

Contour line thinning and multigrid generation of raster-based digital elevation models

ENDRE KATONA

Institute of Informatics, University of Szeged

H-6720 Szeged, Árpád tér 2, Hungary

e-mail: katona@inf.u-szeged.hu

Abstract. Thin plate spline interpolation is a widely used approach to generate a digital elevation model (DEM) from contour lines and scattered data. In practice, contour maps are scanned and vectorized, and after resampling in the target grid resolution, interpolation is performed. In this paper we demonstrate the limited accuracy of this process, and propose a high resolution processing method (without vectorization) that ensures maximum utilization of information in the source data. First, we discuss the mathematical background of thin plate spline interpolation, and explain the multigrid relaxation principle used to speed up convergence. After, we will show why fine tuning is necessary, especially when contour lines and elevation points are processed at the same time. Finally, our own contour thinning method is described that produces a significant reduction of elevation bias.

Keywords: digital elevation model, contour-to-grid interpolation, multigrid relaxation

1. Introduction

It is quite natural to use a regular grid (elevation matrix, DEM) for terrain modelling, and such a data model can be derived from the digitized contour lines of existing topographic maps. The procedure normally begins with manual (or automatic) vectorization of contour lines. Then vector

data are rasterized with a resolution corresponding to the desired elevation matrix, and grid cell values are estimated between contours via some interpolation process. (For an overview and comparison of various interpolation methods, see Lam 1983, Eklundh and Mårtensson 1995, Carrara et al. 1997.)

Usually, the DEM resolution is much lower than scanning resolution, and this fact results in a loss of source information. Moreover, this procedure involves an unnatural raster → vector → raster conversion. Although vectorization algorithms are guaranteed to place vectors inside the raster lines, significant deviations may occur from medial lines (see figure 1).

Figure 1. Contour line distortions during vectorization.

To ensure maximum preservation of the information coded in source data, we abandon the vectorization part of the process: contours are thinned to get medial lines of one pixel width, and pixels between contours are computed using thin plate spline interpolation. The whole processing is performed with scanning resolution but, as we shall see, time and storage requirements are realistic owing to the power of today's computers. Our technology consists of the following steps:

1. *Scanning contour maps* with a resolution of 300–500 dpi.
2. *Co-ordinate transformation* of the raster image.
3. *Cleaning the image* by manual raster editing.
4. *Algorithmic thinning of lines*: only the pixels of the medial axis are retained. For an overview of thinning algorithms, see Lam et al. (1992).
5. *Algorithmic detection of line breaks and junctions* with interactive support to correct these anomalies. (Junctions normally arise at the merging points of two neighbouring contours. These junctions should be corrected in any case, but line breaks may remain in the final contour structure.)
6. *Providing an elevation value for each contour line* via a semi-automated process, supported by a connected component labelling algorithm (see e.g. Stefano and Bulgarelli 1999). Further

support is possible using some recognition techniques (Shimada et al. 1995, Katona and Hudra 1999). The result will be a digital contour matrix like that shown in figure 2.

7. *Assigning single elevation points*, when they are available in the source data (figure 2).

8. *Spatial interpolation* to compute elevation values for the raster points between contours (see pixels denoted by X in figure 2). To reduce computational costs, the final DEM may have lower resolution than the scanning resolution, depending on current application demands.

This paper concentrates on the problem of *spatial interpolation*. Section 2 sketches the mathematical background based on the approach by Terzopoulos (1983 and 1986). After, sections 3, 4 and 5 outline the multigrid relaxation principle with the application of our own reduction technique. In the final section we give some comments on practical applications.

Figure 2. Part of a digital contour matrix where X means unknown point, values 200, 240 and 280 are contour line points, and the element 253 represents a single elevation point.

2. Thin plate spline interpolation

There are several ways of performing spatial interpolation (Lam 1983). Here we discuss just the thin plate spline interpolation, as it can be considered to be one of the best approaches to the problem. This classical method has been applied in several systems like the ANUDEM program (Hutchinson 1989) and the Topogrid module of Arc/Info (ESRI 1994).

For the continuous case, we search for a terrain function $f(x, y)$ that gives predefined elevation values d_1, \dots, d_m at the points $(x_1, y_1), \dots, (x_m, y_m)$, that is, $f(x_i, y_i) = d_i$ for any $i = 1, \dots, m$. To get a sufficiently smooth terrain, we should minimize the 'surface energy' in some sense. Terzopoulos (1986) gives a general functional

$$E_r(f) = \iint \left[\sum_{i=0}^r \binom{r}{i} \left(\frac{\partial^r f}{\partial x^i \partial y^{r-i}} \right)^2 \right] dx dy$$

defining an energy value to any function f . This functional has two important special cases.

When $r = 1$, we get the *membrane model* expressing the small deflection energy of a membrane (f_x and f_y denote partial derivatives):

$$E_1(f) = \iint (f_x^2 + f_y^2) dx dy \quad (1)$$

Minimizing $E_1(f)$ gives a continuous surface that needs not have continuous partial derivatives, and tends to minimize the surface area. When applied to terrain modelling, the surface may break at contour lines, and closed contour loops (e.g. around mountain peaks) are covered by flat surfaces.

When $r = 2$, we get the *thin plate model* expressing the small deflection bending energy of a thin plate (f_{xx}, f_{xy} and f_{yy} are second order partial derivatives):

$$E_2(f) = \iint (f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2) dx dy \quad (2)$$

Minimizing $E_2(f)$ gives a continuous surface with continuous first order partial derivatives. As a consequence, no breaks appear at contours and mountain peaks have a realistic shape. Normally the thin plate model is used for terrain modelling, but it can be replaced by membrane at a priori given break-lines (Szeliski 1990), or the two models can be combined as $E(f) = 0.5E_1(f) + E_2(f)$ for point elevation data (Hutchinson 1988, 1989).

The question is how to find a terrain function f of minimum energy with the constraint $f(x_1, y_1) = d_1, \dots, f(x_m, y_m) = d_m$. Bookstein (1989) gives an analytical solution for a given class of functions, but this approach has very high computational complexity when applied on contour maps. Instead, we follow the finite element approach applied by many authors (see, for instance, Terzopoulos 1983). The function $f(x, y)$ is replaced by a matrix $Z[i, j]$ and each element of Z represents the average value of f on a grid square. Furthermore, elements of Z can be interpreted as variables for modelling the function f , and their values can be determined by relaxation (iteration).

The iteration masks (figure 3) can be derived from (1) and (2), respectively (for details see Terzopoulos 1983, Katona 2001). The interpolation process consists of the following steps:

1. Start with a digital contour matrix (figure 2) containing data points with known elevation values and unknown points.
2. Give a suitable initial value for each unknown point. For instance, any unknown point may get the value of the nearest defined point (figure 4). Such an initial matrix can be generated using a modified distance transform (Borgefors 1984, Katona 2001).
3. Perform relaxation (iteration) for several cycles. Given data points are kept fixed during relaxation.

Figure 3. Sample masks for Jacobi iteration: a) membrane, b) thin plate.

Note that uncertainty of source data can be modelled by assigning weights w_1, \dots, w_m to the data points $(x_1, y_1), \dots, (x_m, y_m)$, where a higher value of w_i means a more accurate measurement at the point (x_i, y_i) . In such a case, data points are not kept fixed during the relaxation process, but their values are corrected at each step according to their weights (Szeliski 1990). Nevertheless, examples in this paper were generated without such weights.

Figure 4. Initial terrain generated by a modified distance transform.

The membrane relaxation is fast, after 50 iterations we may get a satisfactory result. However, the thin plate process is much slower, several thousand iteration steps are needed to get a result like that in figure 8. Terzopoulos (1983) says that n^r iterations are required for an $n \times n$ matrix, where r is the degree of partial derivatives. To overcome the slow convergence of the thin plate procedure, the multigrid method is applied as discussed in the next section.

3. Speed-up by multigrid relaxation

The basic idea of the multigrid concept is as follows.

1. Create a q -reduced matrix R from the original digital contour matrix Z . A straightforward way of reduction, called *simple reduction*, is to divide Z into tiles of $q \times q$ pixels, and replace each tile in Z with a single pixel in R by averaging the data points. If a tile does not contain a data point, the corresponding reduced pixel will be undefined.

2. Give initial values to undefined elements of R somehow.

3. Perform relaxation for R in the way described in the previous section. Here the convergence is faster because of the reduced matrix size. The relaxation result will be denoted by R' .

4. Give initial values to undefined pixels in Z using the corresponding values of R' . In this way we get a much better initial matrix (figure 5) than that obtained in the previous section (figure 4).

5. Perform relaxation for Z . Due to the better initial matrix, only 20 to 50 iterations are required to get a good result.

Figure 5. Initial terrain obtained from an 8-reduced matrix.

If the above procedure is applied on a hierarchy of reduced matrices, we will have the *multigrid relaxation method*, which is widely used in image processing and in solving some differential equations (Brand 1982). To perform a *full multigrid process*, assume – for the sake of simplicity – that the Z matrix is of size $2^n \times 2^n$. A sequence of reduced matrices $R_0, \dots, R_{n-1}, R_n=Z$ is generated, where R_i (of size $2^i \times 2^i$) is computed with a reduction factor $q = 2^{n-i}$ ($i = 1, \dots, n$). The process starts by creating a single element matrix R_0 containing an average elevation value. In the i -th step the reduced matrix R_i is created from the original Z , but undefined elements of R_i get initial values from the iterated matrix R'_{i-1} . The final result will be given by the matrix R'_n .

Next we show that the creation algorithm of reduced matrices is crucial and it has a major influence on the surface quality if the number of required iteration steps is fixed.

4. The reduction problem

There are different ways of producing a q -reduced matrix R from a digital contour matrix Z . *Simple reduction methods*, like that described in point 1 of section 3 of this paper (or a similar one, used by Hutchinson 1989) give good results when applied for scattered data points. However, in the case of high resolution contour matrices the source data have an excessive number of data points along contours (oversampling) and no data between contours (undersampling), as emphasized by Weibel and Heller (1991). That is why a simple reduction may generate thick contours in R and the multigrid process produces flat areas (terraces) along contours (see figures 6 and 7 below and figure 16.3 in Mackey et al. 2000).

Simple reduction methods can be improved by *thresholding* as follows. Consider a $q \times q$ tile of Z , assume that there are k data points in it, and choose a threshold T_q between 0 and q^2 . When calculating the reduced pixel r for that tile, there are three distinct cases:

(i) If $k > T_q$, then r is calculated by averaging data points in the tile, and r is kept fixed during relaxation.

(ii) If $0 < k \leq T_q$, then r is also calculated by averaging, but it is allowed to vary during relaxation.

(iii) If $k = 0$, then r is treated as undefined. (It gets initial value from the previous reduction stage and is not kept fixed during relaxation.)

Figure 6. Result of a thin plate interpolation: full multigrid process using simple reduction, 50 iterations.

Figure 7. Histogram of elevation data in figure 6. The peaks represent terraces along contours.

The above thresholded method gives better results for contour lines (figures 8, 9), but it has two crucial problems:

- Single elevation points normally fall into case (ii) during reduction, and they are not kept fixed when iterating the reduced matrix. This may result in significant distortions at single elevation points (figure 10).

- A further problem is that longer contour breaks may occur in the reduced matrix. For instance, with the diagonal line in figure 11 in matrix Z , the choice $T_q = 1$ results in a fixed 2-pixel wide line in R while in the case of $T_q = 2$ the line is not fixed at all in R .

That is why we decided to develop our own reduction method, which will be discussed in the next section.

Figure 8. Result of a thin plate interpolation: full multigrid process with threshold $0.6 \cdot q$ for each reduction level, 50 iterations.

Figure 9. Histogram of elevation data in figure 8. The terraces have mostly been eliminated.

Figure 10. Incorrect result of the thresholded reduction at a single elevation point.

Figure 11. A critical contour line section with reduction factor $q = 4$.

5. The contour thinning reduction algorithm

The thinning method we applied was designed for terrain interpolation and it differs markedly from morphological line thinning approaches (Lam et al. 1992). Our method avoids oversampling (thick contours) and undersampling (vanishing longer line sections) in the reduced matrix, but it does not guarantee topology preservation, because this is not important in our case. On the other hand, the contour thinning reduction takes special care for single elevation points (figure 12), and works well even if several contour lines merge during reduction.

To realize our algorithm, a qualifier code is attached to each element of Z and R respectively, with four possible values: C0 denotes a free point the value of which will be calculated by relaxation; C1 denotes a weak contour point (defined later); C2 denotes a normal contour point; C3 identifies a single elevation point. Matrix elements with codes C1, C2 and C3 do not change during relaxation. The initial Z matrix contains only C0, C2 and C3 codes based on the source data.

Figure 12. Result of a full multigrid process with contour thinning reduction.

The single elevation point has been handled correctly (cf. figure 10).

The contour thinning reduction process consists of three main steps.

Step 1. A buffer zone of width $q/2$ is generated for points C2 and C3 in Z , the result being denoted by Z^+ (figure 13). Computation can be performed in linear time using the modified distance transform (see point 2 of section 2) with distance threshold $q/2$. (Here ‘linear time’ means $O(N)$ operations, where N is the number of pixels in Z .)

Step 2. An initial q -reduced matrix S is calculated from Z^+ as follows. Z^+ is tiled into $q \times q$ squares and the average of all defined elements is calculated for each tile. (If all the elements in the square are undefined then the reduced element will also be undefined.) Qualifier codes of reduced elements are determined using thresholds t_q and T_q : if k is the number of defined elements in a tile

of Z^+ , then the corresponding reduced element gets a qualifier code C0 if $k < t_q$, C1 if $t_q \leq k < T_q$ (weak contour point), C2 if $T_q \leq k$ (normal contour point) or C3 if the $q \times q$ square contains at least one element with code C3. (We choose $t_q = q(q+1)/2$, which is the number of defined pixels in a tile of Z^+ when a contour line crosses the tile according to figure 11. Similarly, $T_q = q(3q+2)/4$ is the number of defined pixels when a contour line crosses a tile at the main diagonal.)

Step 3. R is calculated from S in the following substeps (figure 14):

- Height values are moved from S to R , but the qualifier codes in R are set to C0.
- C3 codes are moved from S to R .
- Thinning: any code in S is set to C0 if it has a neighbour in R with code different from C0 (here just the four orthogonal neighbours are considered).
- Place a chessboard on matrix S , and move all the remaining C2 codes to R on white positions.
- Thinning as above.
- Move all the remaining C2 codes to R on black positions.
- Thinning as above.
- Move all the remaining C1 codes to R on white positions.
- Thinning as above.
- Move all the remaining C1 codes to R on black positions.

Figure 13. Example of a Z^+ matrix containing a contour line and a single elevation point with buffer zone ($q = 4$)

Figure 14. Qualifier codes in the S matrix (generated from Z^+ in figure 13) and the R matrix (calculated from S)

The above procedure results in reduced contours not wider than 1 pixel (even in the case of figure 11), and the reduced contours may have breaks of maximum 2 pixels (such short breaks do not cause any degradation of accuracy during relaxation). Formal proof of these properties is given in Katona (2001).

It is not hard to see that steps 1 and 2 require $O(N)$ operations (N is the number of pixels in Z), while each stage of step 3 can be computed with $O(n)$ operations (n is the number of pixels in reduced matrices S and R). Overall we can say that the whole contour thinning reduction procedure is linear in time.

6. Conclusions and application notes

A widely used spatial interpolation method, called multigrid relaxation, was applied to create a digital elevation model (DEM) from scanned contour line maps. The customary procedure was modified by adding a contour line thinning algorithm that reduces elevation bias around data contour heights, thereby improving DEM quality.

We generate the DEM with the scanning resolution to preserve the accuracy of source data. However, this high resolution DEM can be resampled at a lower resolution according to application demands, using *pixel model* or *lattice model* as defined by Wise (2000). Note that a lower resolution pixel DEM can be also obtained by stopping the multigrid process before it reaches the final resolution; in this way the computation time can be reduced drastically.

An earlier version of our approach was applied in a Hungarian project that produced the first DEM covering the whole country with a 10×10 metre resolution (Katona 1995). To overcome the problem of big computation time at high resolution processing, a special parallel processor was applied to speed up the relaxation (Katona 1992). We should emphasise here that the power of current computers makes it unnecessary to use any additional hardware. As an example, consider a 1:10 000 scale topographic map sheet of size 600×400 mm, scanned with resolution 12 dots per

mm (about 300 dpi). Storage requirements and processing time (40 iterations, 1 GHz Intel processor) are given in table 1. The maximum multigrid level corresponds to the scanning resolution, and accuracy here means the pixel size related to the paper map.

Table 1. DEM generation parameters for a map sheet of size 600×400 mm.

Considering the customary 0.2 mm accuracy of paper maps, the $max - 1$ multigrid level provides maximum utilization of the source information. However, even the $max - 3$ level gives a DEM resolution of 6.7 metres in the real world (for a scale of 1:10 000) which is quite sufficient for most practical applications. Note that Hutchinson and Gallant (2000) give another approach to determine the optimal DEM resolution based on a root mean square slope criterion.

Finally we should mention that our method can be supplemented with a drainage enforcement algorithm based on the approach of Hutchinson (1989), and predefined break-lines can also be handled to describe elevation and orientation discontinuities (Szeliski 1990). These issues are important in practical applications but they exceed the scope of this paper.

References

- BRAND, K., 1982, Multigrid bibliography. In *Multigrid Methods*, edited by W. Hackbusch and U. Trottenberg, Lecture Notes in Mathematics Vol. 960 (Springer-Verlag), pp. 631-650.
- BOOKSTEIN, F. L., 1989, Principal warps: thin-plate splines and the decomposition of deformations. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **11**, 567-584.
- BORGEFORS, G., 1984, Distance transformations in arbitrary dimensions. *Computer Vision, Graphics and Image Processing*, **27**, 321-345.

- CARRARA, A., BITELLI, G., and CARLA, R., 1997, Comparison of techniques for generation digital terrain models from contour lines. *Internat. Journal of Geographical Information Science*, **11**, 451-473.
- EKLUNDH, L., MÅRTENSSON, U., 1995, Rapid generation of Digital Elevation Models from topographic maps. *Internat. Journal of Geographical Information Systems*, **9**, 329-340.
- ESRI, 1994, *Arc/Info Version 7 – Arc Commands*. (Environmental System Research Institute, Inc.)
- HUTCHINSON, M. F., 1988, Calculation of hydrologically sound digital elevation models. In *Proceedings of the Third International Symposium on Spatial Data Handling*, Sydney, 17-19 August 1988 (Columbus, Ohio, International Geographical Union), pp. 117-133.
- HUTCHINSON, M. F., 1989, A new procedure for gridding elevation and stream-line data with automatic removal of spurious pits. *Journal of Hydrology*, **106**, 211-232.
- HUTCHINSON, M. F., and GALLANT, J. C., 2000, Digital elevation models and representation of terrain shape. In *Terrain Analysis – Principles and Applications*, edited by J. P. Wilson and J.C. Gallant (Wiley & Sons, New York), pp. 29-50.
- KATONA, E., 1992, Cellular processing. In *Fuzzy, Holographic and Parallel Intelligence*, edited by B. Soucek and the IRIS Group (Wiley & Sons, New York), pp.215-230.
- KATONA, E., 1995, Digital terrain modelling by multigrid relaxation. *Geodézia és Kartográfia* 1995/5, pp. 20-25 (in Hungarian).
- KATONA, E., and HUDRA, GY., 1999, An interpretation system for cadastral maps. In *Proceedings of 10th Internat. Conf. on Image Analysis and Processing* (IEEE Computer Society), pp. 792-797.
- KATONA, E., 2001, *Automatic map interpretation*. Ph.D. Thesis, Department of Physical Geography, University of Szeged (in Hungarian).
- LAM, S.-N., 1983, Spatial interpolation methods – a review. *The American Cartographer*, **10**, 129-149.

- LAM, L., LEE, S. W., and SUEN, C. Y., 1992, Thinning methodologies – a comprehensive survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **14**, 869-887.
- MACKAY, B. G., MULLEN, I. C., BALDWIN, K. A., GALLANT, J. C., SIMS, R. A., and MCKENNEY, D. W., 2000, Towards a Spatial Model of Boreal Forest Ecosystems: The Role of Digital Terrain Analysis. In *Terrain Analysis – Principles and Applications*, edited by J. P. Wilson and J.C. Gallant (Wiley & Sons, New York), pp. 391-422.
- SHIMADA S., MARUYAMA K., MATSUMOTO A., and HIRAKI K., 1995, Agent-based Parallel Recognition Method of Contour Lines. In *Proceedings of Internat. Conf. on Document Analysis and Recognition* (IEEE Press, Los Alamitos, California), pp. 154-157.
- STEFANO, L. and BULGARELLI, A., 1999, A Simple and Efficient Connected Components Labelling Algorithm. In *Proceedings of 10th Internat. Conf. on Image Anal. and Processing* (IEEE Press), pp. 322-327.
- SZELISKI, R., 1990, Fast Surface Interpolation Using Hierarchical Basis Functions. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **12**, 513-528.
- TERZOPOULOS, D., 1983, Multilevel Computational Processes for Visual Surface Reconstruction. *Computer Vision, Graphics and Image Processing*, **24**, 52-96.
- TERZOPOULOS, D., 1986, Regularization of Inverse Visual Problems Involving Discontinuities. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **8**, 413-423.
- WEIBEL, R., and HELLER, M., 1991, Digital Terrain Modelling. In *Geographical Information Systems, Vol. 1, Principles*, edited by D. J. Maguire, M. F. Goodchild and D. Rhind (Wiley & Sons, New York), pp. 269-297.
- WISE, S., 2000, GIS data modelling – lessons from the analysis of DTMs. *Internat. Journal of Geographical Information Science*, **14**, 313-318.

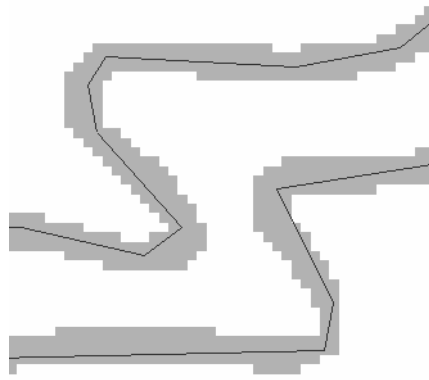


Figure 1. Contour line distortions during vectorization.

```

x   x  200  x   x  240  x   x   x   x
x  200  x   x   x  240  x   x   x   x
x  200  x   x   x  240  x   x   x  280
200  x   x   x  240  x   x   x  280  x
x   x   x  240  x   x   x  280  x   x
x  240 240  x   x   x  280  x   x   x
240  x   x   x   x   x  280  x   x   x
x   x   x  253  x   x   x  280  x   x
x   x   x   x   x   x   x   x  280 280

```

Figure 2. Part of a digital contour matrix where X means unknown point, values 200, 240 and 280 are contour line points, and the element 253 represents a single elevation point.

$$\text{a) } \frac{1}{4} \begin{bmatrix} & & 1 & & \\ & 1 & & & \\ 1 & 0 & 1 & & \\ & & & 1 & \\ & & & & \end{bmatrix}$$

$$\text{b) } \frac{1}{32} \begin{bmatrix} & & & & -1 & & \\ & & & & -2 & 8 & -2 \\ & & & & -1 & 8 & 12 & 8 & -1 \\ & & & & -2 & 8 & -2 & & \\ & & & & & & & & -1 \end{bmatrix}$$

Figure 3. Sample masks for Jacobi iteration: a) membrane, b) thin plate.

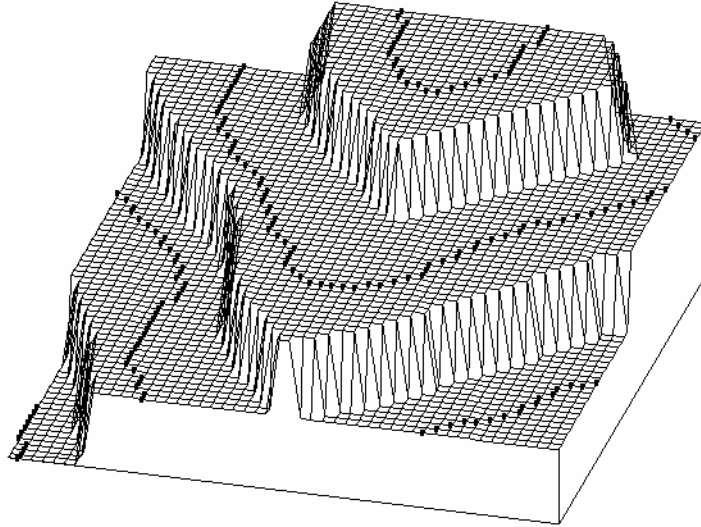


Figure 4. Initial terrain generated by a modified distance transform.

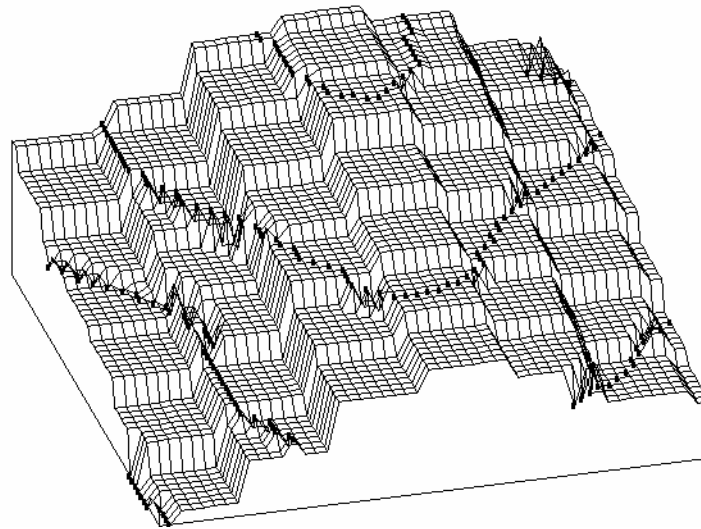


Figure 5. Initial terrain obtained from an 8-reduced matrix.

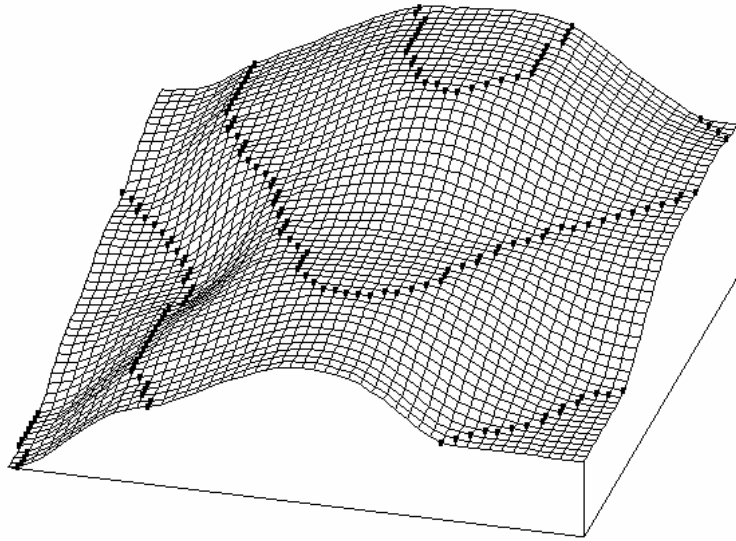


Figure 6. Result of a thin plate interpolation: full multigrid process using simple reduction, 50 iterations.

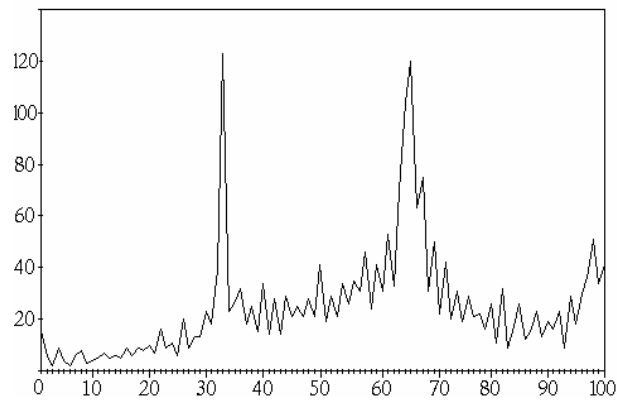


Figure 7. Histogram of elevation data in figure 6. The peaks represent terraces along contours.

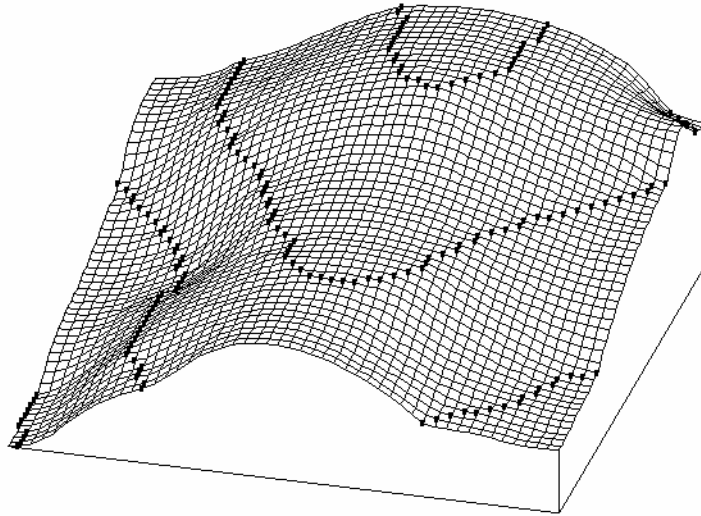


Figure 8. Result of a thin plate interpolation: full multigrid process with threshold $0.6 \cdot q$ for each reduction level, 50 iterations.

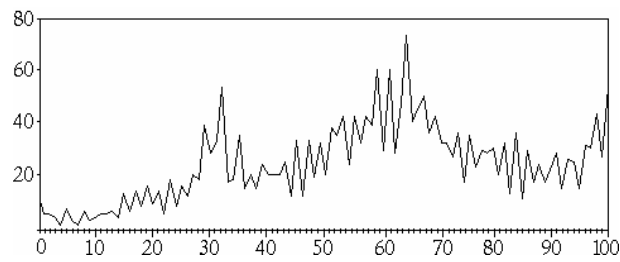


Figure 9. Histogram of elevation data in figure 8. Terraces have mostly been eliminated.

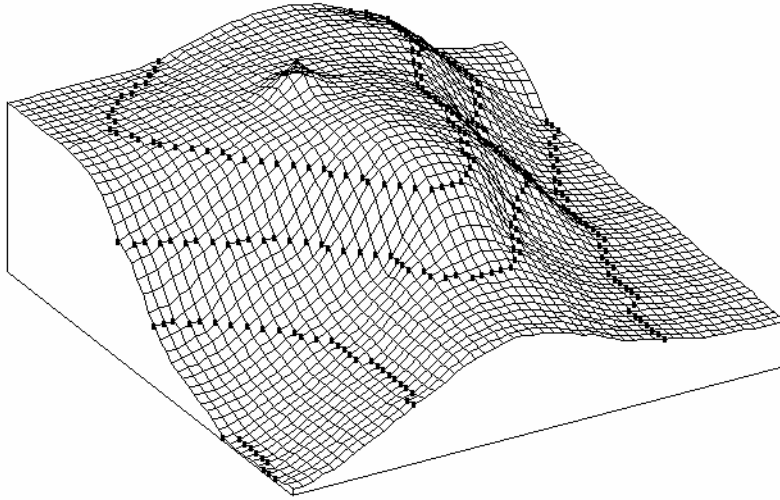


Figure 10. Incorrect result of the thresholded reduction at a single elevation point.

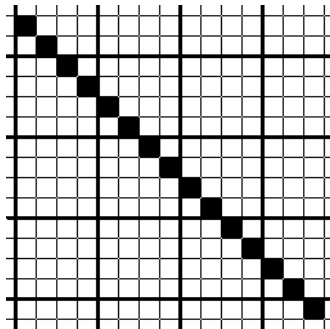


Figure 11. A critical contour line section with reduction factor $q = 4$.

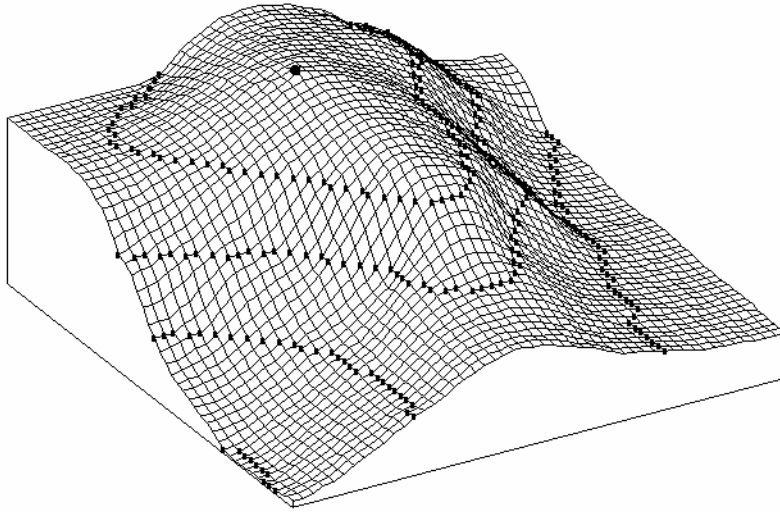


Figure 12. Result of a full multigrid process with contour thinning reduction.

The single elevation point has been handled correctly (cf. figure 10).

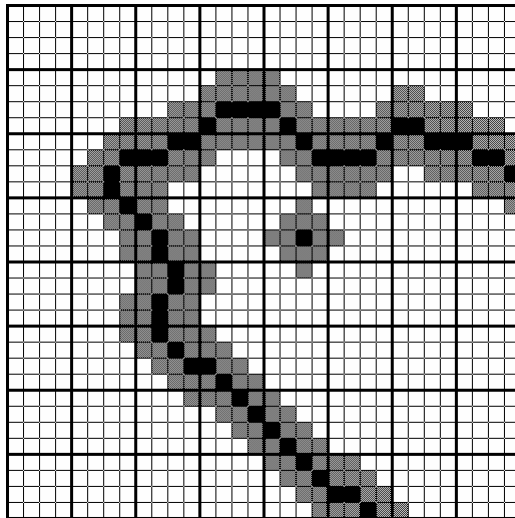


Figure 13. Example of a Z^+ matrix containing a contour line and a single elevation point with buffer zone ($q = 4$)

			C2	C1		C1	
	C1	C2		C1	C2	C1	C2
		C1		C3			
		C2					
		C1	C1				
			C1	C1			
				C1	C1		

Reduced matrix S

			C2			C1	
		C2			C2		C2
				C3			
		C2					
			C1				
				C1			
					C1		

Reduced matrix R

Figure 14. Qualifier codes in matrix S (generated from Z^+ in figure 13) and matrix R (calculated from S)

<i>Multigrid level</i>	<i>Accuracy</i>	<i>Storage</i>	<i>Running time</i>
maximum	0.08 mm	132 MB	8.4 min.
max – 1	0.17 mm	33 MB	2.2 min.
max – 2	0.33 mm	8.3 MB	35 seconds
max – 3	0.67 mm	2.1 MB	12 seconds

Table 1. DEM generation parameters for a map sheet of size 600 × 400 mm.