex, and $g = 2mn/N(N - 1)$. Friedman and Rafsky use the Euclidean minimum spanning tree (EMST) to define G and demonstrate that in this case the procedure is equivalent to a multivariate generalization of the Wald–Wolfowitz runs test. McIntosh (1988) uses the Delaunay tessellation instead of the EMST. He shows that while the test using L performs well when the populations being compared differ only in location, it performs poorly when the populations differ only in scale. In such situations, he proposes an alternative procedure in which each point in the pooled sample is assigned a rank according to the number of Delaunay tessellation edges that must be traversed to reach the point from the boundary of the convex hull ∂CH of the pooled sample. Ties in rankings are broken using the Euclidean distance from each point to the centroid of the points that are farthest from ∂CH, with points which are closer to the centroid receiving higher ranks. The ranks can then be assessed by means of either the standard Smirnov or Wilcoxon tests. Simulations show that when the underlying distributions are normal, the tests involving the Delaunay triangulation perform as well as or better than traditional tests.

The ranking procedure just described is related to *peeling* the data set. This is a form of data ordering which is useful in such procedures as the

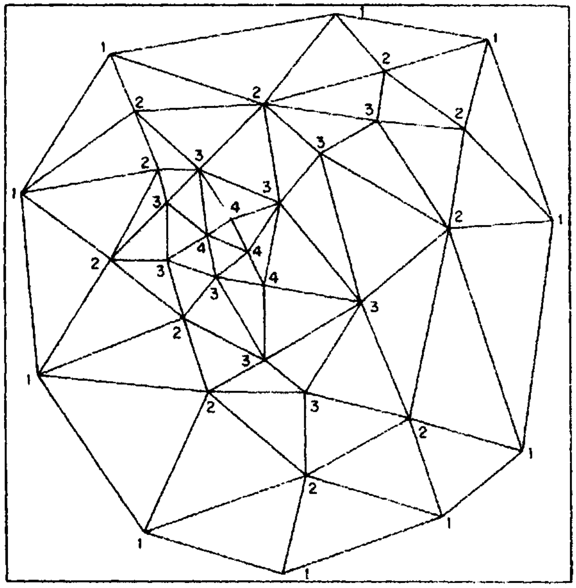


Figure 6.6.4 Peeling according to adjacencies defined by the Delaunay triangulation (numbers indicate the numbers of specific peels).

identification and treatment of outliers in P, establishing significance levels for Monte Carlo tests, estimating correlation and deriving measures of location (Green, 1981). In general, peeling involves an iterative process which begins by finding the smallest region, R, of a given form which contains P. The points on the boundary of R are assigned the index 1. These points are then discarded and R is redefined for the remaining points. This process is repeated until no points are left. Those points which are discarded on the jth iteration form the jth peel. Various suggestions have been proposed for the form of R including, in the bivariate case, rectangles, circles and ellipses. The first two take no regard of the joint distribution of the sample and may thus fail to identify outliers in strongly correlated data. The last two forms create peels that may not be nested spatially. Consequently, other forms of R have been proposed. One is the boundary of the convex hull of P, $\partial\mathrm{CH}(P)$, (Bartlett, 1976) which as we saw in Section 2.3 is a subgraph of $\mathcal{D}(P)$. Another peel (Green, 1981) can be defined by the neighbour relationships in $\mathcal{V}(P)$ or $\mathcal{D}(P)$ as shown in Figure 6.6.4. Here those points on the boundary of $\partial\mathrm{CH}(P)$ have the index 1 while their Delaunay neighbours have the index 2. The neighbours of these latter points are, in turn, indexed 3, and so on. So far there has been no analysis of the relative merits of the different peels in the various applications of data ordering mentioned above.