

Comparison of Three Data Expansion Algorithms for Air Pollution Data in Irregularly Placed Measuring Stations

Hiram Calvo

Instituto Politécnico Nacional, Center for Computing Research, Mexico City, Mexico
hcalvo@cic.ipn.mx; hiramcalvo.com

Abstract. In some major cities there are stations for measuring atmospheric pollutants. These stations are often distributed in an irregular pattern. In order to predict pollutant's behavior, it is necessary to order data in a regular, uniform grid. For this, we employ expansion data algorithms. Our work centers on the software implementation and evaluation of three of these algorithms: Cressman, Voronoi and Kriging. For evaluation, we use real data of atmospheric pollutants, including the actual position of stations that measure air pollutants in Mexico City. We use actual values taken from different pollutants.

Keywords: air pollution, data expansion, interpolation, Cressman, Voronoi, Kriging, OpenGL.

1. Introduction

In some major cities there are stations for measuring atmospheric pollutants. These stations are often distributed in an irregular pattern. In order to predict pollutant's behavior, it is necessary to order data in a regular, uniform grid. Such an interpolated field is critical for calculus such as wind field divergence, reduction, data value edge, and initialization of calculus of pollutant transport [13]. In real world, it is impossible to obtain exhaustive values of data at every desired point because of practical restrictions. Because of this, the interpolation is important to graph, to analyze and to understand bi-dimensional data [5].

There are expansion data algorithms well known for data interpolation. These algorithms are used in a wide range of applications, but they have not been compared for the particular problem of interpolating scattered data of air pollutants. In this work we describe the software implementation and evaluation of three of these algorithms: Cressman, Voronoi and Kriging.

For evaluation, we use real data of atmospheric pollutants, including the actual position of stations that measure air pollutants in Mexico City. We use actual values taken from different pollutants.

Firstly, we present a review of the current situation with the software that allows data interpolation. We particularize on the software related to geological applications. In Section 2, we describe the mathematical fundamentals of each of the algorithms selected (Cressman, Voronoi and Kriging) and present some details on their implementation in Section 2. Afterwards, we compare the algorithms using the actual

data on the air pollution in Mexico City, including the exact positions of the measuring stations. For comparison, we follow the procedure of removing one point from the actual data set; then we interpolate the data for this point, and after that we compare the interpolated value with the actual value for this point. We apply the procedure repeatedly to obtain a measure of the quality of interpolation of our algorithm. This is explained in detail in Section 4.

We show the implementation details in Section 3 which allow interactive visualization of surfaces. For the visualization purposes, we use OpenGL and the Graphics Library Utility Toolkit (GLUT).

1.1. Related Software

To our knowledge, no particularly specialized software currently exists for the air pollution modeling. However, there are programs for geologic modeling that include at least one of the algorithms presented. One example of these programs is **GMS: Submarine Surface Modeling System**. This program has the ability to perform surface interpolation using the Kriging algorithm in two variants:

Zonal Kriging. A variogram can be defined for each stratographic area. Zones are defined by the material identifiers associated with the cells of a three-dimensional mesh which surrounds scattered points. When a cell is interpolated in a zone, only the scattered points found in the same zone of the mesh are used to interpolate the cell.

Indicator simulation. Indicator Kriging is a form of Kriging used for interpolating integral or zonal information instead of scalar values. For each scattered point a material identifier is assigned, and material identifiers are interpolated to the mesh cells. This makes possible to establish stratographic zones for a model of submarine surface using Kriging.

Other programs including the algorithms studied are those related to mesh processing. There are many programs for mesh processing. All these programs are for general mesh processing. These programs have interpolating algorithms implemented, but they are not specialized in data expansion, much less particularly in the problems of air pollution. Examples of these programs are: Algor, ARGUS MeshMaker, EarthVision, FEMAP, Finite-Octree, GOCAD, GRIDGEN, Hypermesh, ICEM CFD, Patran, STRATAMODEL, and TrueGrid.

Among the data expansion algorithms used we will focus on three of them (namely, Cressman, Kriging and Voronoi), which have been traditionally used in the problems similar to the one we are trying to solve [10]. We describe those algorithms in the following section.

2. Algorithms

In this section, we describe three algorithms for data expansion: Cressman [4], Kriging [11], and an algorithm based in Voronoi triangulation [8, 2, 1].

2.1. Cressman

A common way of addressing scattered data for its interpolation into a regular grid is to assume that the grid value is a weighted value of the surrounding data values, that is:

$$C_{ij} = \frac{\sum_{k=1}^n C_k W_k(r)}{\sum_{k=1}^n W_k(r)}, \quad (1)$$

where C_k is the value measured at the k -th measuring station, $W_k(r)$ is the weighting function, and r is the distance from the grid point to the station.

Cressman proposed a procedure for height surface analysis, in which he used the following weighting factor [4]:

$$W(r) = \frac{R^2 - r^2}{R^2 + r^2}, \quad (2)$$

where R is the distance from which the weighting factor tends to zero; that is, the 'influence ratio'. This weighting technique helps to the interpolation procedure in scattered data areas. Decreasing values of R are used in consecutive tests to analyze a scale spectrum. Values obtained from each test are averaged to produce the final field.

Endlich and Mancuso [7] combined the polynomial adjustment and the distance weighting in their interpolation technique. An adjustment of minimal-quadratic to a first order polynomial was performed using the closest five stations, in correspondence with:

$$W(r) = \frac{a}{(r + r^*)^2 + a}, \quad (3)$$

where a is a constant, r is the distance to the station, and r^* is a distance factor ($0 \leq r^* \leq r$) which depends on the observation. It can be top-down ($r^* = r$) or crossing ($r^* = 0$) from the viewpoint of the grid.

2.2. Kriging

2.2.1. Fundamentals

Kriging technique is known also as "optimal prediction". It is a method of interpolation that predicts unknown values of observed data in certain places. This method uses a variogram to express the spatial variation, and minimizes the error of predicted values that are estimated by the spatial distribution of the predicted values.

Ordinary Kriging estimates the unknown value using linear combinations weighted from the available sample [12]:

$$\hat{v} = \sum_{j=1}^n w_j * v \quad \sum_{i=1}^n w_i = 1 \quad (4)$$

The estimated i -th error, r_i , is the difference between the estimated value and the true value at the same location:

$$r_i = \hat{v} - v_i \quad (5)$$

The average error in a set of k estimated values is:

$$m_r = \frac{1}{k} \sum_{i=1}^k \tau_i = \frac{1}{k} \sum_{i=1}^k \hat{v}_i - v_i \quad (6)$$

The error variance is:

$$\delta_R^2 = \frac{1}{k} \sum_{i=1}^k (\tau_i - m_R)^2 = \frac{1}{k} \sum_{i=1}^k \left[\hat{v}_i - v_i - \frac{1}{k} \sum_{i=1}^k (\hat{v}_i - v_i) \right]^2 \quad (7)$$

2.2.2. Behaviour

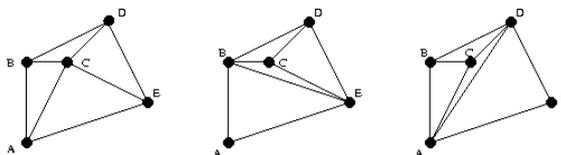
Many properties of Earth’s surface vary in an apparently random, although spatially correlated, manner. Using Kriging for interpolation allows us to estimate confidence in every interpolated value in a better way than other methods [9].

Kriging is also the method which is associated with the acronym B.L.U.E. (best linear unbiased estimator). It is linear because the estimated values are the weighted linear combinations of the available data. It is unbiased because the average of error is 0. It is the ‘best’ because it tends to minimize the error variance. The difference between Kriging and the other estimation methods is its tendency to minimize the error variance.

2.3. Voronoi Algorithm

2.3.1. Delaunay Triangulation

There are several ways of triangulating, given a set of points:



Sometimes it is necessary to perform a triangulation of the points with certain properties. One of the most common and useful triangulations is the Delaunay

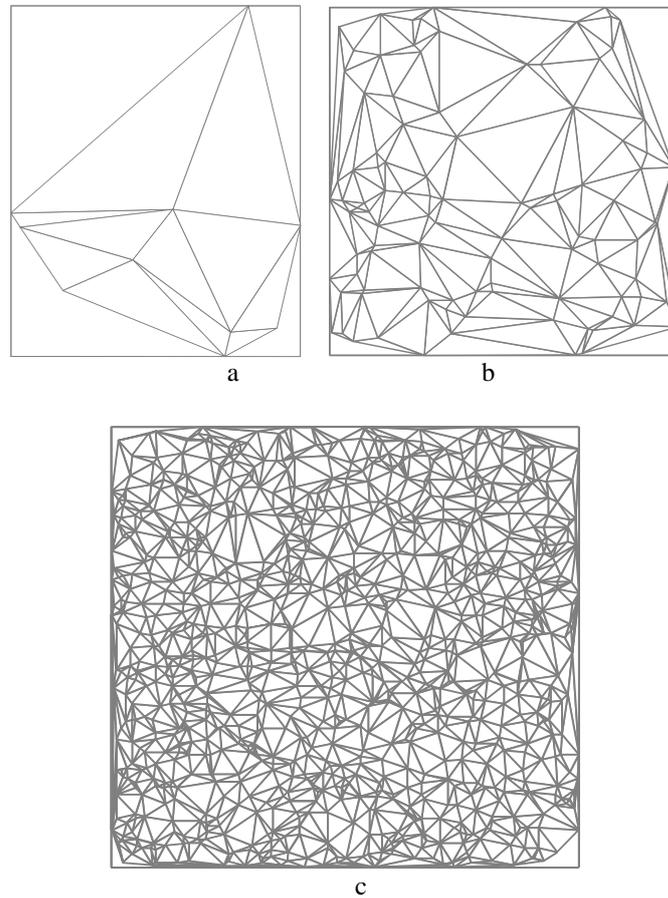


Fig. 1. Examples of Voronoi diagrams with (a) 10 points (b) 1000 points and (c) 10,000 points.

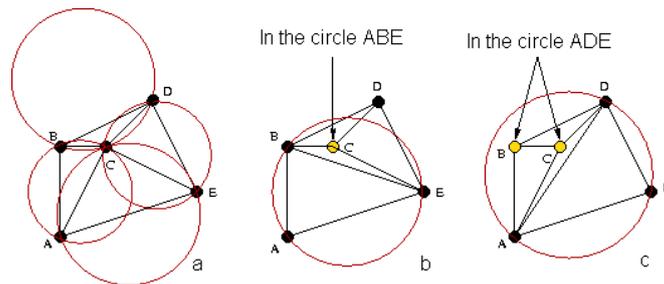


Fig. 2. A Delaunay Triangulation (a) and two non-Delaunay triangulations (b, c).

triangulation, named after the Russian mathematician Boris Delaunay. Delaunay's triangulation of a set of points is given by the following property:

AB is an edge of Delaunay's triangulation if and only if, there exists a circle that passes through A and B in such a way that any other point in the set of points, C, where C is different from A and B, is outside of the circle.

Equivalently, all circles in the Delaunay's triangulation for a set of points will have empty circumscribed circles. That is, there are no points inside the circumference of any triangle (see Fig. 2).

We can see immediately that the first triangulation is a Delaunay triangulation, as all the circumscribed circles are empty.

Delaunay triangulation always exists for any set of points in two dimensions. It is always unique as long as it does not happen that four points in the set of points are co-circular. Because it minimizes small angles and circumscribed circles, Delaunay's triangulation is geometrically convenient, and in general, appealing to sight.

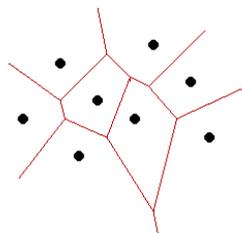
To generate the Delaunay's triangulation, we chose to implement the algorithm of 'divide and conquer' presented by Guibas and Stolfi in [6].

Some examples of the Delaunay triangulations are presented in Fig. 1.

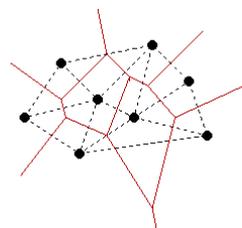
2.3.2. Voronoi Diagrams

Named after the Russian mathematician Georges Voronoi, Voronoi diagrams are also known as Thiessen polygons or the Blum's Medial Axe Transformation. Cells are called Dirichlet Regions, Thiessen polytopes, or Voronoi polygons.

The Voronoi diagram for a set of points, S , in two dimensions (assuming that there are no three collinear points or four co-circular points) is a plane division in polygons. Each point in S is inside some polygon, and each polygon contains exactly one point inside. Each polygon cuts the region that it is closer than any other point in S , to its contained point.



Voronoi diagram is the linear dual of Delaunay's triangulation. This means that we can change from Voronoi diagram to Delaunay's triangulation drawing the perpendicular edges to the region limits, and *vice versa*.



Now that we have the unique triangulation of Voronoi, we face the problem of knowing if a point is inside a triangle, so that this way, and using the equations that

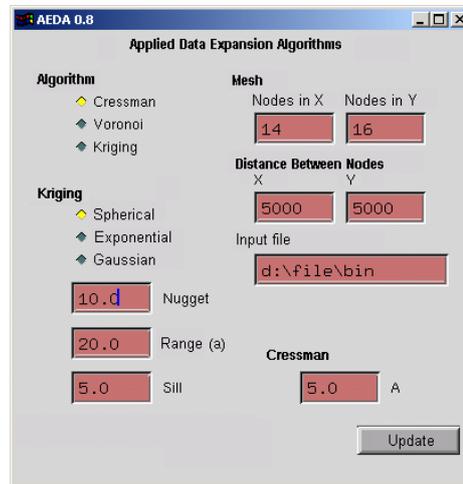
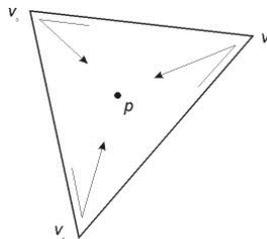


Fig. 3. Program's main interface.

describe a plane in the space, we can obtain the values of the regular grid that overlaps to the generated triangles.

2.3.3. Test of the Inclusion of a Point in a Triangle

Given a triangle (v_1, v_2, v_3) and a point p , the test of inclusion of p in the triangle is:



If we travel across points v_1, v_2, v_3 and the point is inside, we will always look to the point at the same side of the segment we are visiting. If v_1, v_2, v_3 are arranged counter-clockwise, the points outside it will always be to the left of the segments. If the point is outside, at least one of the segments of the point will be on the right side (see figure). If the vertices are arranged clockwise, our reasoning is identical, except that a point that is inside the triangle will be always to the right of the segment we are visiting.

This way, to determine if the point p is inside the triangle (v_1, v_2, v_3) , we must obtain the rotation directions along triplets (v_1, v_2, p) , (v_2, v_3, p) y (v_3, v_1, p) . The point is inside if, and only if, the three directions are the same.

Now that we have an efficient way of knowing if a point is inside a triangle [3], we obtain the value of z using known equations of analytic geometry, which describe a plane.

3. Implementation

3.1. OpenGL and GLUT

For visualizing the results, we chose OpenGL because it is a free graphic library, it is efficient, it is available for every platform which can compile a C program, and it has a wide range of users.

OpenGL is a low level specification of graphic libraries. It gives the programmer a small set of primitive geometric objects: points, lines, polygons, images, and bitmaps. OpenGL provides with a set of commands that allow the specification of geometric objects in two and three dimensions, using the provided primitives, along with commands that control how these objects are rendered.

GLUT is a set of utilities designed for OpenGL. GLUT means OpenGL Utility Toolkit. It is a programming interface which links with ANSI C and FORTRAN to write OpenGL programs independent of the window system being used.

3.2. Interface

We implemented our software in C language, using the OpenGL library for graphical output, and the μ -UI (Micro User Interface) library for the user interface. μ -UI takes advantage of the bi-dimensional drawing in OpenGL. This way, the application can be easily ported to other operating system with a C compiler installed.

The main program's window allows to use the desired data expansion algorithm; some parameters particular to each one of the implemented algorithms, the size and spacing of the mesh desired, and the input file (see Fig. 3).

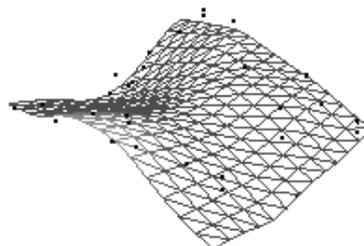
When the 'Update' button is pressed, a window with the mesh rendering will appear. For demonstration purposes, when the input file is not valid, or it does not exist, the program uses the original default functions:

$$x^2 + y^2,$$

or

$$x^2 - y^2.$$

From these default functions, 30 samples are taken in randomly chosen points, and afterwards, as it is done with normal data input, the mesh is reconstructed using the chosen data-expansion algorithm:



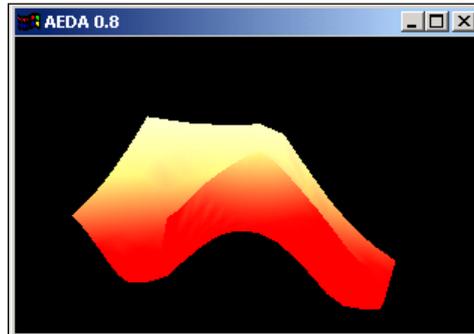
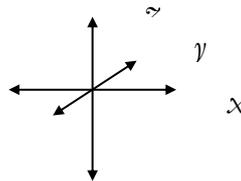


Fig. 4. Sample function reconstructed with the sample points hidden in solid surface mode.

In this image, we show the mesh generated with the Cressman algorithm, with parameter $A = 5.0$. The points where the height sample was taken appear in black dots. These points are used to reconstruct the surface.

Axis convention is as follow:



Rotating movements by keyboard can be done also with mouse, clicking somewhere in the window and dragging the mouse.

28660	41020	0.101	15000	35000	0.161997
24540	43990	0.171	15000	40000	0.166417
45800	50030	0.136	15000	45000	0.154865
28521	49233	0	15000	50000	0.129265
42000	49090	0.251	15000	55000	0.084798
37475	37620	0.292	15000	60000	0.034853
28553	26673	0.013	15000	65000	-0.018312
42150	27800	0.091	15000	70000	-0.053776
41200	37370	0.143	15000	75000	-0.074067
33260	31870	0	20000	0	-0.202291
31711.8	62477	0	20000	5000	-0.200391
23440	54620	0	20000	10000	-0.190037
39516.6	62932	0	20000	15000	-0.140032
28950	31340	0	20000	20000	-0.058771
			20000	25000	0.011562
			20000	30000	0.060174

(a)

(b)

Fig. 5. Fragment of input file and fragment of output file *mesh.out*

Input files must be in text format (ASCII). The first two values are for the position of the measuring station (X and Y coordinates, respectively). Values are separated by spaces. The third value is the value of the measuring. The first two values are integers. The last value is real.

The number of measuring stations (rows) must be less than 65535. Output file format of *mesh.out* has the same format.

4. Experiment and Results

We can judge subjectively the best data expansion algorithm if we examine the accuracy shown for the reconstruction of the original function, because we already know its original form (see Fig. 6 and 7).



Fig. 6. Cressman and Voronoi algorithms.

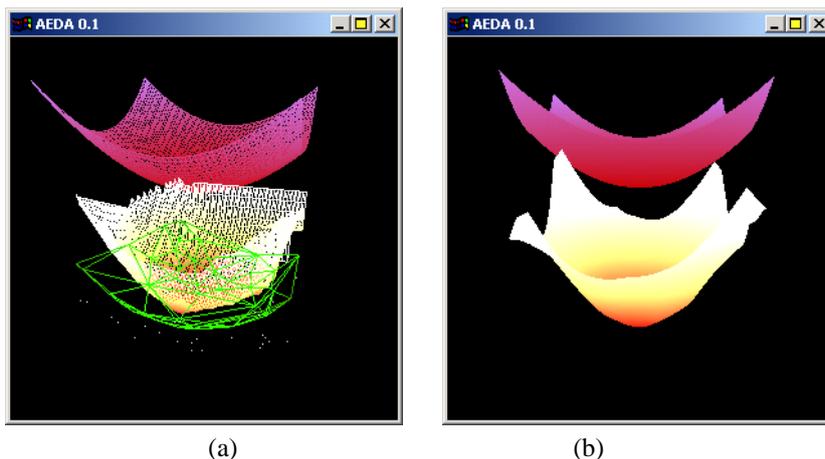


Fig 7. (a) Voronoi in wire frame with overlapping triangulation; (b) Kriging.

For Cressman, we can see the influence rations of the measuring points. This produces an irregular surface, because this algorithm works locally.

Voronoi produces a uniform surface, although we can see spaces at the edges. This is because they are not included in the triangulation, so that they are undefined. In the next image we show the triangulation overlapped in green color.

We can see that Kriging tends to adjust closer to the original function.

Now we will evaluate quantitatively the performance of each data expansion algorithm using the real data of air pollution measuring stations. The measured pollutants are: NO, NO₂, Ozone, RCHO₁, RCHO₂, RCHO₃, RH₁, RH₂, and RH₃. To obtain the deviation error with respect to the original values, we compared the original data within a cell defined by 4 nodes of the mesh, with the value of the closest node of the reconstructed mesh. The sums of differences are shown in Table 1.

Table 1. Absolute sums of differences of original points with reconstructed points.

	Cressman	Kriging	Voronoi
NO	0.466401	0.313273	0.371092
NO ₂	0.109421	0.102081	0.147478
Ozone	0.024482	0.022138	0.132038
RHCO ₁	0.121287	0.244838	0.302807
RHCO ₂	0.47808	0.376325	0.351358
RHCO ₃	1.399525	0.983684	0.99482
RH ₁	0.11386	0.074988	0.076807
RH ₂	0.464336	0.323694	0.319748
RH ₃	1.856715	1.294978	1.280415

From this table we can see that in most cases, Kriging has the smallest difference values, followed after Voronoi. In some cases, Voronoi presents a better approximation, as for RH₂ y RH₃.

Table 2 shows the normalized values displayed as a percentage.

Table 2. Comparison of the three algorithms based on the differences of reconstructed points vs. original points

	Cressman	Kriging	Voronoi
NO	76.68%	84.34%	81.45%
NO ₂	94.53%	94.90%	92.63%
Ozone	98.78%	98.89%	93.40%
RHCO ₁	93.94%	87.76%	84.86%
RHCO ₂	76.10%	81.18%	82.43%
RHCO ₃	30.02%	50.82%	50.26%
RH ₁	94.31%	96.25%	96.16%
RH ₂	76.78%	83.82%	84.01%
RH ₃	7.16%	35.25%	35.98%
	72.03%	79.24%	77.91%

It can be seen from this table that the Kriging is the best data expansion algorithm, in terms of the differences it presents with respect to random sampling points.

5. Conclusion

Kriging is the best linear estimator, although adjusting its parameters is a crucial task for achieving good performance. In addition, it has a longer response time. For real-time applications, we found that the Voronoi algorithm is the best, which in addition, requires no tuning of parameters. Voronoi has a small answer to local alterations. It tends to soften surface's form. Kriging and Cressman, by the contrary, can respond actively to small local alterations. In the case of the gas' diffusion in the atmosphere, the presence of local alterations decreases as altitude increases.

Acknowledgements. Work done under partial support of Mexican Government (CONACyT, SNI, PIFI-IPN, CGEPI-IPN), and RITOS-2, and the PAPIIT UNAM grant IN100405.

References

1. Aurenhammer, F., Klein, R.: Voronoi Diagrams. Ch. 5. In: Handbook of Computational Geometry (Eds. J.-R. Sack and J. Urrutia). Amsterdam, Netherlands: North-Holland, pp. 201–290 (2000)
2. Aurenhammer, F.: Voronoi diagrams – a survey of a fundamental geometric data structure. ACM Computing Surveys, 23(3), pp. 345–401 (1991)
3. Bernardini, F., Bajaj, C., Chen, J., Schikore, D.: Triangulation – based object reconstruction methods. In: ACM Sympos. Comput. Geom., pages 481–484 (1997)
4. Cressman, G.P.: An operational objective analysis system. Mon. Wes. Rev., 87, pp. 367–374 (1959)
5. Devillers, O.: Improved incremental randomized Delaunay triangulation. In: 14th Ann. ACM Sympos. Comput. Geom, pp. 106–115 (1998)
6. Edelsbrunner, H., Mücke, E.P.: Three-dimensional alpha shapes. ACM Trans. Graph. 13(1), pp. 43–72, Ene. (1994)
7. Endlich, R. M., Mancuso, R. L.: Objective analysis of environmental conditions associated with severe thunderstorms and tornadoes. Mon Wea. Rev., 96, pp. 342–350 (1968)
8. Eppstein, D.: Nearest Neighbors and Voronoi Diagrams. <http://www.ics.uci.edu/~eppstein/junkyard/nn.html>.
9. Melkemi, M.: A-shapes and their derivatives. In: 13th Ann. ACM Sympos. Comput. Geom., pp. 367–369 (1997)
10. Moreno, C.: Efficient 2-D geometric operations. http://www.mochima.com/articles/cuj_geometry_article/cuj_geometry_article.html
11. Oliver, M. A., Webster, R.: Kriging: a method of interpolation for geographical information system. Int. J. Geographical Information Systems, 1990, vol. 4, No. 3, pp. 313–332 (1990)
12. Sethian, J.A.: Level Set Methods. Cambridge University Press (1996)
13. Watson, D.F.: Contouring: A guide to the Analysis and Display of Spatial Data. Pergamon (1992)