Interpolation of Scattered Data: Investigating Alternatives for the Modified Shepard Method

KAREN BASSO[1]
PAULO RICARDO DE ÁVILA ZINGANO[2]
CARLA MARIA DAL SASSO FREITAS[1]

[1]UFRGS-Universidade Federal do Rio Grande do Sul, Instituto de Informática, Programa de Pós-Graduação em Computação – PPGC, Caixa Postal 15064, 91501-970, Porto Alegre, RS, Brasil
{karen,carla}@inf.ufrgs.br
[2]UFRGS-Universidade Federal do Rio Grande do Sul, Instituto de Matemática, Curso de Pós-Graduação em Matemática – CPGMat, Caixa Postal 15091, 91509-900, Porto Alegre, RS, Brasil
zingano@mat.ufrgs.br

Abstract. Many scientific applications use scattered data originated from samples. Interpolation techniques are necessary to estimate the values on non-sampled regions. In a previous work, the Modified Shepard interpolation method was implemented and some unpredictable results occurred. Due to the flexibility of parameter definition for the interpolant generation in this method we decided to investigate solutions for those problems. In this paper we present four versions for the Modified Shepard method, and compare their results through the analysis of the images, processing time and accuracy. Future work for further improvements in the method is also discussed.

Keywords: Interpolation Method, Modified Shepard method, Scattered data

1 Introduction

The problem of constructing approximations based upon scattered data are encountered in many areas of scientific applications, like meteorological information, such as the amount of rainfall, or geological information, such as depths of underground formations. This is done using interpolation techniques that estimate values on unexplored points of a region considering the values sampled on it. These techniques could be locally or globally defined. In global functions, the interpolant depends on all sampled points and the addition, modification or exclusion of any point will propagate through all the function domain, see e.g. Franke [1]. In local methods, the modification of one point will affect only the neighboring points. Some locally designed methods construct general-purpose interpolants defining radii of influence for the data points. One example is the Modified Shepard method, see Franke and Nielson [2], which has its flexibility supported by radii R_q and R_w.

In previous work [3], the Modified Shepard interpolation method was implemented to build geological models using values sampled at certain wells. The interpolation method achieved some undesired results, especially on poorly sampled regions. In an attempt to eliminate the problems of the method, some solutions were found and new techniques applied.

The aim of this paper is to describe the changes proposed to the Modified Shepard Method and compare their performance through the analysis of the images, the processing time and the accuracy of results.

2 Description of the Modified Shepard Method

The purpose of this method is the definition of a smooth bivariate interpolant *S* with the property that $S(x_i, y_i)=f_i$, given N scattered data points (x_i, y_i, f_i), i=1,…,N. This interpolant is built taking into account all samples (one nodal function is found for each sample point), with the influence of the sample depending on its position on the region being calculated.

The first step in this method is the definition of the radii of influence, R_q and R_w. While R_q denotes the radius of influence of the data points in the nodal functions, R_w denotes the radius of influence of the nodal functions in the interpolant. These radii are computed according to the relationships:

$$R_q = \frac{D}{2}\sqrt{\frac{N_q}{N}} \qquad R_w = \frac{D}{2}\sqrt{\frac{N_w}{N}}$$

where D = $\max_{i,j}$ d_i (x_j,y_j), and N is the total number of data points. The values of N_q and N_w can be interpreted as representing the number of data points estimated to lie within circles of radii R_q and R_w respectively. While locally defined interpolants require small values for N_q and N_w, a global interpolant is generated making these values near the total number of samples. The default values recommended by the autors (see Franke and Nielson [2]) are N_q=18 and N_w=9 for uniformly distributed data. When some regions are sparsely populated and others are comparatively dense, or when data sets are small (N<25), it may be necessary to increase the values of N_q and N_w The use of the relationship N_q / N_w ~ 2 is useful.

After calculating the radii R_q and R_w, the nodal functions are computed using the least squares method. These functions are calculated based only on data points sampled inside radius R_q. This slight modification gives a local behavior to the method. The distance function ρ_i taken is given by:

$$\frac{1}{\rho_i(x,y)} = \frac{(R_q - d_i)_+}{R_q d_i} \qquad \text{where}$$

$$d_i(x,y) = \sqrt{(x-x_i)^2 + (y-y_i)^2}$$

Here the subscript + denotes the positive part of the quantity concerned, i.e., (R_q - d_i)+ = (R_q-d_i) when (R_q-d_i) ≥ 0, and zero otherwise. Hence, the weight is zero at distances greater than R_q.

The next step is to solve (for each nodal point (x_k,y_k), k=1, … , N) the least squares problem:

$$\min_{akj,\, j=2,\ldots,6} \sum_{\substack{i=1 \\ i\neq k}}^{N} \frac{1}{\rho_i^2(x_k,y_k)} [\, f_k + a_{k2}(x_i - x_k) + a_{k3}(y_i - y_k) + a_{k4}(x_i - x_k)^2 + a_{k5}(x_i - x_k)(y_i - y_k) + a_{k6}(y_i - y_k)^2 - f_i \,]^2$$

This may be done by solving the associated set of normal equations (obtained by setting to zero the gradient of the above quadratic expression) using Gaussian elimination or another linear solver, which is the fastest approach, or we may use other standard techniques like QR factorization or singular value decomposition (i.e., the pseudo-inverse) of the underlying Nx5 least-squares matrix – the extra work may well be worth the effort due to better stability properties of these algorithms, particularly when the matrix is close to rank-deficient, see e.g. Trefethen and Ban [4].

Having solved these N minimization problems, the nodal functions $Q_k(x,y)$ are known:

$$Q_k(x,y) = f_k + a_{k2}(x - x_k) + a_{k3}(y - y_k) + a_{k4}(x - x_k)^2 + a_{k5}(x - x_k)(y - y_k) + a_{k6}(y - y_k)^2, k = 1,\ldots,N$$

These are used in the Modified Shepard formula:

$$D[f](x,y) = \frac{\displaystyle\sum_{k=1}^{N} \frac{Q_k(x,y)}{\vartheta_k^2(x,y)}}{\displaystyle\sum_{k=1}^{N} \frac{1}{\vartheta_k^2(x,y)}}$$

Each grid point is calculated based on the nodal functions computed for the sampled points inside a circle of radius R_w. This is defined by:

$$\frac{1}{\vartheta_k(x,y)} = \frac{(R_w - d_k)_+}{R_w d_k}, \text{ where}$$

$$d_k(x,y) = \sqrt{(x-x_k)^2 + (y-y_k)^2}$$

Again, (R_w-d_k)+ = (R_w-d_k), when (R_w-d_k) ≥ 0 and zero otherwise.

Figure 1 presents the purpose of radii R_q and R_w in the method. There are 18 samples depicted in the region and we need to build a 13x14 grid. According to the method, nodal functions must be determined for each sample point. The function for sample 4, for example, is computed based on samples 4, 1, and 2, while that for sample 18 will use samples 18, 8, 9, and 14. Once we have calculated the nodal functions, we can process the grid positions. Consider the grid position marked by the center of the R_w circle: it will be calculated using the nodal functions of samples 12 and 6. Increasing grid dimensions allows the generation of more detailed regions.

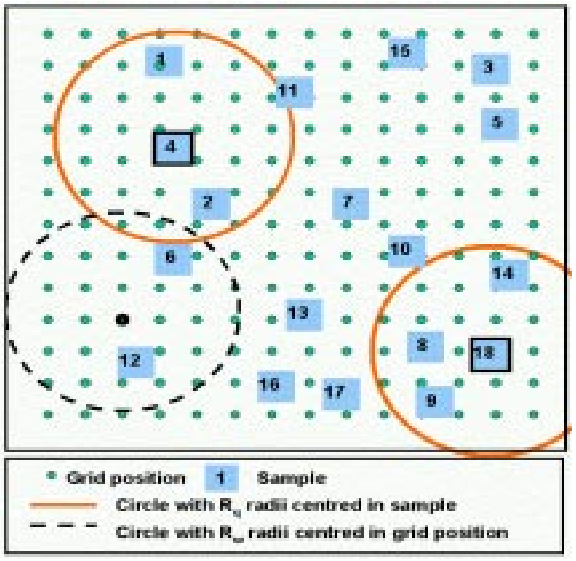

1
15
3
11
5
4
2
7
10
14
6
13
8
18
12
16
17
9
Grid position
1
Sample
Circle with R_q radii centred in sample
Circle with R_w radii centred in grid position

Figure 1 Modified Shepard radii R_q and R_w.

3 Previous results – Version 1

The method implemented following the description in section 2 will be referenced as Version 1. In this version, some problems were detected in the method, which influence the results. The first part of the method, the creation of nodal functions for all sample points, is crucial for the accuracy of the result. In this case, if there are too few neighboring points in the sample, the computed coefficients for this equation could be wrong. An example will be shown below.

Consider the surface $z=2y^2$ and eight sample points: $P_0(0,0,0)$, $P_1(0,3,18)$, $P_2(1,2,8)$, $P_3(2,1,2)$, $P_4(2,4,32)$, $P_5(3,2,8)$, $P_6(3,3,18)$, $P_7(4,0,0)$. In figure 2 we see the result of the interpolation process and the exact (intended) result. Their differences were caused by wrong nodal functions generated for some points.

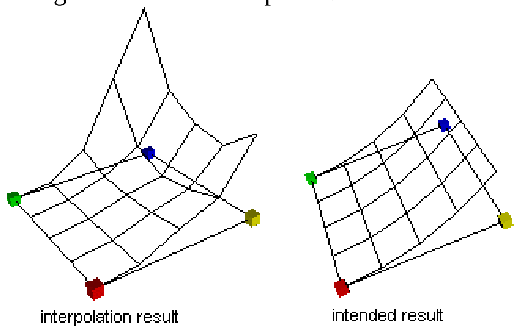


Figure 2 Version 1 result compared to intended[*].

Figure 3 shows the equations generated for sample points. While the equations generated for points 0, 2, 3, 5 and 7 are similar to the correct result, the equations generated for the other points don't agree with the expected ones.

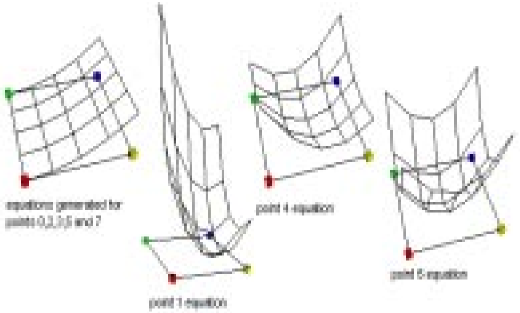


Figure 3 Nodal equations for sample points: (1) sample points 0,2,3,5 and 7; (2) point 1; (3) point 4; (4) point 6.

4 Version 2

In this version some changes were made to guarantee better results avoiding the problem of wrong nodal equations. Essentially, in the first part of the method, the program recalculates the distance functions ρ_i , amplifying the radii R_q until there exist at least five samples around the point of the nodal equation being calculated. For the other nodal functions the radii R_q remain unchanged. A standard requirement of the least squares method is that the number of points involved in the process should be larger than the number of coefficients to be found. In this case there are five coefficients and six points in the process.

In the second part, the grid calculation, we can increase the radius R_w, until at least five nodal functions have been used in the computation

The achieved result was quite better and can be seen in figure 4.

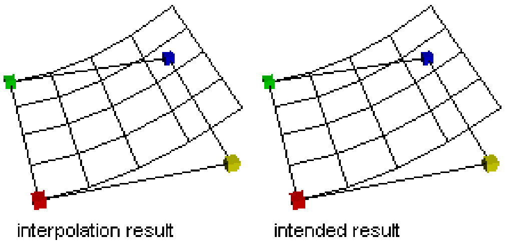


Figure 4 Version 2 result compared to intended.

5 Version 3

This version was made in order to reduce the processing time of version 2. In the present case, the least squares problem fits a linear equation to the data points, instead of a quadratic equation. Thus, the nodal equation coefficients to be found are only two:

$$Q_k(x,y) = f_k + a_{k2}\,(x\text{-}x_k) + a_{k3}\,(y\text{-}y_k)$$

In the first part of the method, the program recomputes the distance functions ρ_i , increasing the radius R_q until there exist at least two samples around the point of the nodal equation being calculated. For the other nodal functions the radii R_q remain unchanged. Here there are two coefficients to be found and three points involved in the process.

In the second part, the grid calculation, we can increase the radii R_w, until at least two nodal functions have been used in the computation.

Figure 5 shows the achieved and intended results from version 3. Note that the result is better than in version 1, with less processing work. Figure 6 shows the plane equations generated to each sample point.

[*] The four reference points helps the viewer orientation for comparison purposes.

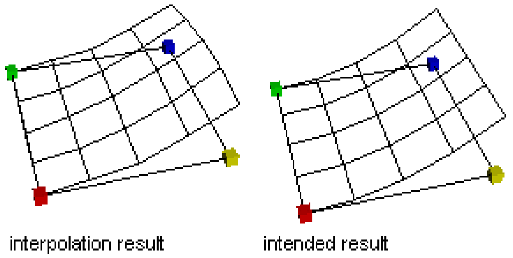


Figure 5 Version 3 result compared to intended.

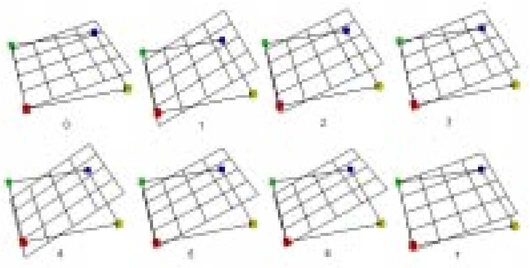


Figure 6 Equations generated for the sample points.

6 Version 4

This is an upgraded version for the previous version 2. Here we added a monotonization step for the coefficients of the nodal equations. The goal of this stage is to smooth the surface. This is accomplished by making the nodal equations take into account the neighboring equations.

After finding the coefficients for a nodal equation, the table of ρ_i values is used to determine what equations will influence the coefficient values and how much (the weight is inversely proportional to the distance). To the ρ_i distance is added a factor *f (influence factor),* a parameter that can be initialized with 1 and can be increased if the user wants to reduce the neighboring influence on the nodal function. For the equation:

$$Q_k(x,y) = f_k + a_{k2}(x-x_k) + a_{k3}(y-y_k) + a_{k4}(x-x_k)^2 + a_{k5}(x-x_k)(y-y_k) + a_{k6}(y-y_k)^2, k = 1,\ldots,N$$

the coefficients found are: a_{k2}, a_{k3}, a_{k4}, a_{k5} and a_{k6}. To perform the monotonization process for the coefficient $a_{k2,}$ for example, we compute:

$$\hat{a}_{k2} = \frac{\left[\sum_{\substack{i=1\\ i\neq k}}^{N} \frac{a_{i2}}{(\rho_i + f)^2}\right] + a_{k2}}{\left[\sum_{\substack{i=1\\ i\neq k}}^{N} \frac{1}{(\rho_i + f)^2}\right] + 1}$$

Figure 7 shows the result obtained with the other test equation ($z=x^4-x^3y+y^2$) in the monotonization process (version 4) , without monotonization (version 2) and the intended result.

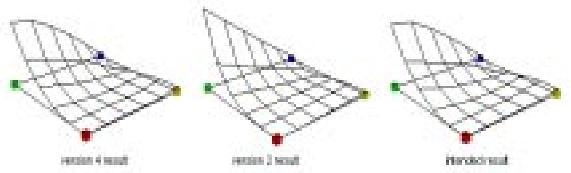


Figure 7 Results with and without monotonization.

7 Discussion

In the previous sections we described all the versions implemented for the Modified Shepard method. In order to compare the results obtained from each version some tests were carried out.

Taking a rectangular region [0,0] x [2,2], random points (x,y) were generated for two sample test files, one with 100, and the second with 200 points. Figures 8 and 9 show the points distribution in files t100 and t200.

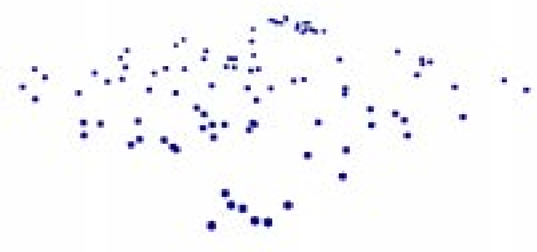


Figure 8 Samples distribution in file t100

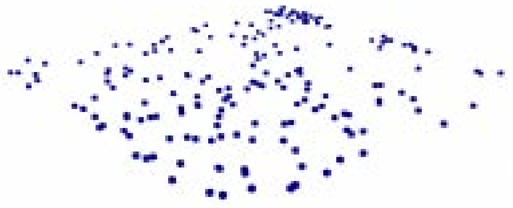


Figure 9 Samples distribution in file t200.

Next, four test equations were used to generate the z values for each sample. The equations were:

f(x,y) = 10 * sin(x) * sin(y) (1)
f(x,y) = (10+sin(x)) * (10+cos (y)) (2)
f(x,y) = 10 * sin(x) * $\sin(y)^2$ (3)
f(x,y) = 10 * sin(x*y) (4)

These equations were chosen in order to know exactly the form of the surfaces to be approximated by the interpolation method, thus allowing computing the error easily. They produced eight test files referred here as t100e1, t100e2, t100e3, t100e4, t200e1, t200e2, t200e3, and t200e4.

All tests measure processing time and accuracy. Processing time is taken in milliseconds. The accuracy measurement is obtained by computing the difference between the intended and the interpolation results for all grid positions. Dividing the computed difference by the intended result at each grid position gives the relative error at that point. In this measurement technique, when the correct (intended) values are too close to zero, the generated relative error could be classified in a range near or over 100%. In this case, the simultaneous analysis of the resulting images and other error ranges are necessary for a correct conclusion.

For better understanding, we will first present specific results from one test equation file (t100e4). Then, we will generalize the results through tables and graphics using average results from all tests.

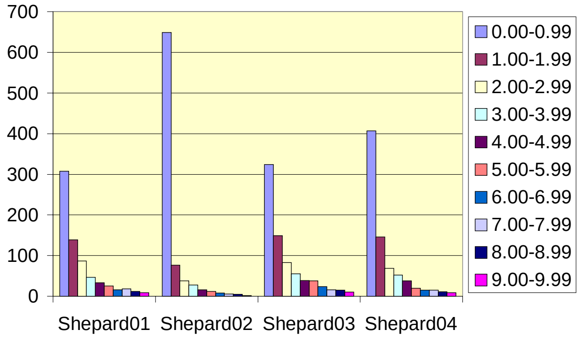


Figure 10 Number of points with errors ranging from 0 to 9.99%. Test file t100e4, 900 grid points.

Error	Shepard01	Shepard02	Shepard03	Shepard04
0-10 %	694	841	753	782
10-20%	32	6	24	10
20-30%	28	3	13	6
30-40%	7	2	2	1
40-50%	2	4	4	5
50-60%	2	1	3	0
60-70%	2	1	1	0
70-80%	3	1	0	0
80-90%	3	0	1	0
more than 90%	97	28	41	40

Figure 11 Number of points with errors ranging from 0 to more than 90%. Test file t100e4, 900 grid points.

Figure 10 shows the distribution of points per error ranging from 0 to 9.99%. We can observe that the version 2 gives more than 600 points with less the 1% of deviation from the intended result. This observation is confirmed by the data shown in figure 12. Version 2 gives the maximum number of correct values (841 points with error ranging from 0 to 10%) and the minimum number of wrong values (only 28 points with error greater than 90%). Both figure 10 and 11 allow to assert that version 4 shows better results than those obtained with version 3, which is better than version 1.

Regarding to processing time, version 3 takes 15% less time than version 1 giving results with similar accuracy (see figures 10 and 11), while version 2 takes only 1.6 % more processing time than version 1 but its results are quite more accurate. Version 4 takes 0.4 % more time than version 2.

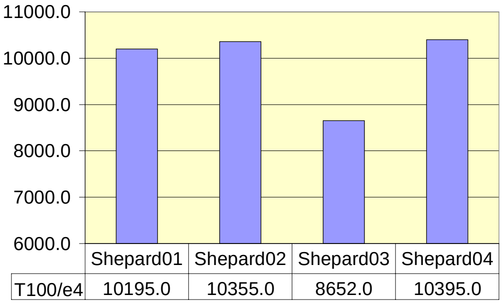


Figure 12 Processing time in milliseconds using test file t100e4.

Although statistics provide useful means to compare general versions' characteristics, images allow a rapid and efficient analysis of results. Figures 13 to 17 show the intended result using equation 4 in a 30x30 grid, and the Modified Shepard versions results.

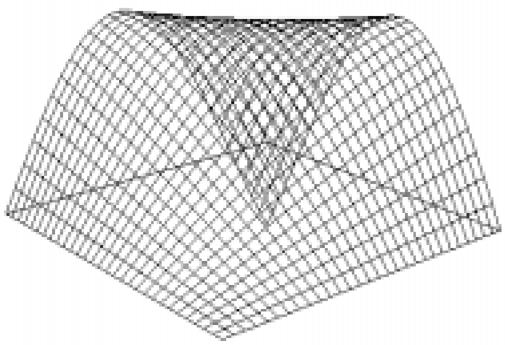


Figure 13 Intended result using equation 4.

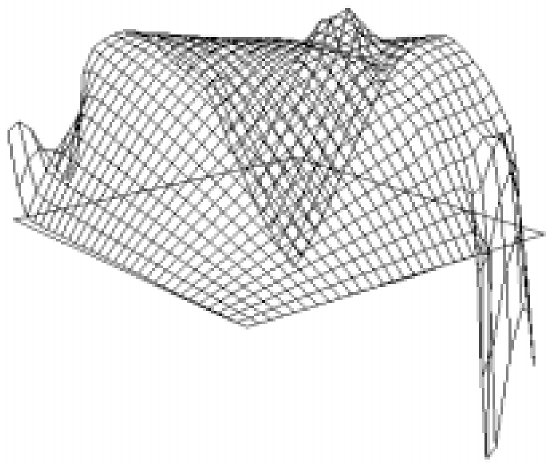


Figure 14 Shepard01 result for equation 4, 100 samples, 30x30 grid, Nq=12, Nw=8.

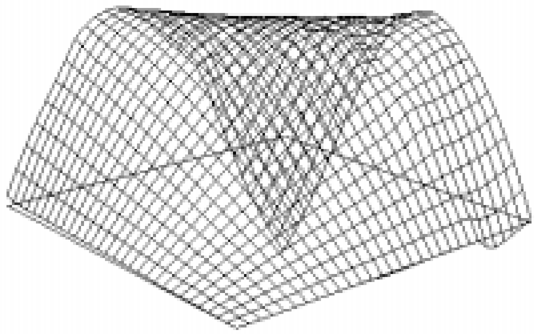


Figure 15 Shepard02 result for equation 4, 100 samples, 30x30 grid, Nq=12, Nw=8.

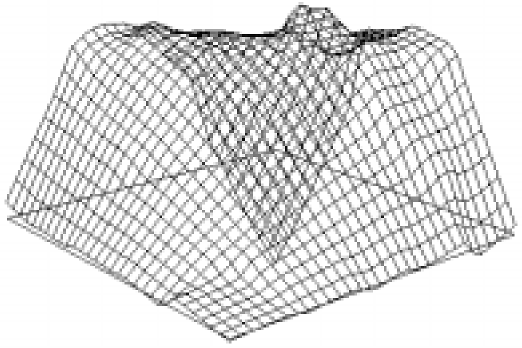


Figure 16 Shepard03 result for equation 4, 100 samples, 30x30 grid, Nq=12, Nw=8.

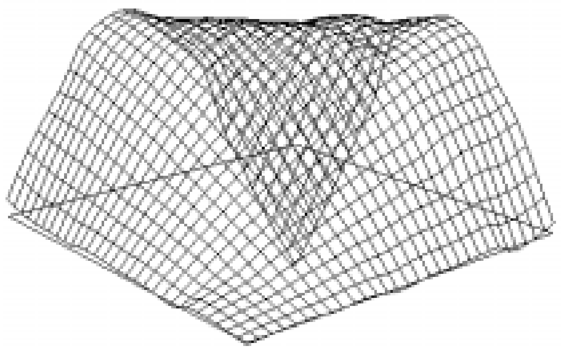


Figure 17 Shepard04 result for equation 4, 100 samples, 30x30 grid, Nq=12, Nw=8.

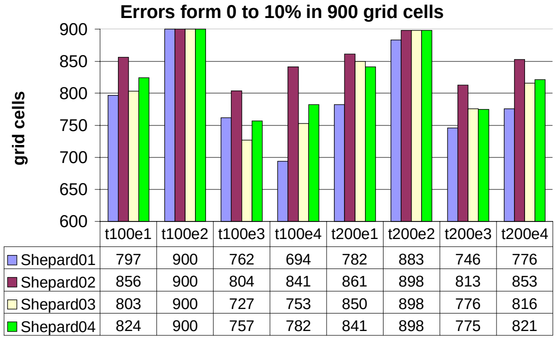
We can observe that the results obtained with version 4 are smoother than those with version 2, only in some sub-regions. This is due to the monotonization process introduced in version 4. In this process the errors are propagated around the grid points, resulting in a less accurate global result. Despite the surface obtained with version 4 shows an overall decrease in correctness, we can noticed a better approximation to the real surface near the bottom-right border.

Version 3 uses planes instead of quadric functions and thus the resulting surface is not so smooth as those generated with versions 2 and 4, but the image (figure 16) gives an idea of the shape of the interpolated function. It could be used as a preview function saving processing time.

Figures 18 to 21 show data collected from all eight tests. The results of tests with all files are shown in figures 18 and 19. These allow a global analysis of the four versions by comparing results from all tests at the same time. While these figures show absolute values for each test, figures 20 and 21 present average values.

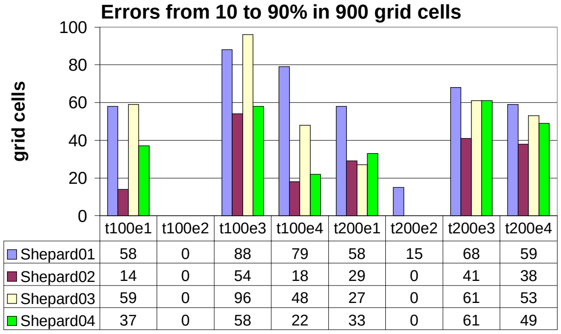
Figure 18 shows the number of grid cells with errors ranging from 0 to 10% in each test file, per Modified Shepard version. In figure 19, the errors ranging from 10% to 90% are shown. In both figures the results obtained previously are confirmed. Again, the version 2 achieved the best results, considering the criterion accuracy.

In figure 20 we can observe the average processing time for all versions. Considering version 1 as basis of comparison, version 2 takes 2.1% more time, version 3, 13.5% less time, and version 4, 2.5% more time, thus confirming the results (figure 12) already discussed in this section. The same occurs when analyzing the average number of correct and incorrect z values. Figure 21 shows that the average of correct grid values is greater in version 2 than in the other versions. Moreover, the average number of grid cells with an error greater than 10% is smaller in version 2.



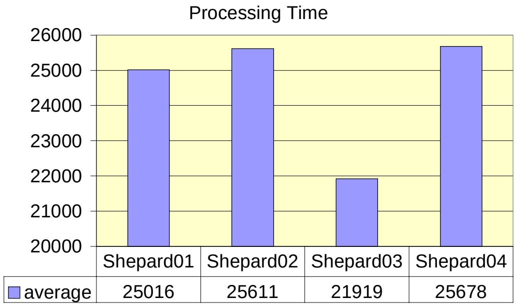
	t100e1	t100e2	t100e3	t100e4	t200e1	t200e2	t200e3	t200e4
🟦 Shepard01	797	900	762	694	782	883	746	776
🟥 Shepard02	856	900	804	841	861	898	813	853
🟨 Shepard03	803	900	727	753	850	898	776	816
🟩 Shepard04	824	900	757	782	841	898	775	821

Figure 18 Errors of Modified Shepard versions from 0 to 10% obtained with all test files.



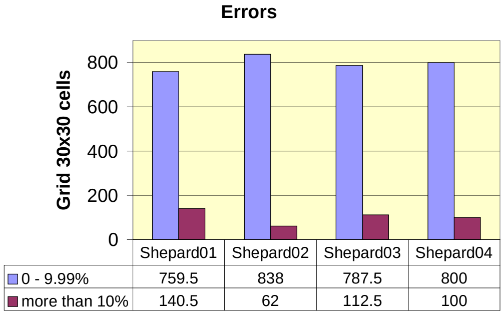
	t100e1	t100e2	t100e3	t100e4	t200e1	t200e2	t200e3	t200e4
🟦 Shepard01	58	0	88	79	58	15	68	59
🟥 Shepard02	14	0	54	18	29	0	41	38
🟨 Shepard03	59	0	96	48	27	0	61	53
🟩 Shepard04	37	0	58	22	33	0	61	49

Figure 19 Errors of Modified Shepard versions from 10 to 90% obtained with all test files.



	Shepard01	Shepard02	Shepard03	Shepard04
🟦 average	25016	25611	21919	25678

Figure 20 Average processing time (in milliseconds) of each Modified Shepard version.



	Shepard01	Shepard02	Shepard03	Shepard04
🟦 0 - 9.99%	759.5	838	787.5	800
🟥 more than 10%	140.5	62	112.5	100

Figure 21 Average grid errors (900 cells) per Modified Shepard version.

The results of the application of versions 2 and 4 to real data are presented in figures 22 to 27. It can be observed that in this dataset there is a global smoothing from version 2 to version 4, especially in the region where there are less sample points.

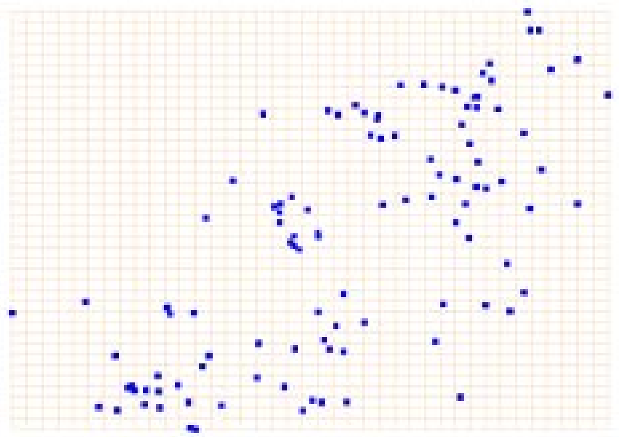


Figure 22 Real sample distribution – top view.

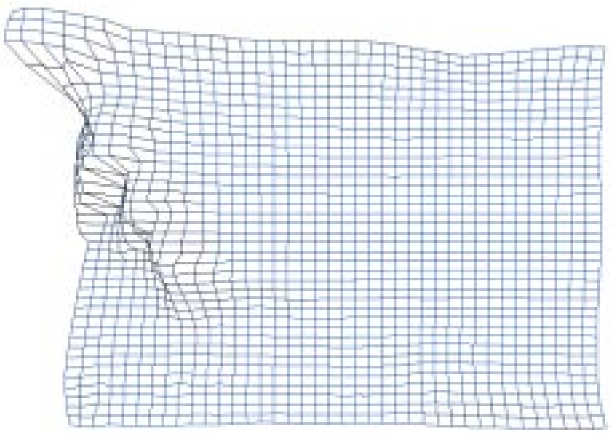


Figure 23 Shepard02 result – top view.

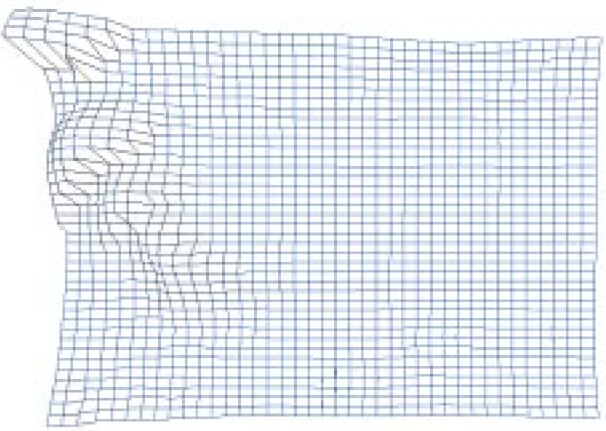


Figure 24 Shepard04 result – top view.

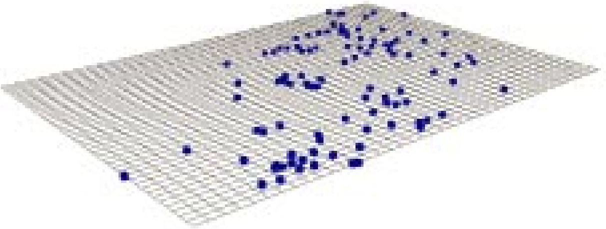


Figure 25 Real sample distribution – perspective view.

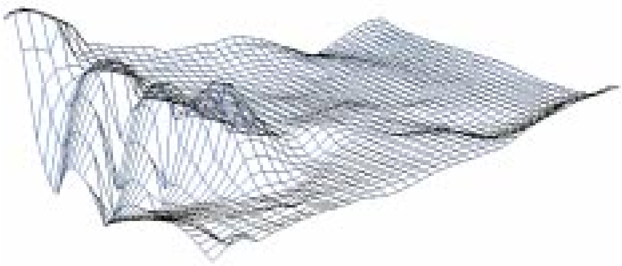


Figure 26 Shepard02 result – perspective view.

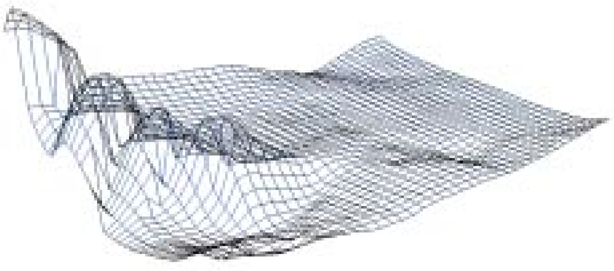


Figure 27 Shepard04 result – perspective view.

8 Final remarks and future work

Achieving an optimized interpolation method taking into account both accuracy and performance is very important for many applications. As reported by Basso and Freitas [3], our motivation was to visualize three-dimensional surfaces approximated from scattered data obtained from oil prospecting activities. However, we can find application of these results in diverse areas such as fluid dynamics and biomedical data. This is possible because of the flexibility of the Modified Shepard interpolation method. Although being locally designed, this method allows to vary radii R_q and R_w, which determines the influence of data points, thus controlling how much global or local will be its application.

We shall consider next the introduction of monotonization techniques of the type found in modern high-resolution methods for partial differential equations like the PPM code originally developed by P. Collela, H. Glaz and P. R. Woodward in the early 1980's. Those methods proved very successful in the simulation of very complex nonlinear phenomena, like compressible convection, fluid instabilities, turbulent flows and interaction of shock waves, see e.g. [5,6,7].

These techniques compute local linear or quadratic interpolants to describe the structure of certain carefully selected flow variables within the grid cells (cell sub-structuring), which are subsequently modified by examining behavior on the neighboring cells, see e.g. [8]. The latter is a monotonization step adding some artificial diffusion, which turns out to be very important for the high resolution/accuracy features of the method.

It seems likely that similar techniques can be employed on the nodal functions $Q_k(x,y)$ above before they are used to generate the final interpolant improving stability and accuracy properties at the expense of a small extra work. This, however, will be the subject of another paper.

References

[1] R. Franke, "Scattered data interpolation: tests of some methods", *Mathematics of Computation.* 38, 157, (1982), 181--200.

[2] R. Franke, G. Nielson, "Smooth Interpolations of Large Sets of Scattered Data", *International Journal for Numerical Methods in Engineering* 15 (1980), 1691-1704.

[3] K. Basso, C.M.D.S. Freitas, "Visualization of geological prospecting data", In: SIBGRAPI'98 - 1998 International Symposium on Computer Graphics, Image Processing, and Vision, Rio de Janeiro, RJ, Brazil. *Proceedings.* IEEE, INPE/USP-SC, 1998,142--149.

[4] L. N. Trefethen, D. Ban III., *Numerical Linear Algebra.* SIAM, Philadelphia, 1997.

[5] P. Colella, P. R. Woodward, “The piecewise-parabolic method (PPM) for gas-dynamical simulations”, *Journal of Comput. Phys.* 54 (1984), 174--201.

[6] P. R. Woodward, P. Colella, “The numerical simulation of two dimensional fluid flow with strong shocks”, *Journal of Comput. Phys.* 54 (1984), 115--173.

[7] D. H. Porter, P. R. Woodward, “High-resolution simulations of compressible convection using the piecewise-parabolic method”, *Astrophys. J Suppl. Ser.* 93 (1994), 309--349.

[8] P. Zingano, *High resolution finite difference methods: HOG, TVD, ENO, PPM.* Short course, Cesup/UFRGS, may, 1995.

Acknowledgments

This work has been supported by CNPq.