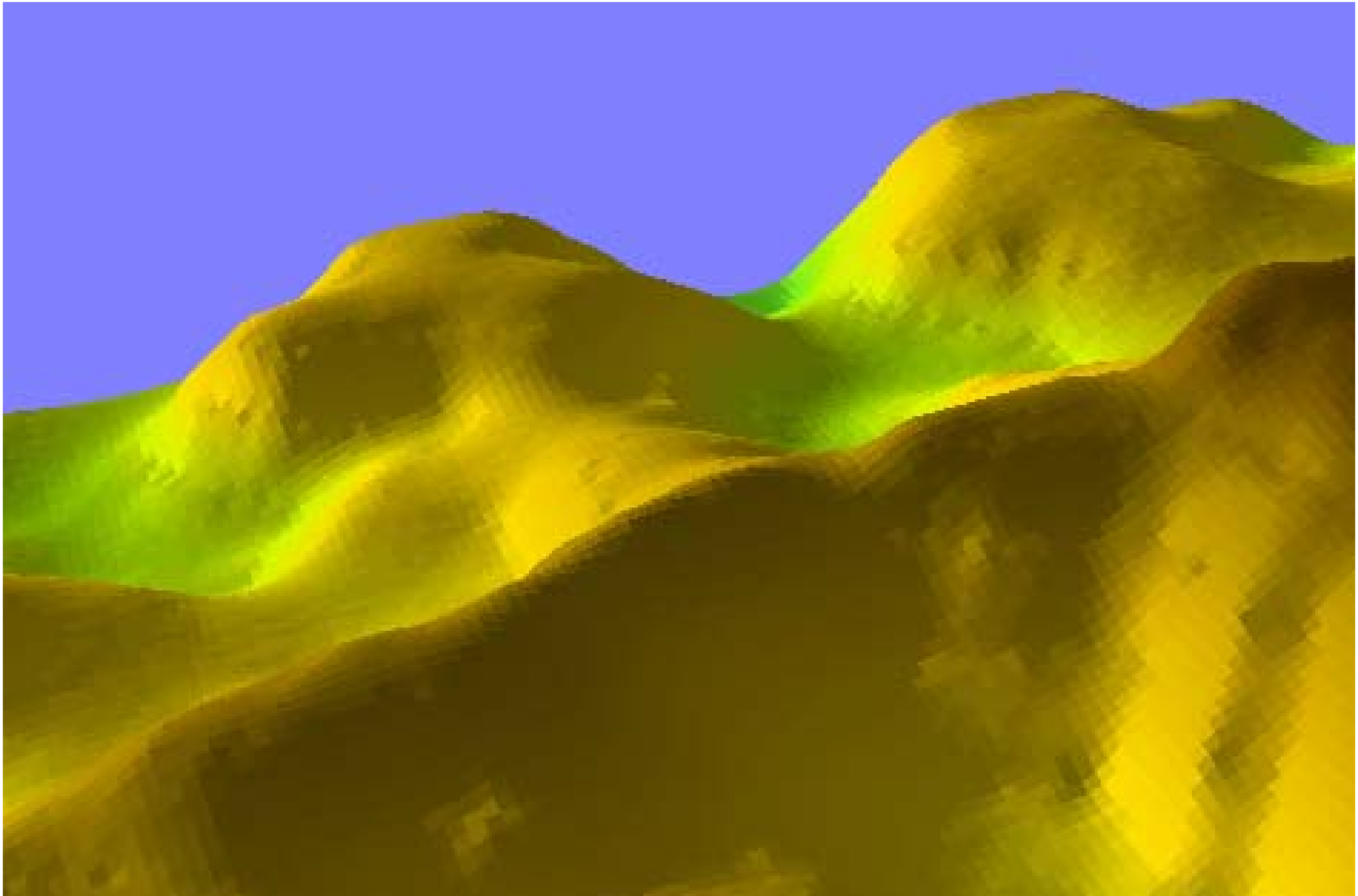
DIGITAL TERRAIN MODELLING

Endre Katona
University of Szeged
Department of Informatics
katona@inf.u-szeged.hu

The problem:

- data sources
- data structures
- algorithms



DTM = Digital Terrain Model

Terrain function: *h*(*x*, *y*) with continuous partial derivatives, excepting some special cases:

– the function is not continuous (bench).

– partial derivatives are not continuous (breakline).

2.5 dimensional modeling: not suitable for caves, for instance.

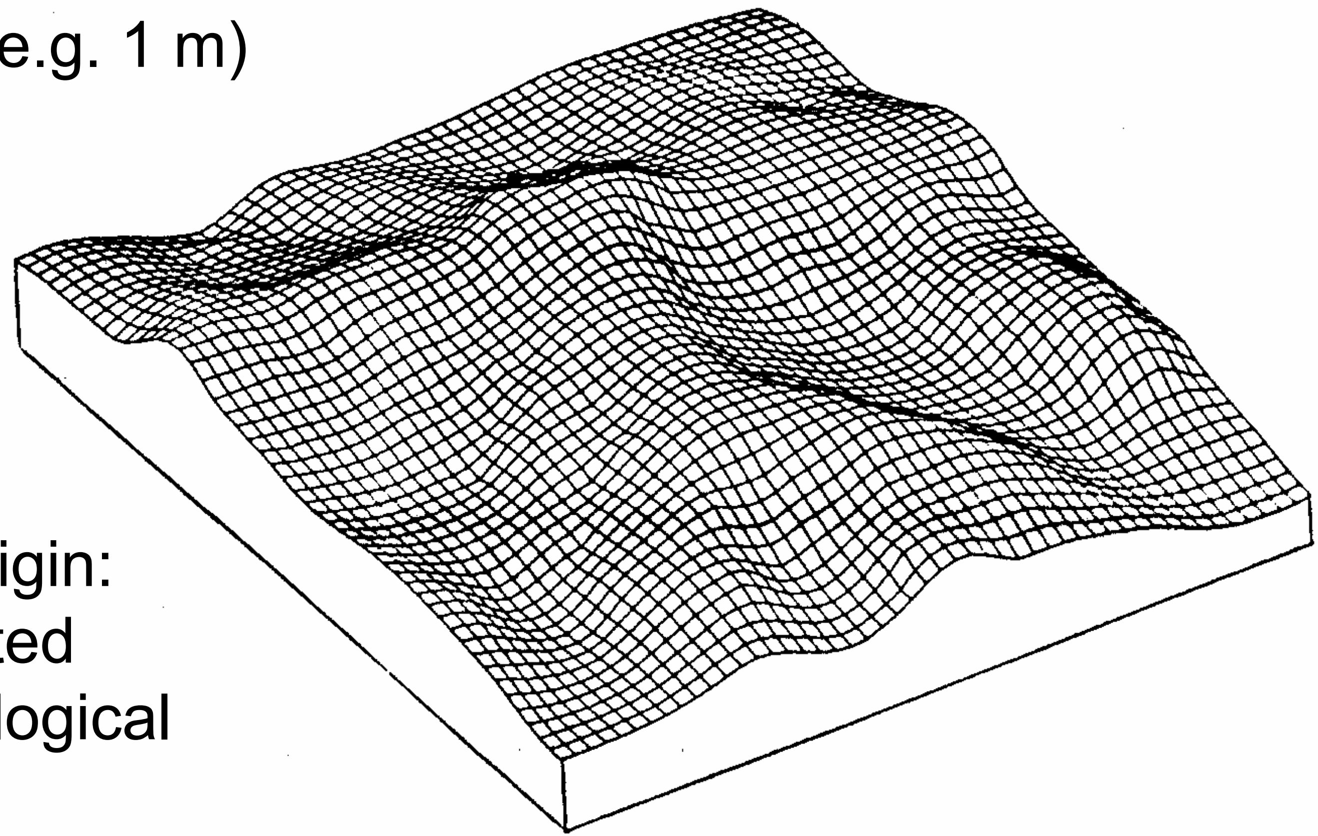
Model requirements:

- good approximation of the real world
- to determine h for any (x, y)

DEM = Digital Elevation Model

Raster model: matrix of height values

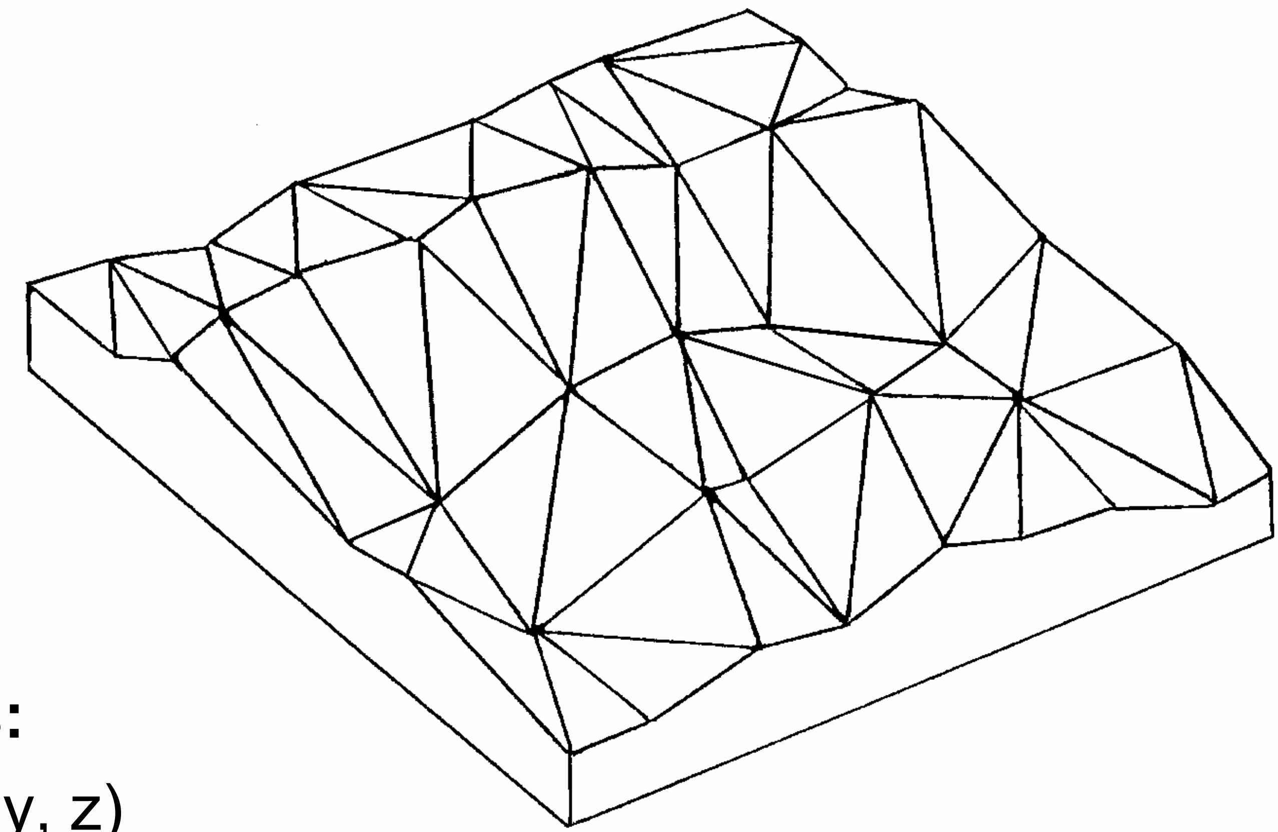
- resolution (e.g. 20 m)
- accuracy (e.g. 1 m)



Note: DEM origin: USGS (United States Geological Survey)

TIN = Triangulated Irregular Network

Vector model



Data structures:

a) NODE (id, x, y, z)

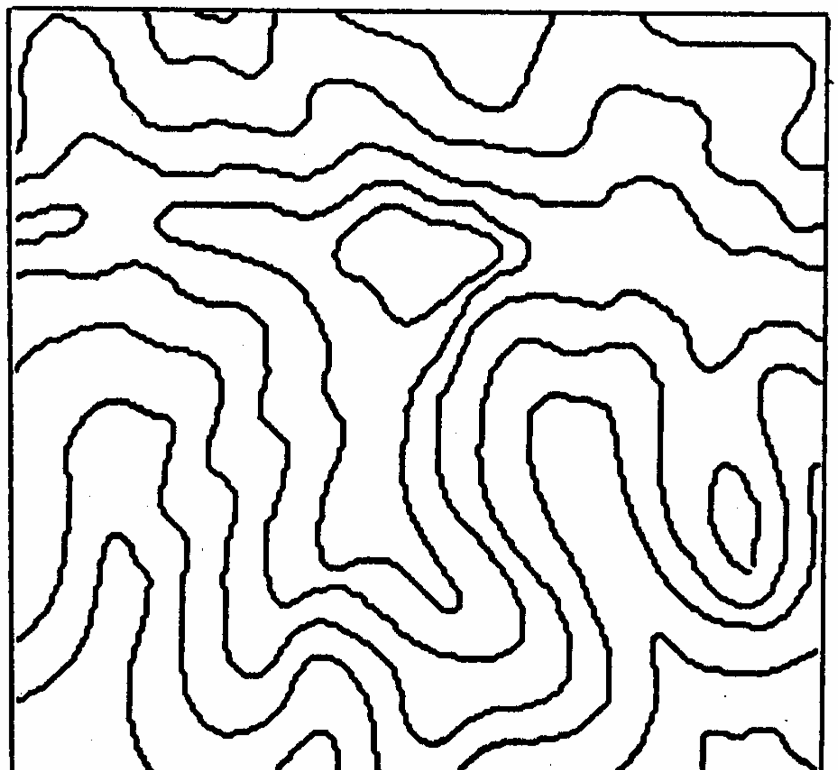
TRIANGLE (id, $node_1$, $node_2$, $node_3$, tr_1, tr_2, tr_3).

b) NODE (id, x, y, z, $node_1$, ..., $node_n$)

Contour line (level line) representation

Vector approach:

- LINE x_1, y_1, ..., x_n, y_n, z (2D line string with height value z)
- LINE x_1, y_1, z_1, ..., x_n, y_n, z_n (3D line sting with z_1 = ... = z_n)



DEM versus TIN

DEM:

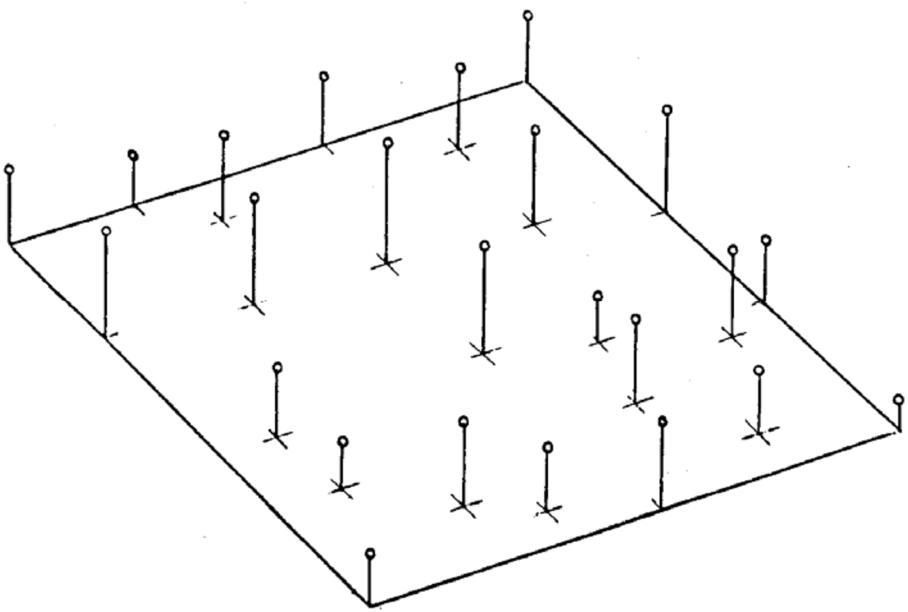
- simple data structure
- easier analysis
- high accuracy at high resolution
- high memory demand
- time-consuming processing

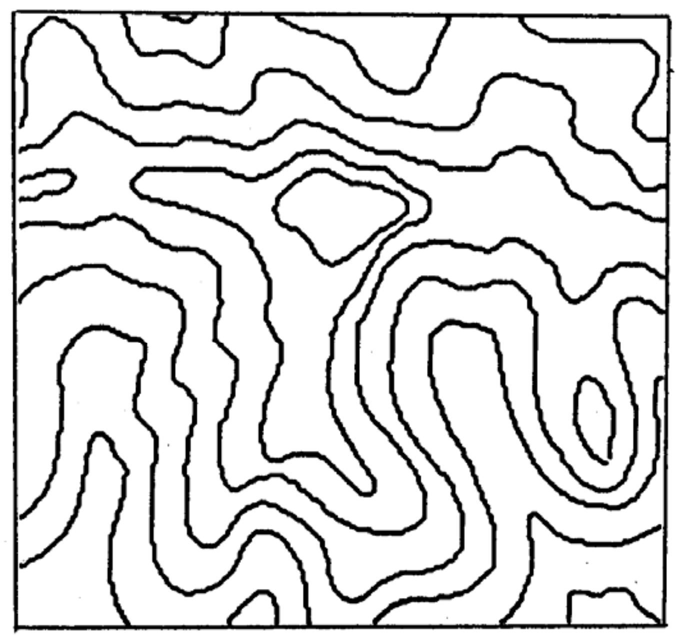
TIN:

- restricted accuracy
- complex algorithms
- less memory required
- time-efficient processing

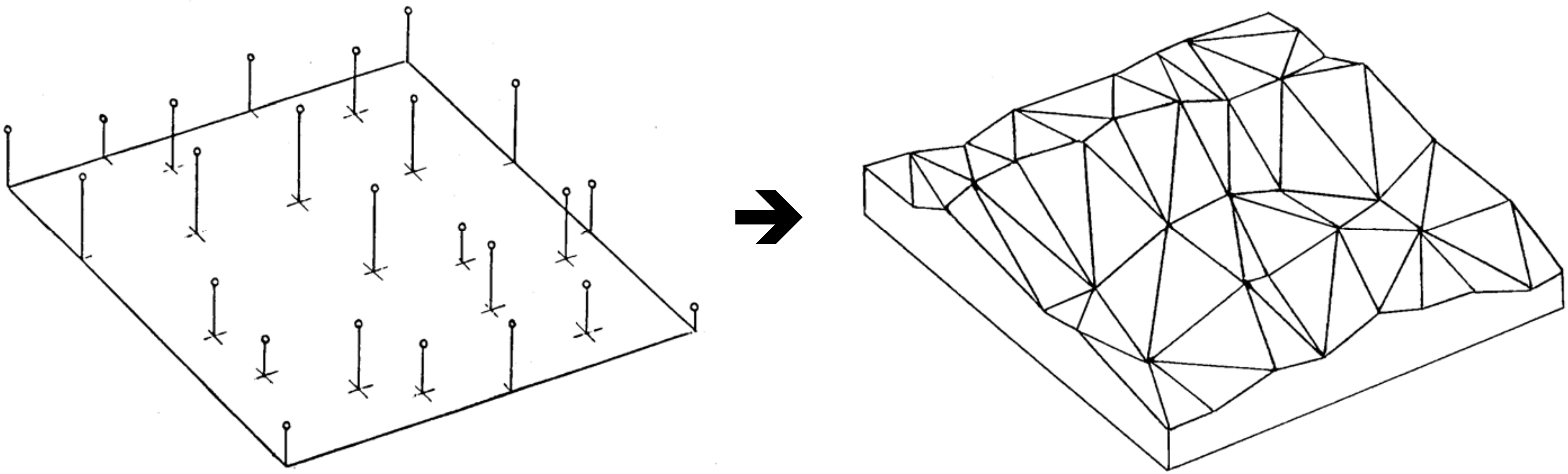
Data sources for DTM

1. Stereo aerial photos (photogrammetry)
2. Measured height values
3. Existing contour line maps





Generating TIN from height values

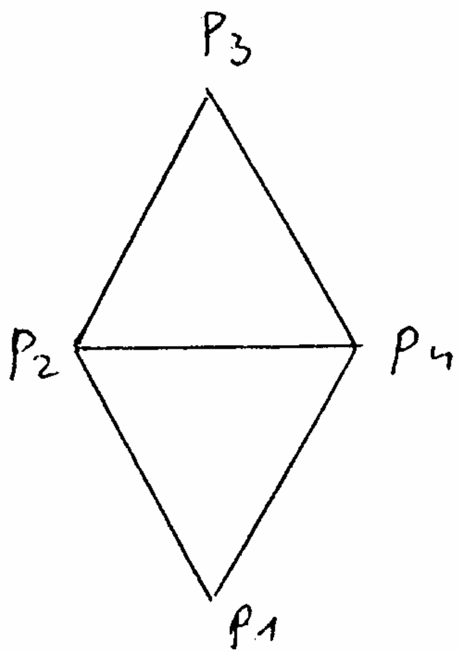


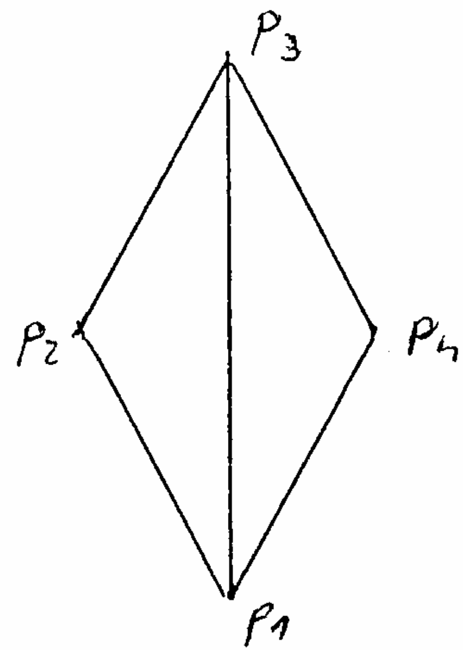
a) Direct triangulation.

*b) S*patial interpolation and triangulation.

Delaunay-triangulation - 1

- Given: a set of 3D nodes (x, y, z)
- Reduction to 2D: instead (x, y, z) we take (x, y).
- Prefer "fat" triangles.



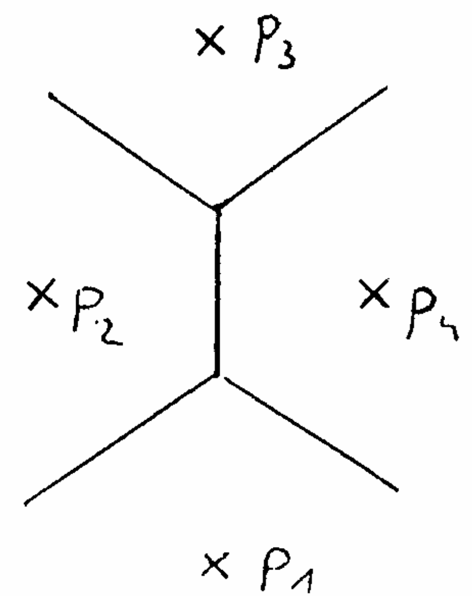


Delaunay-triangulation - 2

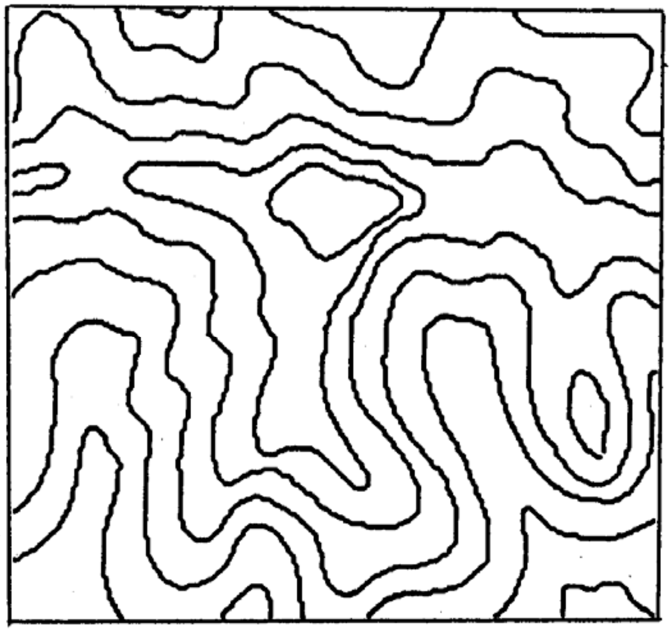
Delaunay triangle**:** the circumscribing circle does not contain further node.

Delaunay-triangulation**:** each triangle is a Delaunay-triangle.

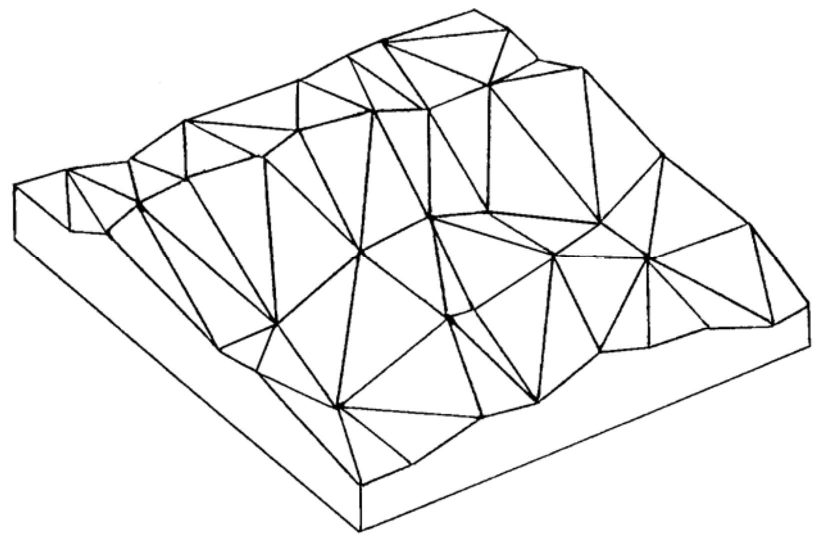
Voronoi diagram: a set of disjoint territories. Each node has a territory. Each point in the plane is classified into the territory of the closest node. Nodes with neighboring territories can be connected by an edge.



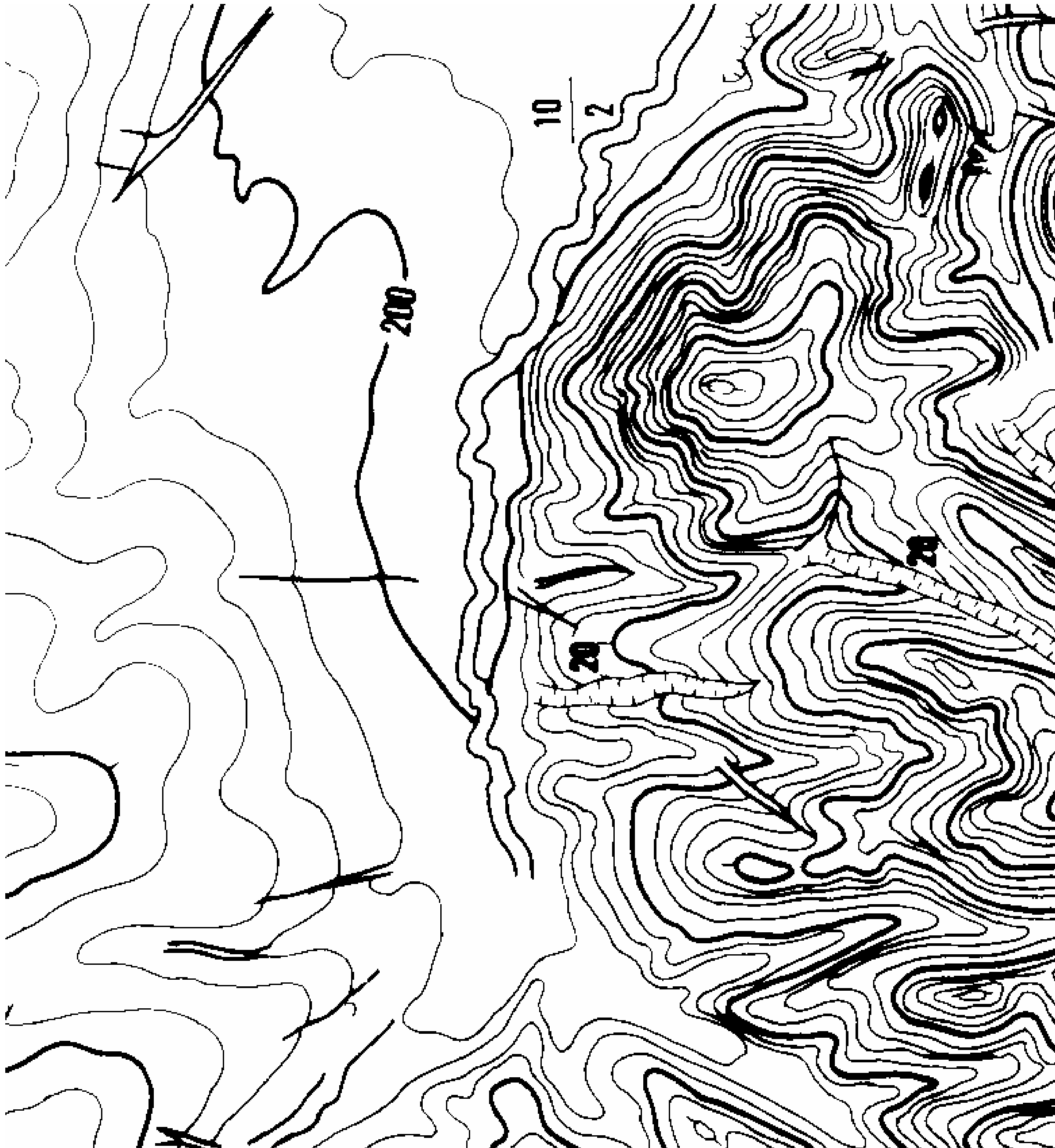
Generating TIN from contour line maps



➔



Part of a scanned contour line map (before processing)



Generating TIN from contour lines - 2

Processing steps:

1. Scanning the contour line map sheet.
2. Manual correction (eliminating gaps and junctions, handle special notations).
3. Vectorization (manual or automatic), we get a set of nodes and edges as a result.
4. Assigning a height value to each contour line (manual or half-automatic).
5. Assigning triangles between contour lines (Delaunay-algorithm).

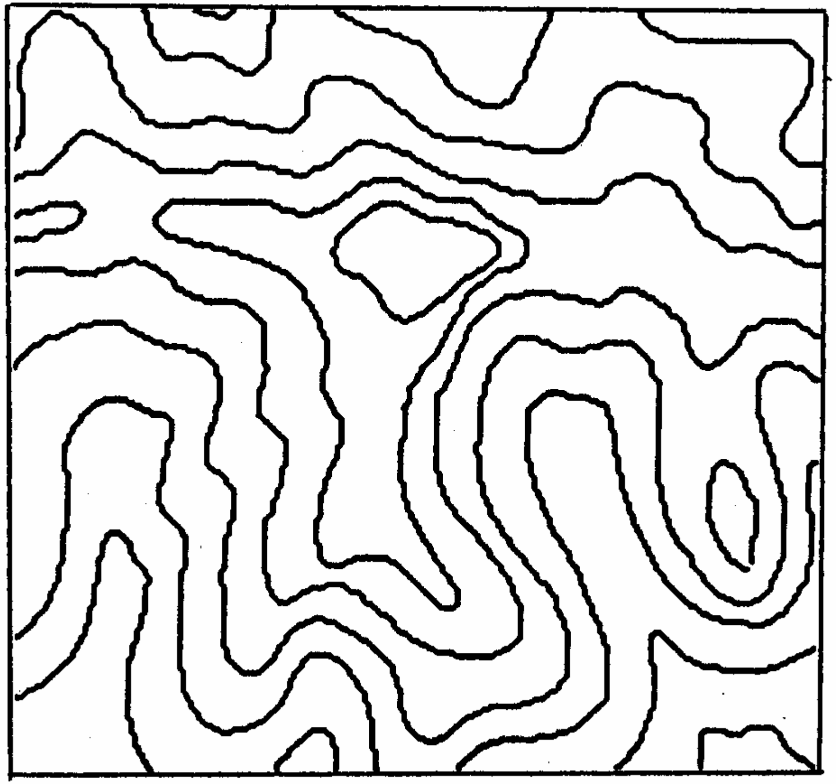
Generating TIN from contour lines - 3

Problem: flat areas

- at mountain peaks,
- at ridges.

Solution:

spatial interpolation.



Spatial interpolation

Consider a terrain function $f(x, y)$.

Given: $f(x_1, y_1) = h_1, \ldots, f(x_m, y_m) = h_m$

Problem: estimating $f(x, y)$ in other points.

Solutions:

- Inverse distance weighted moving average
- Polynomial interpolation

Inverse distance weighted moving average

Given: height values $h_1, ..., h_m$ at points $P_1, ..., P_m$

Unknown: height value h of a given point P.

Estimation: $h = (h_1/d_1 + ... + h_m/d_m) / (1/d_1 + ... + 1/d_m)$

where d_i is the distance between P_i and P.

Properties:

- Good for ridges.
- Flat areas at peaks.
- Local maxima and minima may occur only at given points.

Polynomial interpolation

Given: $f(x_1, y_1) = h_1, ..., f(x_m, y_m) = h_m$

Task: approximate $f(x, y)$ with a polynomial $p(x, y)$ of degree r. For example, if $r = 2$:

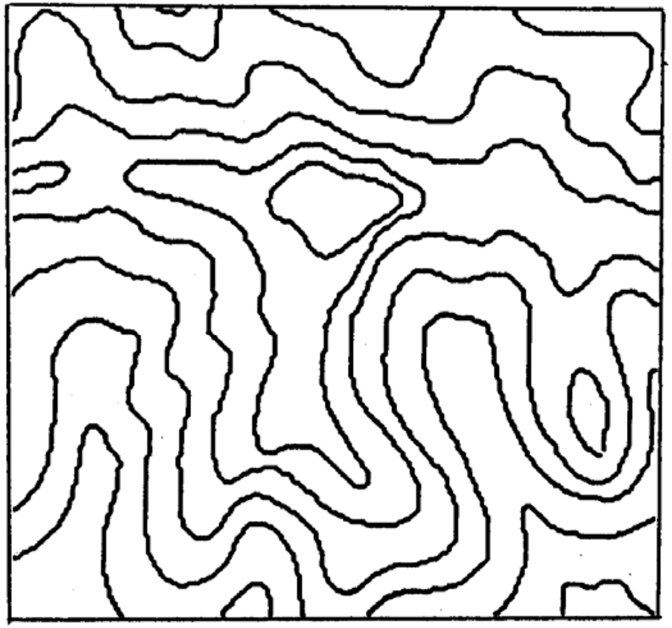
$p(x, y) = a_{00} + a_{10}x + a_{01}y + a_{20}x^2 + a_{11}xy + a_{02}y^2$

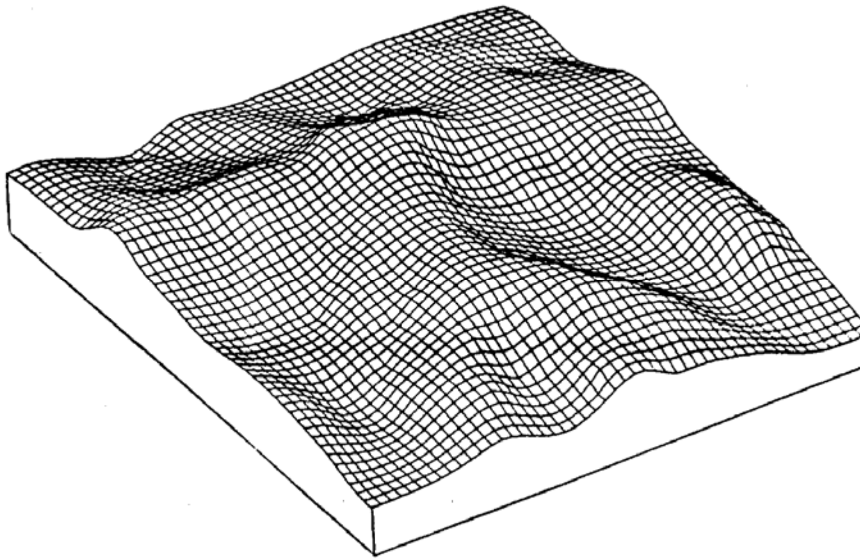
Solution: Coefficients $a_{i,j}$ are determined by least squares method: $E = \Sigma_i (p(x_i, y_i) - h_i)^2 \rightarrow \min$.

Properties:

- Local maxima or minima may occur ***not*** only at given points.
- Expensive procedure for contour lines (too many given points)
- Physical terrain features are not considered.

Generating DEM from contour line maps



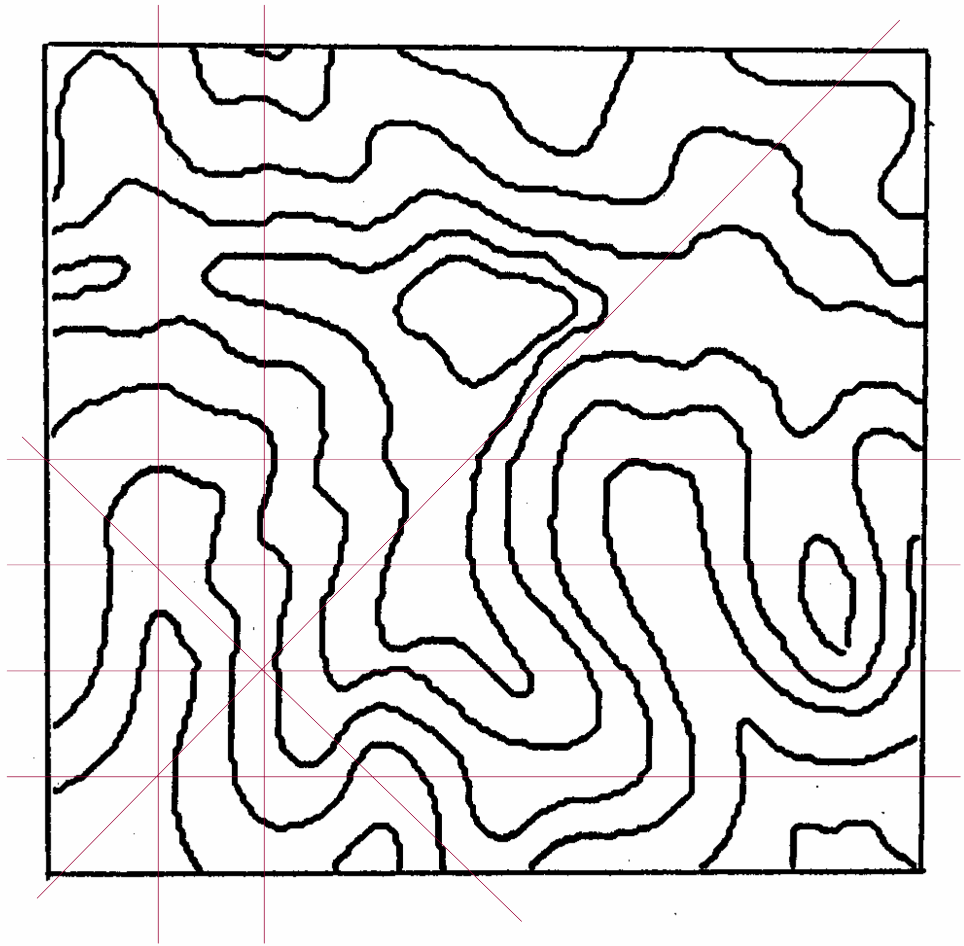


Solutions:

- The Intercon method
- Variational spline interpolation

The Intercon method (IDRISI)

- Place a regular grid on contours.
- Create cross-sections along horizontal, vertical and diagonal lines of the grid.
- Calculate height and slope values for each grid point (by linear interpolation).
- Heuristics: choose the height value belonging to the maximum slope.



The Intercon method - 2

Disadvantages:

- Local maxima and minima may occur only at given points.
- Interrupted contour lines may cause significant distortions.
- Single elevation points cannot be handled.

Variational spline interpolation

Source data:

- contour line map, and/or
- a set of elevation points.

Target data: DEM matrix

Preprocessing of contour lines:

- Scanning of contour line map sheets.
- Manual editing of the scanned raster image.
- Contour line thinning.
- Assigning height values to contour lines.

Variational spline interpolation - 2

Initial DEM matrix (X denotes unknown point):

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

Variational spline interpolation - 3

Continuous case

Given: $f(x_1, y_1) = d_1, \ldots, f(x_m, y_m) = d_m$

Task: approximate $f(x, y)$ with a minimum energy function.

General energy functional of degree* r*:

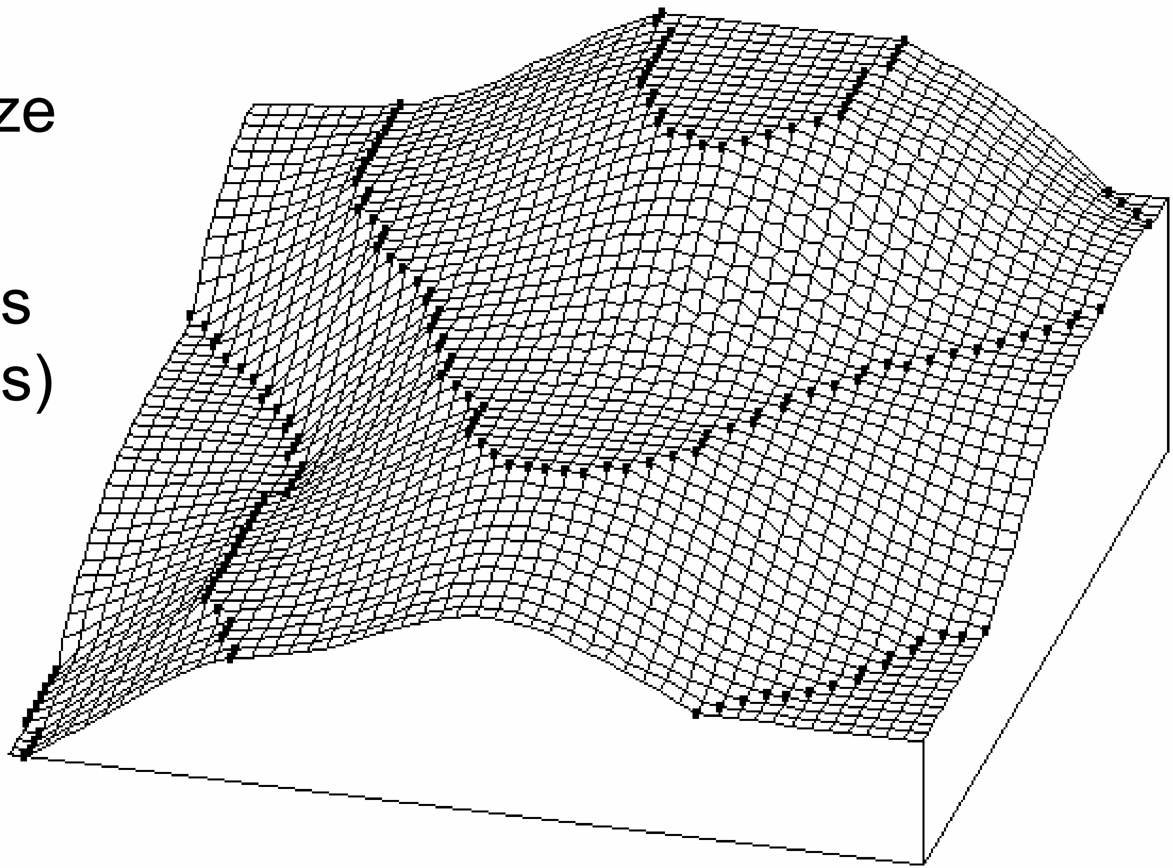
$$E_r(f) = \iint \left[\sum_{i=0}^{r} \binom{r}{i} \left(\frac{\partial^r f}{\partial x^i \partial y^{r-i}} \right)^2 \right] dx\, dy$$

Variational spline interpolation - 4

Membrane model: $$E_f^1 = \iint \left(f_x^{\,2} + f_y^{\,2} \right) dx\, dy$$

It tends to minimize the surface area.

(Partial derivatives are not continuous)



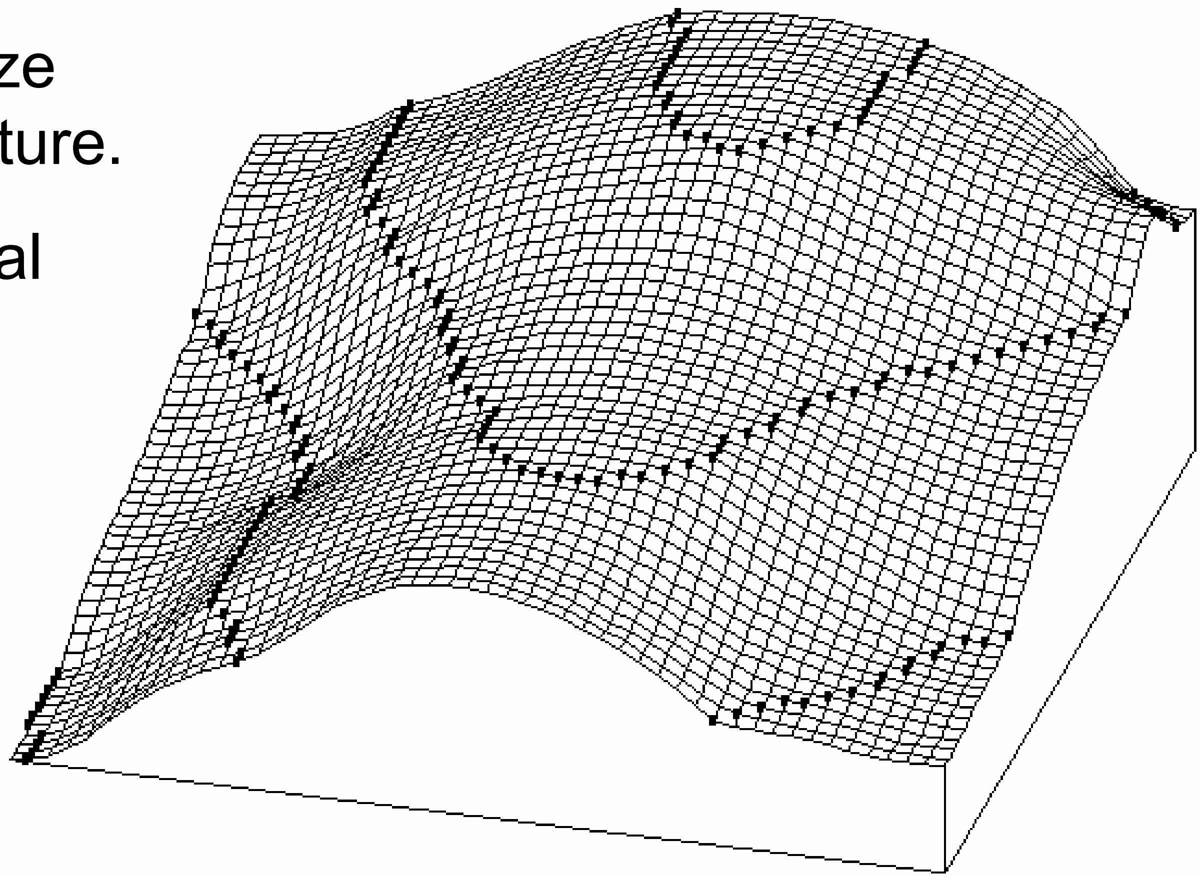
Variational spline interpolation - 5

Thin plate model:

$$E_f^2 = \iint \left(f_{xx}^2 + 2 f_{xy}^2 + f_{yy}^2 \right) dx\, dy$$

It tends to minimize the surface curvature.

(Continuous partial derivatives)



Variational spline interpolation - 6

Membrane model

Continuous case: $E_f^1 = \iint \left(f_x^2 + f_y^2 \right) dx\, dy$ ➔ min

Discrete case (matrix Z instead of function $f(x,y)$)**:**

$E_Z{}^1 = \Sigma_i \Sigma_j [(z_{i,j+1} - z_{i,j})^2 + (z_{i+1,j} - z_{i,j})^2]$

To find the minimum of $E_Z{}^1$, take the partial derivatives for any i, j:

$\partial E_Z{}^1 / \partial z_{i,j} = 8z_{i,j} - 2z_{i+1,j} - 2z_{i-1,j} - 2z_{i,j+1} - 2z_{i,j-1} = 0$

Variational spline interpolation - 7

After dividing by 2:

$4z_{i,j} - z_{i+1,j} - z_{i-1,j} - z_{i,j+1} - z_{i,j-1} = 0$ for any *i, j*.

➔ Linear equation system.

Solution with Jacobi-iteration ➔ repeated convolution with mask

	1/4	
1/4	0	1/4
	1/4	

Note: Given points are kept fixed during convolution.

Variational spline interpolation - 8

Thin plate model

Continuous case: $E_f^2 = \iint \left(f_{xx}^2 + 2 f_{xy}^2 + f_{yy}^2 \right) dx\, dy$ ➔ min

Discrete case (matrix Z instead of function $f(x,y)$):

$$E_Z^2 = \Sigma_i \Sigma_j [(z_{i+1,j} - 2z_{i,j} + z_{i-1,j})^2 +$$
$$+ 2(z_{i+1,j+1} - z_{i,j+1} - z_{i+1,j} + z_{i,j})^2 + (z_{i,j+1} - 2z_{i,j} + z_{i,j-1})^2]$$

Variational spline interpolation - 9

To find the minimum of E_Z^2, take the partial derivatives:

$$\partial E_Z^2/\partial z_{i,j} = 2z_{i+2,j} + 4z_{i+1,j+1} - 16z_{i+1,j} + 4z_{i+1,j-1} +$$
$$+ 2z_{i,j+2} - 16z_{i,j+1} + 40z_{i,j} - 16z_{i,j-1} + 2z_{i,j-2} +$$
$$+ 4z_{i-1,j-1} - 16z_{i-1,j} + 4z_{i-1,j+1} + 2z_{i-2,j} = 0$$

➔ Linear equation system.

In a more expressive form

(after dividing by 2):

$$\begin{bmatrix} & & 1 & & \\ & 2 & -8 & 2 & \\ 1 & -8 & 20 & -8 & 1 \\ & 2 & -8 & 2 & \\ & & 1 & & \end{bmatrix}$$

Variational spline interpolation – 10

Solution with Jacobi-iteration ➔ repeated convolution with mask

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

Notes:

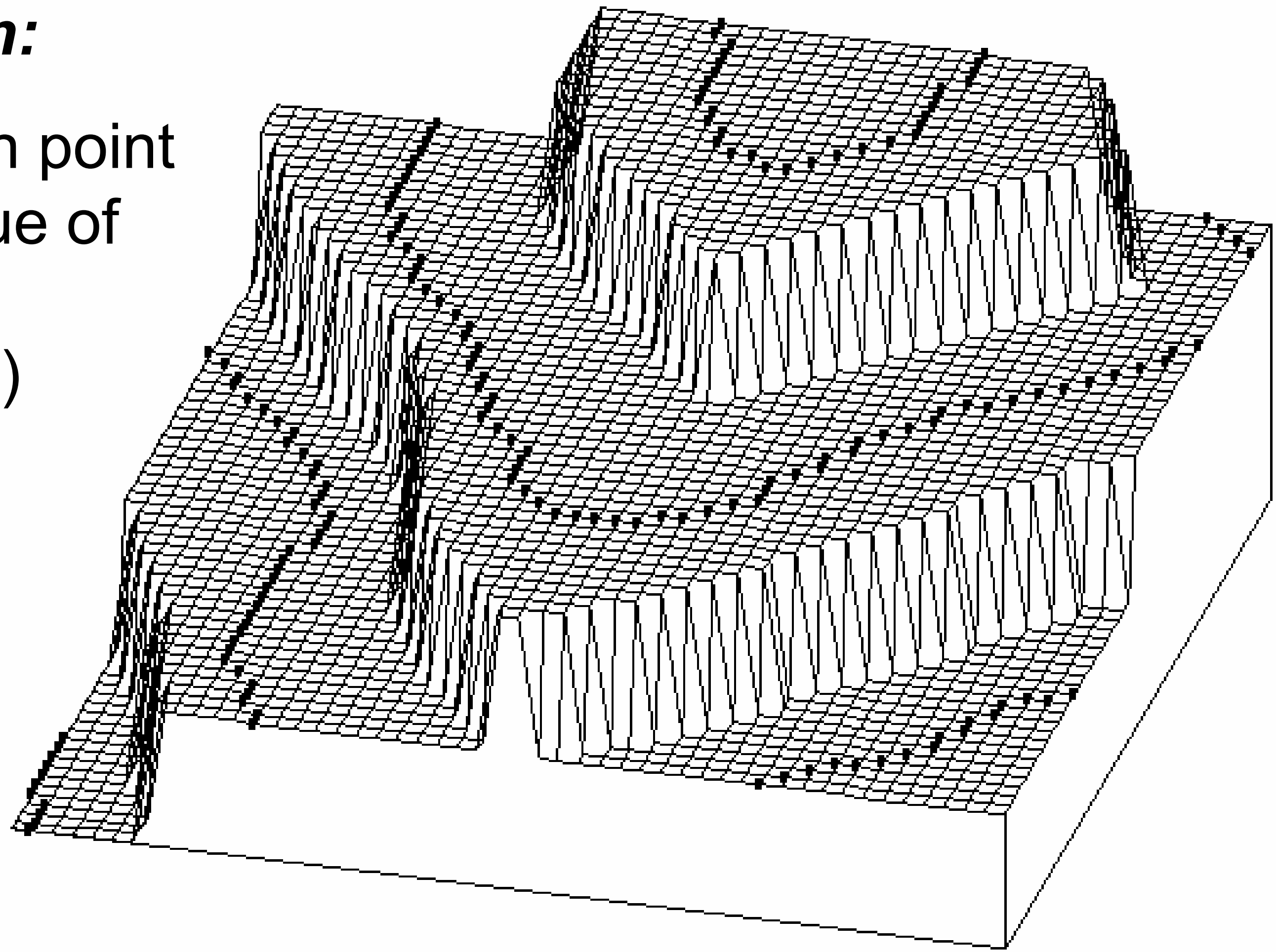
- Given points are kept fixed during convolution.
- Convergence of masks can be studied by Fourier-analysis.

Variational spline interpolation - 11

Sample initial matrix for the iteration:

(Each unknown point gets the value of the nearest contour line.)

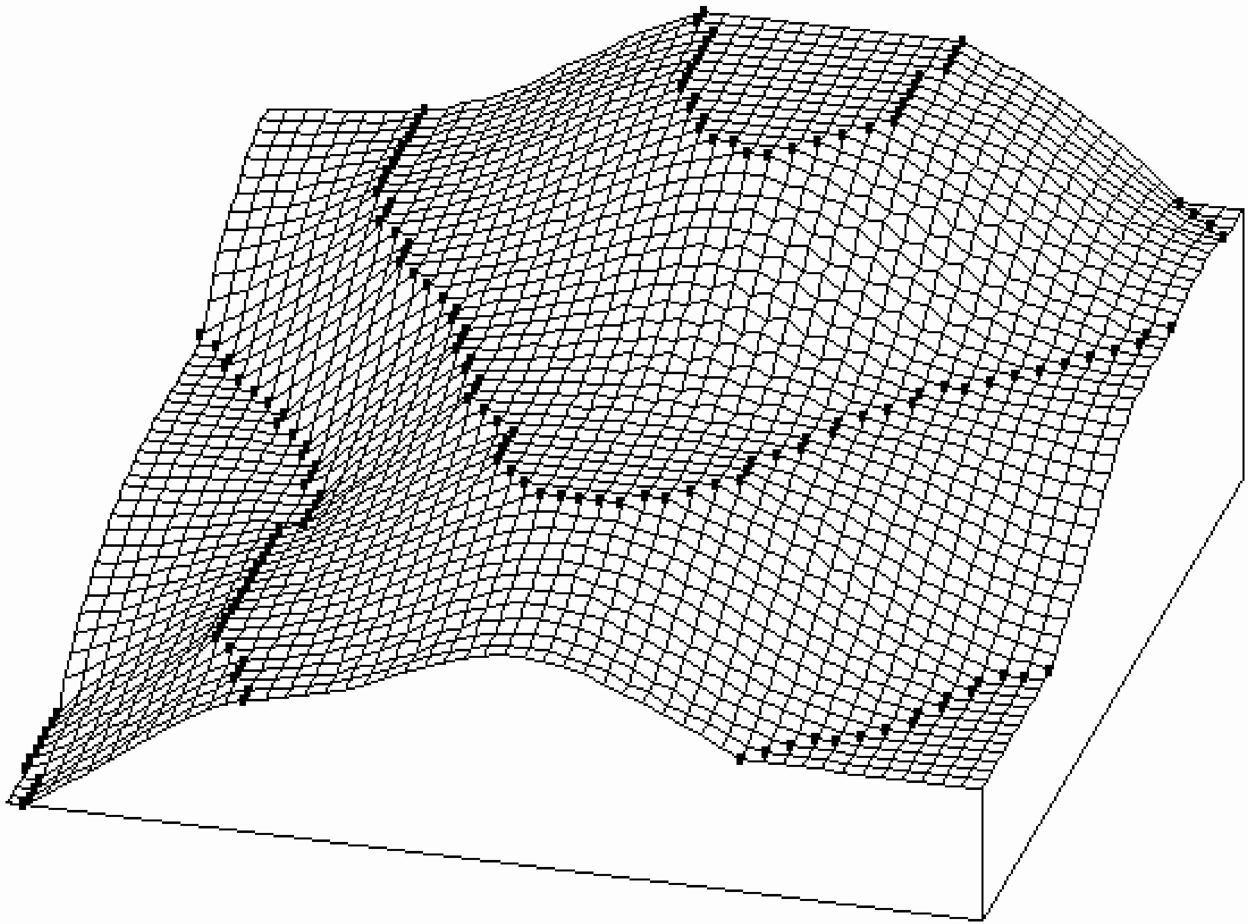
Algorithm: modified distance transform



Variational spline interpolation - 12

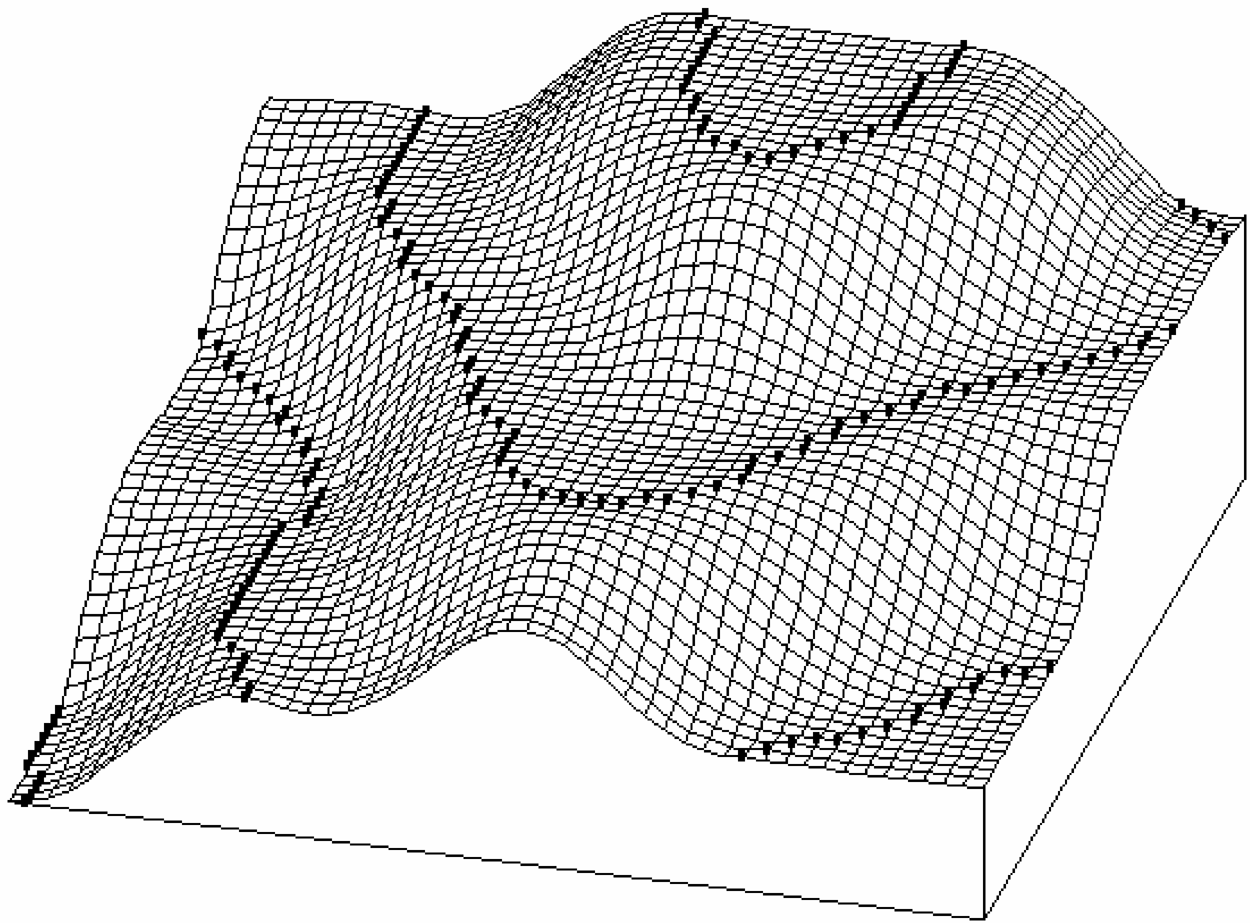
Membrane model after 40 iterations:

➔ Fast convergence



Variational spline interpolation - 13

Thin plate model after 500 iterations:



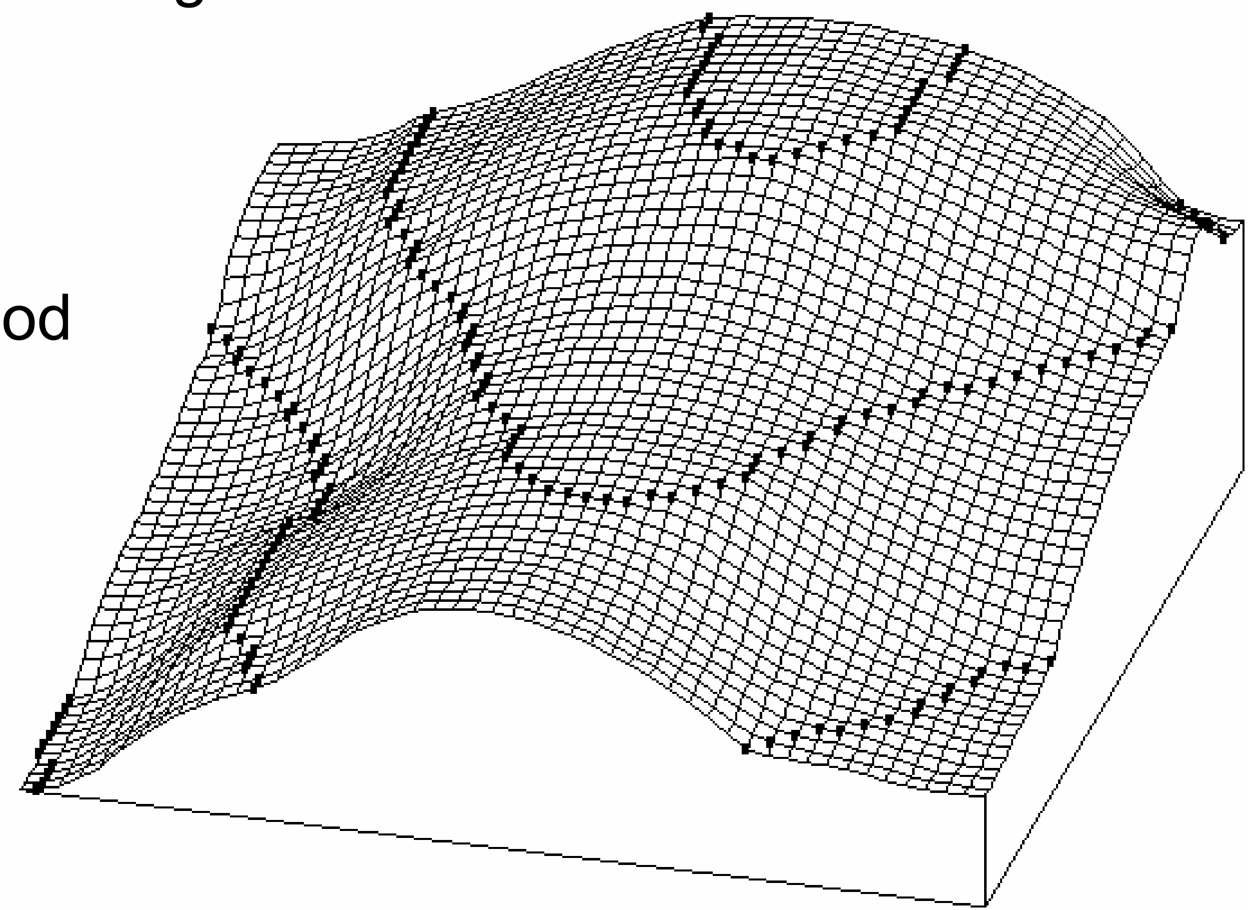
Variational spline interpolation – 14

Thin plate model after 5000 iterations:

➔ Very slow convergence!

The solution:

multigrid method

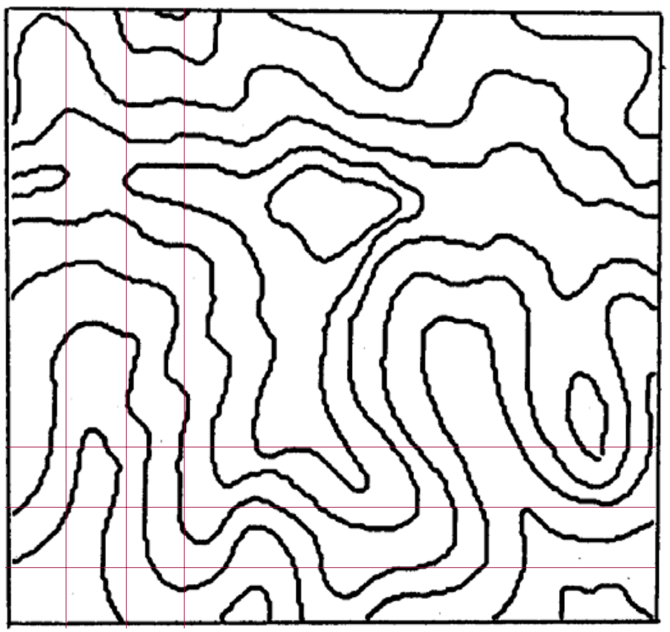


The multigrid method

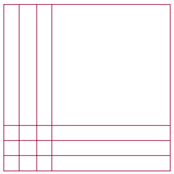
Basic idea:

1. Create a reduced matrix *R* from *Z* by averaging data points. If a tile does not contain a data point, the corresponding reduced pixel will be undefined.

Z



➔



R

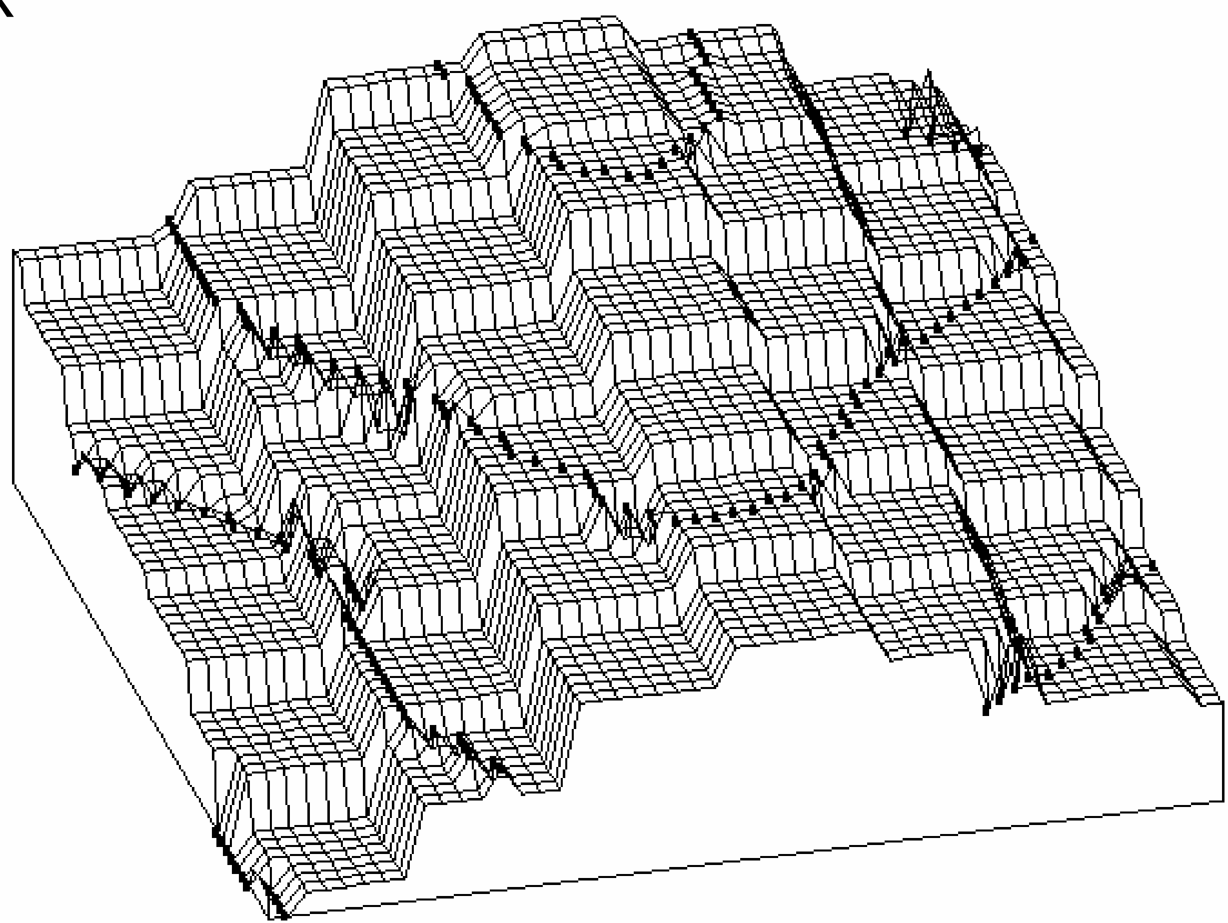
The multigrid method - 2

Basic idea (continued):

2. Give initial values to undefined elements of R.
3. Perform iteration for R. Convergence is faster because of the reduced matrix size.
   The result is denoted by R^*.
4. Give initial values to undefined pixels in Z by enlarging R^* to the original size.

The multigrid method - 3

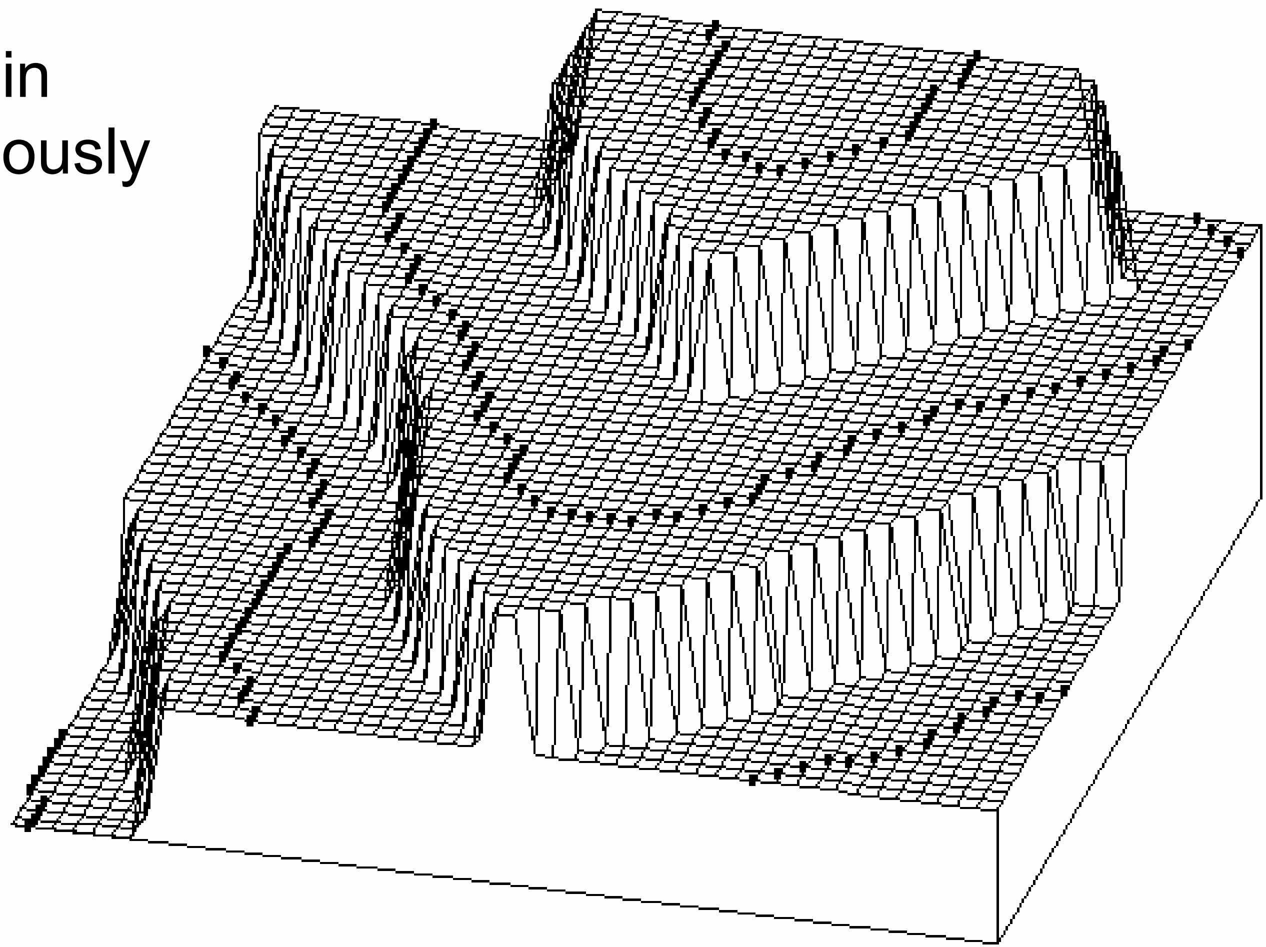
Example: initial terrain gained from an 8-reduced matrix



The multigrid method - 4

Remember:

initial terrain
used previously



The multigrid method - 5

The multigrid principle:

Use a hierarchy of reduced matrices:

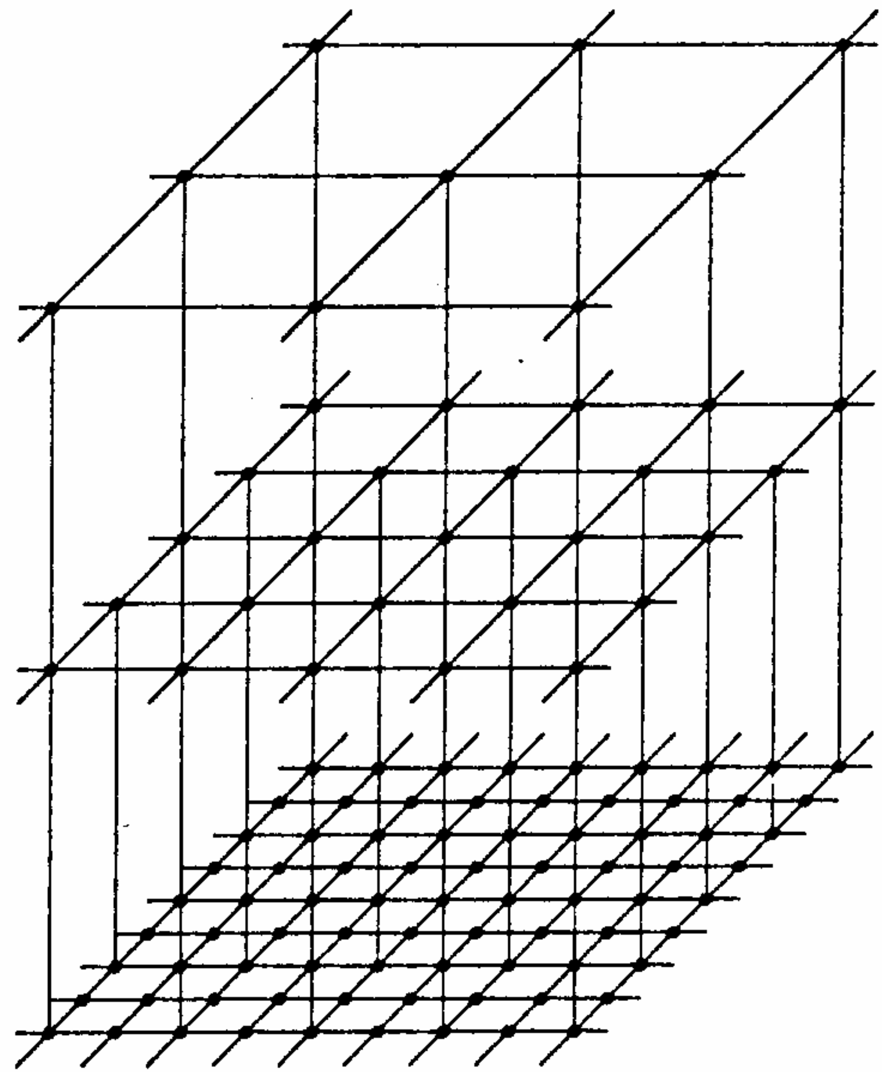
R_0 of size 1 x 1

R_1 of size 2 x 2

R_2 of size 4 x 4

...

$R_n = Z$ of size 2^n x 2^n



The multigrid method - 6

Multigrid algorithm:

The original contour matrix Z is of size 2^n x 2^n.

1. Create R_0 of size 1 x 1 from Z (by averaging all the data points).
2. Create R_1 of size 2 x 2 from Z by reduction. Unknown pixels of R_1 get initial values from R_0. Iterate for R_1, the result is R_1^*.
3. Create R_2 of size 4 x 4 from Z by reduction. Unknown pixels of R_2 get initial values from R_1^*. Iterate for R_2 , the result is R_2^*.
4. Continue the procedure until R_n=Z. Unknown pixels of R_n get initial values from R_{n-1}^*. Iterate for R_n , the result is R_n^*.

The multigrid method - 7

Conclusion:

- Only 10-40 iterations are needed at each multigrid level, even in the case of thin plate model!
- Time and memory required:
  $T = t + t/4 + t/16 + ... + t/4n < 4t/3$

Variational spline interpolation with multigrid

Summary for thin plate:

- Local maxima or minima may occur ***not*** only at given points.
- It is efficient if multigrid method is applied.
- Physical terrain features can be considered in some sense.

References - 1

- 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.
- 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.
- 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.

References - 2

- Lam, S.-N., 1983, Spatial interpolation methods – a review. *The American Cartographer,* **10**, 129-149.
- Terzopoulos, D., 1983, Multilevel Computational Processes for Visual Surface Reconstruction. *Computer Vision, Graphics and Image Processing*, **24**, 52-96.
- E. Katona: Contour line thinning and multigrid generation of raster-based digital elevation models. *Internat. Journal of Geographical Information Science* (Taylor and Francis, London), 2006.