Introduction

The purpose of this article is to provide a review of traditional interpolating techniques, together with their applications and weakness, and to discuss a 2-D analogue of 1-D interpolation that eliminates most of the problems and opens up the possibility of defining 2-D interpolation to match specific user requirements.

As discussed in Gold [1984], ‘contouring’. techniques have frequently confused issues of database storage and retrieval, sub-sampling site selection, neighbouring point selection, surface estimation procedures, and display issues. Due to a preoccupation with ‘technique’, for which any individual example may be more or less appropriate, the underlying objectives of the problem to be solved are often overlooked. In particular, the user frequently is of the impression that his data is precise, relatively sparse across the map sheet, and arbitrarily (and anisotropically) distributed — and that the program will adequately service these assumptions, providing an interpolated surface that precisely honours all data point elevations. The programmer, however, is often assuming (or hoping) that the data will be relatively evenly distributed, that a ‘reasonable’ density of data is available, and that ‘minor’ discrepancies between observations and the interpolated surface will not be significant. While this is certainly not the only scenario, it undoubtedly is only too common.
Clearly this scenario reflects an accident looking for a place to happen, based on the unspecified objectives of both user and programmer. If we accept the user’s objectives as specified above, how do ‘traditional’ interpolation procedures perform?

* Part 1. Traditional Methods

Again from Gold [1984], the most common traditional technique is to specify a grid over the map area and estimate elevation values at each grid prior to stringing contours through the grid. Thus, in general, elevation values of data points will not coincide with grid nodes, and the surface will be ‘smoothed’ to an extent depending on grid size. The most common method of estimating grid node elevations is some form of weighted average of the ‘neighbouring’ data points. Various other techniques are also used, in particular the generation of polynomial patches that fit the data of a small part of a map to some particular degree of accuracy, and are mathematically constrained to have a specified level of continuity between adjacent patches. A particular weakness of the patch approach is that in some systems the patch only approximates the data in that region. For all patch methods, however, the mathematical constraints to make adjacent patches fit together inevitably reduces the degree of continuity along patch boundaries — and this (usually slope continuity and curvature discontinuity) remains surprisingly visible on the resulting map, producing unacceptable results. For weighted-average techniques, the basic difficulty is in the compatibility of two separate steps — the selection of the adjacent data point to be used, and the range of influence of the weighting function applied to each of these neighbours [Gold, 1984]. It is particularly difficult to ensure that a data point’s influence (weighting) has decreased to zero before it is discarded from the neighbourhood set; if this condition is not met, discontinuities in the surface will occur.

Lastly, as more types of data become available to a digital mapping system there is a strong need to be able to distinguish between, and handle appropriately, information that comes from a point source and information that pertains to a whole region or area.
TRENDS AND CONCERNS OF SPATIAL SCIENCES

Part 2. Design of Solution

The Missing Census Data Problem

Of the issues addressed in the previous section, the most significant appear to be the difficulty of always honouring all data point values, and the confusion between point and area observations. The first of these issues led to the necessity of using an adaptive methodology describing the relative locations of adjacent data points. The second issue required that point and area observations be treated alike.

This second point of usually addressed anyhow — by treating all area observations as point observations. For reasons to be discussed further below, the dual viewpoint appears to be more valid: all data points are expanded to areas. These areas, when expanded to meet the areas generated by adjacent point, would form the well-known proximal map (or Thiessen polygons, or Voronoi diagram) of the generating data points. There would thus be no functional difference between this diagram, a map of the coterminous United States, or a set of regular (square or hexagonal) grid cells formed implicitly by a regular grid or raster set of point observations. Thus subsequent processing could be handled consistently for all (or mixed) data types.

The value of this view was enhanced by a question posed by Tobler (originally in an invited talk in 1982, but subsequently published in Tobler and Kennedy [1986]). He was concerned with the problem of estimating the value for a single census area whose observation had been spoiled for some reason. He illustrated with a map showing the state of Kansas (Fig. 1a). If the value (e.g., cars per household) had been invalidated for some reason, how best could it be estimated from the values of the surrounding states? His suggestion was to use a weighted average of the surrounding values, with the weights being proportional to the length of the common boundary between that state and Kansas.

Examination of this technique showed that the common boundary length could easily be a very poor estimator of an adjacent state’s influence; for example, if the common boundary was highly convoluted by comparison with the other relevant boundaries. A more appealing concept was initially to assume that the region under question did not
Fig. 1. (a) First- and second-order neighbours to the state of Kansas (from Tobler and Kennedy [1986]). (b) Regions of Kansas annexed by adjacent states.
exist — no more Kansas. The adjacent states would have to occupy the resulting vacuum — presumably by invasion from the previous border with all the armies marching at the same speed until all available territory was occupied. Then, sneakily, Kansas reappears, and steals all the newly acquired territories of the border states. The relative areas stolen from (or originally lost to) the adjacent states, as shown in Fig. 1b, form the weighting of each state’s census value in order to estimate a value for Kansas.

Two conclusions can be drawn from this analogy, both arguably ‘intuitively reasonable’. Firstly, if a new region is created, it takes an estimated ‘value’ calculated as the areas stolen from the pre-existing regions, times their observed values, and divided by the total area stolen. If values are considered as elevation, this is equivalent to overall conservation of volume. Secondly, any point in ‘vacant territory’ is assigned to the nearest adjacent region, probably using Euclidean distance measure. Thus the boundaries of the newly acquired territories would be a form of the ‘skeleton’, or medial-axis transform, of the region (Kansas) being evaluated. The sub-regions themselves are fairly stable in area, with minor perturbations of common boundaries, and appear to form a good basis for weighted-averaging.

The Point-Interpolation Problem

Thus ‘area-stealing’ seems an empirically plausible technique for weighted-averaging of ‘missing’ regions. The approach may also be used for Voronoi diagrams of point data. How is this done? Let us start by taking an exact analogy with the ‘Kansas’ model. Each data point has an associated ‘nearest Euclidean neighbourhood’ — or Voronoi region, or Thiessen polygon — calculated by one of various well-known techniques. As with Kansas, let us remove this point, and region. The adjacent polygons will advance and take over the vacant territory — producing identical results to a Voronoi region calculated without the ‘Kansas’ point in the first place. If the point is re-introduced, territory is re-captured, and the relative areas of the adjacent regions provide weightings for the neighbouring data points. It must be noted that while this writer developed his methods independently on the basis of Tobler’s ‘Kansas’ problem, Sibson [1982] had suggested a polynomial patch technique based on Voronoi region subdivision and spherical polynomials.
A weighted-average of the neighbouring data point elevations will provide an 
elevation estimate — but we already have our original value! We may thus proceed in 
one of several ways, depending upon the problem. This new elevation estimate is a 
'smoothed' value, based on the neighbours, much like a simple averaging filter used on 
a raster data set. The weightings may be applied to the slopes in each of the X and Y 
directions between the central point and each neighbouring point, in order to provide a 
weighted-average slope estimate of each data point. Or, for proper interpolation 
purposes — to estimate elevations at a new location — the procedure is reversed.
Starting with our original data points and their triangulation and Voronoi polygons (Fig. 
2a), a new dummy data point is inserted at the desired sub-sampling location (Fig. 2b), 
and the neighbouring regions are identified and their areas calculated. Then the dummy 
point is removed, and the increase in size of the previously-defined neighbourhood 
regions calculated (Fig. 2c). The weighted average of the neighbouring point elevations 
is obtained as before, and this interpolated value preserved for further use. Note that the 
definition of ‘adjacent’ or ‘neighbouring’ polygons or data points is internally 
consistent; if no area is stolen, there is no common boundary and there is no adjacency. 
Note also that the insertion of a sampling point and polygon may be made at any desired 
location to suit the sub-sampling scheme required.

Let us examine in one dimension, as in a cross-section, what the implications of the 
above weighted-average technique are. Figure 3a shows three data points (D.P.) spaced 
along the X-axis. The dotted line represents the location of a desired interpolated value. 
In Fig. 3b, the one-dimensional Voronoi regions have been defined, and the ‘elevation’ 
value at each data point associated with each region. The Voronoi boundaries are clearly 
the mid-points between each data point pair. In Fig. 3c, the Voronoi region of the new 
point to be interpolated has been defined, and has stolen part of the two adjacent 
regions. These weights are then applied to the elevation values of the two data points to 
provide an interpolated value at the desired location.

Figure 4a shows the result of this process for a variety of interpolation locations. The 
result is clearly to produce linear interpolation between data point pairs. Thus our basic 
area-stealing process in two dimensions is a direct analogue of linear interpolation in 
one. This, however, is not a desirable end-product — we do not want slope
Fig. 2. (a) Original data point distribution with triangulation and Voronoi polygons. (b) The dummy data point is inserted into the data structure, modifying the existing polygons. (c) Upon dummy data point removal, the adjacent polygons annex the vacant area.
Fig. 3. (a) Original data points. (b) One-dimensional Voronoi of the data points.
(c) Interpolation — point Voronoi region introduced,
Fig. 4. (a) Result of interpolation using method of Fig. 3c.
(b) Introduction of smooth interpolating function.
(c) Introduction of slopes at data points.
discontinuities at our data points. Figure 4b shows one possible solution. A Hermitian interpolating function is applied to each of the weights prior to summation, so that the first derivative (slope) is zero as the interpolating point approaches a data point. Thus ‘smooth’ curves (in one dimension) or surfaces (in two) may be produced. Lastly, in Fig. 4c the function at each data point is no longer a simple elevation with no slope but is any general function — in this case planar. In conjunction with the previous Hermitian interpolant, ‘smooth’ surfaces in the first derivative can be made to match elevation and slope values for any data point distribution.

As discussed in Gold (1984), the selection of a sub-sampling scheme is a basic part of the design of a ‘contouring’ or interpolation package. Fixed grids are, in general, to be avoided as they impose a fixed sampling frequency that may well be inappropriate to the distribution of the particular data set. Triangulating the data set as given (using the Delaunay triangulation, the dual representation of the Voronoi polygons described above), followed by sub-sampling on a regular $n$ by $n$ grid of sub-triangles within each main triangle, provides an adaptive scheme that compensates for data anistrophy and yet ensures that the interpolated surface still passes through each original data point. Nevertheless, for a variety of applications a grid is the desired output; subsequent perspective views are included to illustrate this point. Figure 5 shows a test data’ set quoted in Davis [1973] and its triangulation. Figure 6 shows a perspective view of the basic triangulated model. In Fig. 7, the above-mentioned sub-sampling scheme is used providing interpolation between data points but honouring data point values. In Fig. 8, the model has been re-sampled on a grid for display purposes, to facilitate the illusion of depth, but in consequence original data point values are not preserved. Figures 6, 7, and 8 are from Gold (1987).
Fig. 5. Davis’s data set and triangulation.
Fig. 6. Perspective view of triangulation from Fig. 5.
Fig. 7 Interpolation by sub-sampling within triangles.
Fig. 8. Interpolation by sub-sampling on a grid.
Conclusions

A variety of issues that are handled only with difficulty, if at all by traditional methods, may be readily resolved by creative theft! The concept of deleting areas, permitting encroachment from adjacent regions, and thus determining the relative contribution of each of these neighbours appears to be sufficiently general to be appropriate for missing-census-district type problems and point interpolation. Point data are first converted to area data by using a Voronoi polygon generator prior to calculation of neighbourhood weights by area stealing. Various functions may be applied to these derived weights in order to provide the desired level of continuity of the interpolant at the data points themselves.

This approach is attractive by comparison with traditional methods both because the input data is honoured under all circumstances and because of the generality of the ‘area-stealing’ concept.

This research was partially funded by the Energy, Mines and Resources Canada Research Agreement program.

References


Discussion

**Question:** The Voronoi regions, can they be produced for 3-D objects as well as volumes?

**Answer:** Yes, and of course the nice thing about this is that it is a local process. As Y.C. was talking about with splines, you can guarantee that any point is not going to perturb the whole network. If you use any other definition of a ‘good’ triangulation, I don’t think you can guarantee that it’s not going to make waves over in the far corner of the map when we add a new point. That’s one of the nice things about the Voronoi process.

**Question:** The question Max asked was different. Is there an equivalent to Voronoi diagrams in 3-D space?

**Answer:** Yes, it’s obviously not easy, but conceptually, yes. Crystals grow in 3-D space.

**Question:** Can I come to the blackboard? You have this profile, and you have a point, and you get a Voronoi region around it. Then you have another point, and then you create a Voronoi region around it. Then what you get with your weights is the stolen area from the two sides. Now what if I do an interpolation; I assign that weight back here. Let’s say this is the total area of A, and that would be A1, A2, etc. If I now do an interpolation I can now create some kind of a smooth curve between them. I wonder if there is some way like that to interpolate Voronoi diagrams.

**Answer:** I would expect so if you can define it.

**Question:** The problem of interpolating the Voronoi diagram is that given a point, what can I know about the relative influence of the two sides? The inverse problem is now we get a smooth curve, and then ask the question: Given that point what would be its Voronoi extent?

**Answer:** I have not thought about that. But it’s interesting because here I am approaching this from the idea of weighted averages. There has been no polynomial function in any of this. I did work with polynomials and I found that I was producing artifacts at triangle boundaries, or patch boundaries, and I was clearly getting first or second derivative discontinuities that showed up on my interpretation of my contour map. One thing about the weighted average process is that the Voronoi is generalizable in principle from one to n dimensions, and the weighted average does not intrinsically have any discontinuities in any derivatives. So this has quite a lot of potential for general process. I believe there is some reasonable physical analogue. It was not accidental that I moved on to talk about line Voronois as well as point Voronois. Take the very interesting question of interpolating from contour intervals. Contour input is not a set of points. If you use points for contour input-bluntly, you get strange things happening. The key with this is to define what you mean by an object. Just as the key to the skeleton coding process is what you mean by an object. You have to count the edges as different objects if you want to look at the boundaries between them. If you use the point Voronoi for this, you have taken all of these different points and called them different. You have boundaries between them. We have a great difficulty making any interpolation here that’s different from the contour line itself. Certainly you can use
the slope data, but I have not been very successful. I believe the problem is that we should be treating this as a segment process and not a point process. This is not something that I have worked on in great detail. I know that I have had problems in the point process.

**Question:** I think that one of the problems is that this is not very attractive because what you produce are not very nice curves. You get paraboloids.

**Answer:** That is for the Voronoi diagram itself, yes. But that does not affect the smoothness of your interpolation process. That is simply the boundaries of the areas that you steal from.

**Question:** But you have to calculate areas under them; you have to calculate sections of them.

**Answer:** Yes, they are not cheap.

**Remark:** There are probably a couple of strange conditions.

**Answer:** Well, I have been working on this for a few years. On point processes I have had no difficulty.

**Question:** I very much like this approach, but I would like to ask you probably a dumb question. Your interpolation is based on the Delaunay triangulation which has resulted from this particular technique. This means that every time you interpolate you use at most three values, or at least the immediate neighbours of the point.

**Answer:** You use typically a half dozen values, because typically you are going to get six adjacent points when you steal a piece of the region by adding a new point, even if it’s an imaginary point.

**Question:** But it is going to be the immediate neighbours, which means that the number depends on how many immediate neighbours exist.

**Answer:** That’s right. It’s entirely a function of what’s really there and not how many ought to be there.

**Question:** So this is really quite good for spatial type applications. I am considering more geological type applications, where you have more points around an area. You remember from the Voronoi diagram that you have an area of influence that has been defined from the sample data. If your area of influence is larger than your immediate neighbours, probably you would have to consider more points than the immediate neighbours and the accuracy of these estimations as well, because in digital terrain models all of our points are supposed to have the same accuracy. If you have points which are farther away but which are more accurate than the immediate neighbours, probably you would like to consider them as well. I am wondering how different would the Kriging method be from your method.

**Answer:** I think there is a lot of commonality. I think we are talking about a great many of the same underlying processes. There are three points I would like to make. Firstly, Kriging is a weighted average process despite all of the other fun and games that
go into producing it. I haven’t mentioned weighted Voronoi diagrams. Supposing you have a radio tower from the Voice of America, which is the most powerful one I can think of now, next door to Annapolis Valley Radio. They are at the same frequency. The zone at which the signal is stronger is going to be a very small region around Annapolis Valley Radio, but even on the other side of it there is going to be a region of the Voice of America. So, you no longer have simple straight line boundaries. You can see easily how theregions enclose other regions. So there are lots of good things you can do with weighted Voronoi. I believe that principle will work. But again, the functional algorithm is going to be relatively difficult. The second point is that we use immediate neighbours for the function at each data point. If the function that we are weighting is somewhat complex, let us say elevation plus slope, then in calculating that elevation plus slope, I use the point’s neighbours to produce a weighted average that generates the slope. Then I weight that function, plus others, to interpolate to the new point. So, I have gone to my second generation of neighbours in order to estimate my slope. After all, slope is saying something about the relationship of the point to its neighbours surrounding it. The third point I would like to make is that because Kriging does some very nice things while being able to make some error estimates, we can do the same thing here because we now have no difference between an interpolation point and a real point. The only difference is that we happen to know the value at the real point. Let’s hide it and pretend we don’t. What value do we get? We have weighted average estimates. We can start talking about the normal distribution of estimates, and then start saying how does this distribution relate to — ah ha!. I fooled you. I’ll bring back my real data point. So we have quite a lot of comparison processes available.

**Question:** Also one important thing would be to try to model discontinuities in your structure?

**Answer:** That’s relatively easy because with any weighted average process you simply have to have a structure which allows you to identify the points which count and the points which don’t count.

**Question:** But with all interpolation methods it is very close to black magic. You have a limited amount of information and you produce new data. You put in some understanding of what is likely to be there. For example, Y.C. in his talk pointed out very clearly that you assume that the surface of a curve is continuous in some sense. So you make some additional assumptions. I think in most interpolation methods I have seen there is other information. But very seldom is it explained exactly what the additional knowledge is that is put in to produce that new information. You have to put something in.

**Answer:** It’s a cautionary tale to realize that despite all these fancy pictures at the bottom, the top is data.

**Question:** Only the points at the top are data. That’s the only thing you really know. Everything else is based on assumptions, which are perhaps justified and perhaps not.

**Answer:** That’s true. One of the things that has annoyed me the most with many contouring packages is that they give you an accuracy figure. You get 98.5% and you say that sounds good, but what does that mean? That means that they have approximated the data to match the data points 98.5% of the time. That’s all they are
saying. So they should have done it to 100%. It says nothing about the function that’s underlying the process.

**Question:** You cannot say anything useful unless you know something about what process affected the surfaces between the data points. What type of surface you expected.

**Answer:** But, in general, we do know something. We know that we can mislead by using arbitrary processes — by n-k-selecting the number of neighbours we wish to use in such a manner that we throw out a point because it has fallen out of an octant or it has moved over a fraction. It’s just fallen out of our circle or radius process or whatever, and it still has a significant weight in our function. So we have a strong interaction between points and weight, and one of the strengths of the Voronoi process is that weights and neighbours are directly related.

**Question:** If you use one of the methods you used at the beginning and you shift the grid a little bit, you get a different result.

**Answer:** If you are using a grid, yes. You won’t with mine.

**Remark** Yours is invariant. That’s one thing I like about it. We cannot say that one of the results by one of the non-invariant methods is correct and the other is wrong. We just can’t say that.