CONTOURLINES FROM DEM-GRID USING IMAGE PROCESSING TECHNIQUES

Håkan Malmström
Swedish Space Corporation
P.O. Box 4207, S-171 04 Solna
Sweden
Commission III

Digial Elevation Models (DEM) are often stored and processed in regular grids. This yields efficient addressing methods for accessing individual height points. The regular grid concept is similar to storage of raster images. This opens the possibilities for using image processing systems for storing and processing of DEMs. Both computing efficiency in image processing hardware, and different raster operators implemented in hardware, are of interest for DEM processing.

Displaying a DEM is a simple task for an image processing system. Interactive change of the transfer function (Look Up Table) from image memory to monitor display, gives the operator an efficient tool for visual inspection of a DEM or its derived products (slope, gradient, etc). This interactive visual inspection/analysis can be done in black/white or in colour. The interactively designed display can afterwards be sent to a raster output device, for example a filmwriter or a raster plotter.

Contourlines is the standard technique for DEM presentation on topographic maps. Raster systems are improving all the time (higher resolution on hardware display and output devices, more inexpensive image memories) and they will be used more and more for DEM presentation. However, contourlines will remain the most widely used technique for DEM presentation, as both production and interpretation are well known and familiar to producers and users. This paper describes how image processing hardware can be used for automatic calculation of contourlines from a DEM grid.

1 INTERPOLATION IN DEM GRIDS

Calculation of the height of any given position in a DEM consists of two parts; first, the surface must be described as a mathematical function, and secondly, the actual height value is calculated, using the mathematical function. The choice of mathematical fuction for surface modelling is not an obvious one. Complex or high-order functions need a lot of computational effort. Primitive functions are computationally efficient, but they yield larger modelling errors.

The generation of contourlines as described in this paper uses a bilinear surface model. A simple mathematical model like this is easy to implement in hardware, and the behaviour of the function between gridpoints is well known.

Other image processing methods can be used for generation of contourlines in raster mode. Standard THRESHOLD and BORDER functions yield a contour map in raster format. See Figure 1.

```
8 12 13 14      0  1  1  1     0  1  0  0
5 12 11 12 -> 0  1  1  1 -> 0  1—1—1
9  7  8 10      0  0  0  1     0  0  0  1

   DEM         THRESHOLD      BORDER
  image          image        image
```

Figure 1: Threshold of DEM (lower threshold =10) followed by border operation on threshold image, yield contour lines in raster format

These standard functions work well for nice, well separated curves. Problems arise when the curves are running close to each other, for example in the standard ridge-valley case. See Figures 2 and 3.

```
10   12      10   12     10   12
               +             *
                 +         *
12   10      12   10     12   10
```

Figure 2: Original data, interpreted as valley (+) or ridge (*)

```
11   8   9   8  13    1   0   0   0   1    1 \ 0   0   0   1
13  12   9   8  14    1   1   0   0   1    0 >1 \ 0   0  /1
14   8  11  12  15    1   0   1   1   1    1<  0 >1 —1 /  0
13  11   8  13  16    1   1   0   1   1    0 \1/  0   1   0
14  12   9  11  14    1   1   0   1   1    0   1   0   1   0

      DEM                  THRESHOLD               BORDER
```

Figure 3: Two curves coming close together yield the standard ridge-valley case (threshold = 10)

The ridge-valley case can be solved by formulating rules like 'interpret all ridge-valley cases as ridges'. In order to convert the border image into vector polygons, more logical rules must be formulated, in order to create vector polygons out of different border patterns. This is a standard task in the raster to vector conversion of scanned line maps. A drawback with the border method is that the geometric location of the curve will be wrong. The reason for this is of course that we lost some information about the height in the thresholding procedure.

2 ALGORITHM

In order to get a simple and well defined procedure for contour computation, the following algorithm has been designed.

The entire DEM is divided into boxes. See Figure 4.

```
8 ——12——13——14
|box1 | box2 | box3 |
5——12——11——12
|box4 | box5 | box6 |
9——— 7———8——10
```

Figure 4: DEM with 3 lines and 4 columns containing 6 boxes

The total algorithm is divided into two steps:
A - calculating contourline segments for each box, yielding short curve segments
B - linking all segments together to complete curves

The height inside each box is modelled by bilinear function, and each box is examined to find out if the current contourline level is crossing any of the four edges of the examined box. See Figure 5.

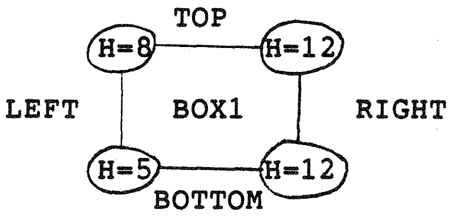

Figure 5: Box no 1 with its four heights H in the corners, and the four edges, named TOP, LEFT, RIGHT and BOTTOM

For countourline level 10m two crosses can be found in box1, one at the TOP edge, and one at the BOTTOM edge. The result for box1 is a curve segment between TOP and BOTTOM.

Step A is performed on one box at the time. Hence, a pipeline filterprocessor will here be of great advantage. The result of step A is a curvecode image, where each code describes:

- which edge of the box the contourline crossed
- where on the edge the contourline crossed the box (linear interpolation)

For a one-dimensional linear height function (****), four cases can be identified for a given contour level (CL). These are:

```
              *    *    *   *
             *      *  *     *
contour     *        **       **       **
         _______  ________ ________ ________
level      *                    *     *  *
 (CL)     *                      *   *    *
          normal     min   terrace     max
         f'(x)#0  ---------f'(x)=0----------
```

In order to get uniquely defined contourlines, a small value (E) is added to the desired contour level (CL) in the interpolation calculation. The value E is much smaller than the smallest unit of height, and the value (CL+E) is only used in the interpolation calculation. As (CL+E) is never found as a value of the discrete height function, only cases with f'(x)#0 need to be handled. Situations with f'(x)=0 will never occur. This is the purpose of the addition of E. A consequence of this approach, is that all maxima of the function are eliminated in the contourline map, and that all contourlines at terrace level are drawn at the "high end" of the terrace.

Step B links all the short curve segments together. The procedure can be described step by step:

- loop through the entire curvecode image until a curvecode #0 is found
- if curvecode #0 is found:
  1) store contourline point
  2) find out where it leaves the box and enters the next box
  3) delete used code
  4) goto next box and repeat steps 1,2,3 until the curve is closed, or until the curve leaves the DEM

The algorithm also includes rules for handling the ridge-valley situation described above. This is necessary when four crosses occur for one single box. Random access of image data is a prerequisite for step B, and therefore this part is done in a processor with that capacity, or done in the software.

3 EVALUATION

The algorithm for contourline generation presented here is designed for implementation in image processing hardware. The separation of the algorithm into two steps gives advantages:

- curve coding with sequential access of data can be processed in a pipeline processor
- linking of curve segments originating from the curvecode image eliminates the need for a plot image - a plot showing all curves that have already been used. Used curve codes are simply deleted in the curvecode image

The interpolation gives a geometrically more correct result, compared to other image processing techniques. The addition of a small value E in the interpolation gives a systematic error in the terrace and max cases. The influence of the systematic error is small, as it concerns only the smallest unit of storage. The measuring error is normally larger than the least storing unit, and hence the systematic error can be neglected. The advantage of this approach is that it reduces the number of cases to take care of, and that it gives uniquely defined contourlines.

The use of image processing hardware offers the use of all the standard tools (filtering, resampling, etc) for preprocessing before the contourline generation. Smoothing of output contourlines can then be performed by preprocessing of DEM with a lowpass filter, instead of by postprocessing of vector data.

4 CONCLUSIONS

The presented algorithm offers a tool for automatic generation of contourlines from DEM-grids using image processing hardware with high capacity.

5 REFERENCES

Computer Assisted Cartography, Principals and Prospects, M.S. Monmonier, 1982 Prentice Hall. ISBN 0-13-165308-3.