AUTOMATED RIVER LINE AND CATCHMENT AREA EXTRACTION FROM DEM DATA

by Wolfgang Rieger
Institute of Photogrammetry and Remote Sensing
The Vienna University of Technology, Vienna, Austria
ISPRS Commission IV

ABSTRACT:

A GIS-based algorithm has been developed for the automated extraction of river paths and catchment areas from a grid DEM, including automatic removal of spurious pits. It consists of a series of independent "general functions on DEMs" working on different data levels, which yield the base of a GIS-structure. Pit removal is done by raising depressions in a nearly form-invariant way and by lowering a "main flow path" within the depression, so that it leads outwards. Rivers result from summing up the water flow. Catchment areas are defined for river sections (between two neighbouring nodes) and found by "flooding" the DEM. For both the summing and the flooding, work is done in the order of sorted elevation. Sorting proves to be fast enough compared to the duration of the other processes. Internally, the DEM is structured as a pyramid. This allows for nearly unlimited DEM sizes as well as for feature extraction in a scale-dependent way. The GIS concept allows for creating any number of new data set layers as intermediate or final products of calculations, thus resulting in high flexibility.

KEY WORDS: DEM, DTM, GIS/LIS, Feature Extraction, Hydrology

1. INTRODUCTION

Recent years have brought up a lot of research in hydrological modelling by means of digital elevation models (DEMs). An excellent overview can be found in (Moore et al., 1991). For hydrological modelling,detailed analysis of large terrain zones is of major importance. Generally, the demands of quality conflict with user's call for computational efficiency. The job is trying to derive global (in regard of the region of interest) terrain features or characteristics from local processing, which would reduce computational cost from proportional to about n^3 (global processing) to n^1, if there are n points involved.

A series of approaches has been made to do this. O'Callaghan and Mark (1984) and followers (Jenson and Domingue, 1988; Sperling, 1992; and others) applied a local operator by a way somewhere in between iteration and special sequencing that arises from adaptation to terrain features. Another approach was to use an expert system shell to apply a series of specific rules to the results of local operators. These rules include similarity tests of elevation, curvature, or orientation of river segments (Qian, Ehrich, Campbell, 1990).

Here, a local operator will work in the order of ascending respectively descending elevations within the model. Sorting proves to be fast enough compared to a lot of other time-consuming tasks and allows for extensive use of the sorted data set. Simply to be used within a series of algorithms, the method proves to be practical even for very large data sets, especially when applied only to picked out parts, such as catchment areas. A more detailed description see (Rieger, 1992).

2. DATA STRUCTURE

2.1 The Digital Elevation Model (DEM)

A regular rectangular grid structure is used for the DEM. Each height value is stored within a matrix in row-ordered form. There are no breaklines embedded. Data are all stored in one matrix or, for very large models, in a few submatrices derived from splitting up a too large matrix. A special value is used to indicate that a point is "undefined", that means it lies outside the region of interest and will be ignored. The rectangular grid structure has a lot of advantages: Most algorithms simply can be implemented; there is nearly no administrative amount necessary; addressing relations between neighboured points are all the same over the whole matrix; algorithms and visualization techniques of digital image processing can easily be applied.

There are some disadvantages, too: The necessity to adapt the grid widths to the roughest terrain may lead to significant overhead in smoother regions; the regular structure cannot handle breaklines (Mark demands that the phenomenon should influence data structures, not computational considerations (Mark, 1979)). Besides, the use of a matrix may cause considerable overhead when working on regions, that do not fit well into an ax-parallel rectangle. And last but not least, the size of available (virtual) memory limits the size of the matrices.

To overcome some of these disadvantages, a pyramid structure is used. Starting with a sparse grid of the whole region as a low level, the next higher level is built up of a grid with half grid widths in both directions. The next level uses quarter of the base grid widths and so on. There are two principal ways to obtain this structure: The first one is to calculate the grid iteratively at decreasing grid widths, as proposed by Hutchinson (1989). He uses this method to derive hydrologically high qualitative grids, especially with removal of spurious pits.

The second way is to calculate the finest grid as a high quality DEM (in this case the SCOP-DTM package is used (Molnar, Waldhaeusl, 1990)) and successively build coarser resolutions by using non-overlapping 2x2-blocks of grid-cells to derive the height of one grid-cell of the next coarser grid. Simple arithmetic mean can be used. Other methods, e. g. 4x4-overlapping blocks, can be used for better smoothing, too, but the arithmetic mean yields fast operation and also smoothes the terrain.

For dealing with very large DEMs (e. g. from several 100,000 points upwards) it may be comfortable or necessary to split up the larger matrices into respectively 4 submatrices of the same size as the matrix in the next coarser resolution.

The total amount of storage will be about 30% more than used for the finest resolution. The main advantages of this structure in regard to the operations proposed here are listed in chapter 7.

2.2 GIS-concept

A grid based GIS-concept is used. Data are organized in any number of layers, each of which has the same structure as shown in chapter 2.1, called a data set. General functions are declared for the manipulation of these data sets. Since each level of each data set is stored as a matrix (or as a series of matrices when splitting up is necessary), all functions can work on any layer in any level. The only restriction is given by the internal data type used for one single element (byte, or floating point value) in a specific layer. Elementary functions may be element-wise addition, subtraction, multiplication or other fundamental arithmetic operations with matrices only or matrices and constant values. Typical DEM-operations may be calculation of slope, aspect or curvature in each grid-cell, thus creating matrices of layer slope, aspect and so on. A more complex function may create a matrix with the size of the pertinent catchment area in each grid-cell. Or, as a coded array, the code-number of the catchment area each grid-cell belongs to.

This concept allows for easy extension by new functions. Thus, a GIS can be created with a set of general functions to combine data sets and create new ones. Data can be transferred from and to existing systems. Most algorithms can easily be implemented. Methods and visualization techniques from digital image processing can be used. The concept is highly modular with all the advantages of these concepts (compare to Jenson and Domingue, 1988; Johnson, 1989). A series of functions can be combined to very complex operations.

2.3 Surface definition

Each value in a matrix is interpreted as the mean value of the surrounding grid-cell. Therefore, the expression "(grid-)cell" is rather used here than "point". Next, transitions are declared between immediately neighbouring cells. This yields four-connectivity, that means, that only the four cells immediately adjacent in ax-directions are taken as neighbours to each single grid-cell. Now, water flow is defined by height relations between the adjacent cells (see chapter 4).

3. PIT REMOVAL

3.1 General

Since pits hinder water flow, it is necessary to pay special attention to them. Completely flat areas will be treated as pits, too, since water flow is not defined in these zones.

There are two general types of pits in a DEM: Pits that represent real depressions at the terrain surface, and artificial pits, that result from data errors, interpolation method, or simply from the grid (i.e., if all neighbours of a grid point happen to be positioned on higher terrain than the point itself, whileas a flow channel would lead downwards between two neighbours). Furthermore, larger water surfaces will result in flat areas.

According to the types of pits, there are two ways to deal with them: Pits shall either be removed from the DEM - thus producing a "depressionless DEM" - or they shall remain. In the latter case water flow stops at the deepest point within the pit, or, if an underground flow path and a point of reappearance is known, the accumulated water from the pit's deepest point can be assigned to the reappearance point directly. This is a special task, that needs for interactive declaration of the assignment. Because of operator's intervention and because of the sparse number of such depressions, this is of minor importance. Mark (1988) notes that depressions on the terrain surface are very rare within a range of about 10 meters or greater; they will only occur within special geomorphologic features (e.g. karst).

The great majority of pits shall be removed to obtain a depressionless DEM. O'Callaghan and Mark (1984) proposed to smooth the terrain as a preprocessing task. This removes only smaller pits and smoothes all the terrain. A smoothing restricted to depressions is possible, too, but does not remove larger pits, either. In general, it is assumed that water would overflow pits at the deepest point along the border of the catchment area of the pit. This assumption is made by most researchers (O'Callaghan and Mark 1984, Jenson and Domingue 1988) because of the lack of information. Jenson and Domingue (1988) also permit splitting especially of flat areas, if there are more than one outflow points.

However, only when dealing with the original data, a more adequate assignment of pits is possible. Hutchinson's procedure produces depressionless DEMs with regard to the original data and special checking, which point is the most reliable outflow point (Hutchinson 1989). Another possibility is to use digitized stream channels (Mark 1988). This is restricted to pits occuring along stream channels, which are the great majority of pits. Though, if stream channels shall be automatically derived from the DEM, it makes little sense to first digitize them.

3.2 Filling depressions

Filling depressions is presented here as one possibility to obtain a depressionless DEM. While Jenson and Domingue (1988) raise all cells within a pit exactly to the height of the outflow point and therefore have to use an additional data set to define water flow, the "flow direction" of each cell, the concept here is to largely retain the form of the depression and the height relations so that the height layer will contain all necessary information for the calculation of water flow.

Each depression is assigned an entry in a "pit-table", including the deepest point of the pit, the outflow point, and the code-number of the catchment area the outflow point belongs to. The pits are detected in progress with a catchment searching procedure as described in chapter 6. The height of the outflow point is reduced, while the deepest point of the pit is raised above the new height of the sunk outflow point. This is done in a way that the deepest point of the pit will still be lower than the old outflow point and the outflow point will still be higher than its deepest neighbour outside the pit. This enables raising the pit beginning from the - now lowered - outflow point, upward to the deepest - now raised - pit point and going on to the upper edge of the depression. For numerical reasons, it may be impossible to reach this constellation: If the outflow point's deepest neighbour has nearly the same height as the outflow point itself and there is a long way to go to the deepest pit point, a pit would remain, but, be much flatter. This method shall, in general, preserve significant height differences and prevent of the formation from new pits.

Filling is done the following way: Starting from the outflow point's cell and growing into the pit by fronts, each cell of the pit is changed in height. Each front-cell is directly neighboured to at least one cell of the previous front. All cells within a front are sorted according to their original height values. The lowest front cell is assigned a value higher than the respective lower neighbour of the previous front, and all cells of the front obtain values higher than the respective lower neighbours within the front. For this, an incremental value is used, that is adapted to the height difference between the old and the new height of the outflow point and the size of the depression to yield proportionate forms.

The same algorithm can be used to deal with completely flat areas. Here, the sorting of the front is done from the middle of the front outside in both directions. This is done because of the lack of other information and yields acceptable results for flat water surfaces. When dealing with flat terrain, however, better criteria should be adapted (e.g. in consideration of the pit's surrounding surface).

3.3 Lowering a main flow path

Another way to eliminate pits is to sink a so called main flow path. One part of this path is found by flooding water from the outflow point into the pit, until it reaches the deepest pit point. Searching is done by following from each grid-cell to the deepest neighbouring cell that is not higher than the cell itself. The second part of the main flow path proceeds from the outflow point outside the pit, again following to the respective deepest neighbouring cell not higher than its predecessor, until it reaches terrain that is deeper than the deepest pit point. Along this path, the height values are changed, so that the grid-cells continuously lead downwards from the deepest pit point to the outer endpoint of the path (fig. 1). The main flow path is a channel, that eliminates the pit, since all cells within the pit can drain to the deepest pit point's cell and this cell by itself now can drain outside the pit via the main flow path.

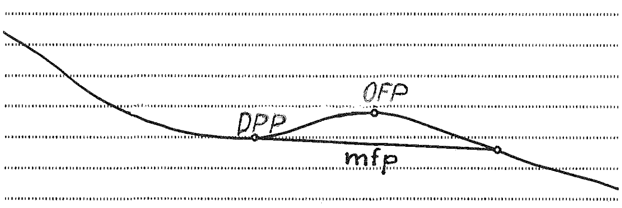

Fig. 1: Pit and main flow path (mfp), longitudinal section. DPP: Deepest pit point; OFP: Outflow point

This algorithm is similar to what is described by O'Callaghan and Mark (1984), except that they only change the "drainage direction values" and do not change the height values. Therefore their drainage data set represents the drainage data set of a depressionless DEM, the DEM by itself remains unchanged. Here, only the height data set is used, thus it is changed.

This method can also be used to deal with systems of nested pits. For that, the pits have to be eliminated in the sequence of ascending outflow points. Whenever a depression is ready for elimination, all its interior depressions already have been dealt with, so that they drain to the depression's deepest point.

In most cases this algorithm will produce depressionless DEMs in one step. Only within flat areas or very large depressions with small height differences pits may remain; in this case the algorithm must be repeated.

If there are deep depressions, as may arise from improper input data combined with improper interpolation techniques (e.g. digitized contour lines in broad valleys with no other height information within the valley), deep flow gullies may result from this method, that give an unrealistic surface. In these cases a combination with the filling method (as described in chapter 3.2) can be used:

3.4 Combination - the final algorithm

As a first step, the DEM is smoothed to fill a lot of single-cell pits. This task is restricted to the pit points and therefore does not effect the rest of the DEM. It is furthermore restricted to pits that would not create new pits in their immediate neighbourhood, when they are eliminated, thus keeping the changes in the DEM small.

Next, all depressions are filled by the method of raising fronts (chapter 3.2). Finally, the remaining pits and flat areas are eliminated by lowering their main flow paths. This last step must occasionally be repeated to yield a completely depressionless DEM. Depressions that shall not be changed must be marked before the start of the pit-removal task to be excluded from these operations.

4. CONCEPT OF THE DRAINAGE QUOTA VALUES

To obtain river networks and catchment areas, some special considerations on the water flow must take place. Here, a special and very simple local operator is proposed, that can be used for a good simulation of the topologic aspects of water flow.

Water always flows in the direction of the steepest slope. Normally, this direction is not restricted to neighboured grid-cells (4 or 8 neighbours). When working in the grid DEM, decision must be made to which cell water would drain. This can lead to gross errors along larger slopes, because water would arbitrarily float in direction of the grid.

In reality all deeper neighbours get some part of the outflow of a grid-cell. A simple method to simulate this is the use of so-called drainage quota values: The drainage quota value (DQV) is defined as the quota of the accumulated water in a cell that will be sent to one of its neighbours. Let w_0 be the water accumulated in a grid-cell; w_i, the amount of water that flows to one of the cell's neighbouring cells in 4-connectivity; h_i be the height of neighbour i, h_0 the height of the grid-cell itself. The drainage accumulation value then can be calculated as (4-connectivity is used, because only 4 neighbouring cells have one common side with the grid-cell):

$$w_i = w_0 \cdot dqv_i \qquad (1)$$

The "drainage quota value", *dqv*, is defined as:

$$dqv_i = \Delta h_i \,/\, \sum_{j=1}^{4} \Delta h_j \qquad (2)$$

$$\text{with} \quad \Delta h_i = \begin{cases} h_0 - h_i & \text{for} \quad h_0 > h_i \\ 0 & \text{for} \quad h_0 \le h_i \end{cases}$$

First the starting cell is initialized with the water unit. Next, calculating these values from up to low, a drainage band is obtained. Figure 2a) shows the drainage band resulting from the drainage quota values, with white being quota values of (nearly or exactly) 1, gray of lower than 1, and black with quotas of (nearly or exactly) 0. Figure 2b) shows the result of following the path of maximum drainage quota values downwards in 4-connectivity.

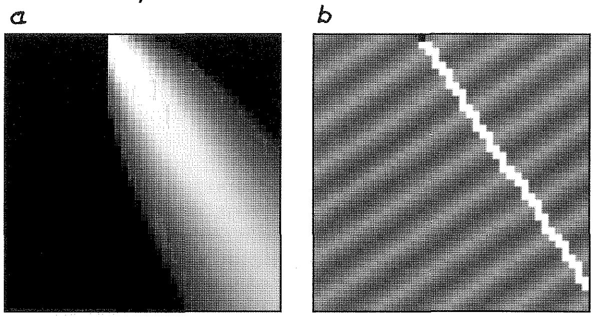

Fig. 2: Drainage in a slope. a) Drainage quota values (DQV) highlighted; b) Path of maximum DQV's on contour-bands (dark gray lines against light gray zones).

The drainage band is unrealistically wide-spread. This results from the simplicity of the assumption: In reality, not all water accumulated in one cell drains to all deeper neighbours as is calculated with the DQVs. Rather, there is only a channel in which water flows. Figure 3 shows splitting up of these channels by the grid lines along a test surface (smooth valley). It is obvious, that most channels drain only to one single neighbouring cell.

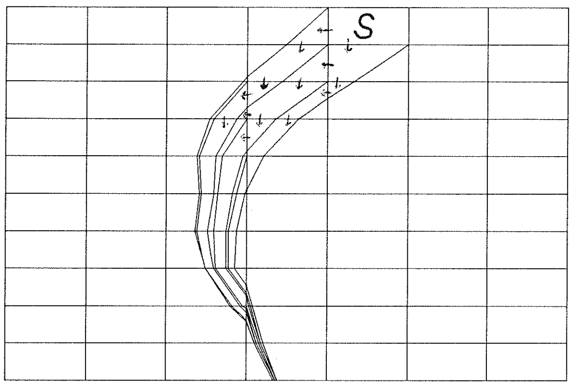

Fig. 3: Splitting of drainage channels in the grid. **S** is the starting cell, the arrows show transitions of water. The channels arise from the edge points around the grid-cells.

The concept of the DQVs allows for a simultaneous calculation of all channels, with the only disadvantage of the unrealistic wide-spread drainage band. The maximum line gives the flow direction. The DQVs can also be used to calculate the water flow of a greater region, as will be shown in chapter 5. Furthermore, in chapter 6 it will be shown how to adapt this concept to the delineation of catchment areas.

5. DERIVING RIVER PATHS

5.1 Test region

Figure 4 shows a test region in a combined representation of contours and hill-shading, created in the following way: At first, the hill-shading value is calculated for each grid-cell according to Tanaka's algorithm as a floating point value between 0.0 and 1.0 (Horn, 1982) and stored in an own data set. Next, the height values are grouped in zones and another new data set is created that includes varying values dependent on the zones (here, periodically changing values from 0.4 to 0.6 in 5 steps are applied, each step representing a height zone of 5m). These two data sets are multiplied and scaled from the range [0.0 ... 1.0] to the range [0 ... 255].

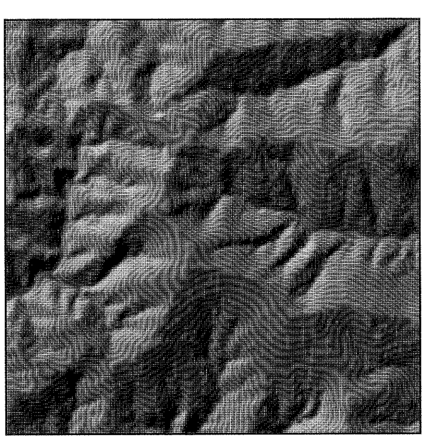

Fig. 4: Test terrain with hill-shading and overlaid contours

5.2 Iterative calculation of the DQVs

The DQVs can be calculated over the whole DEM. Initially, two sum-arrays, **S1** and **S2**, all filled with zeroes, are provided along with the height data set **HGT**. Now a rainfall-simulation with constant amount of water over all the region is made by adding **1** to all grid-cells of **S1**. Each grid-cell now contains an amount of one unit of water. Then in each grid-cell the DQVs to its (deeper) neighbours are calculated according to equation (2) to drain its water to these neighbours according to equation (1). These values are summed up in the respective grid-cells of array **S2**. When the process is finished, **S2** contains in each grid-cell the sum of all water the cell has received from its higher immediate neighbours.

This process gives DQVs greater than 1 for concave zones (valleys), lower than 1 for convex zones (ridges), and about 1 for (slanted) plains. This yields good impression of the local relief. Larger valleys or ridges, however, especially if pretty flat, most often cannot be identified. Thus, iteration is necessary, which simulates water flow across a landscape (without consideration of flow velocity, oazing away, ground water, etc.).

Iteration is done in that the output data set **S2** of one iteration, which contains the accumulated water, is taken as input data set **S1** of the next step of iteration: Rainfall is simulated by adding values of **1** to each grid-cell in the old data set **S2**, the result written to **S1**. Data set **S2** then is set to all zeroes, again, and the calculation of water flow can take part from **S1** to **S2** as described above. Figures 5a to 5e show different steps of iterations.

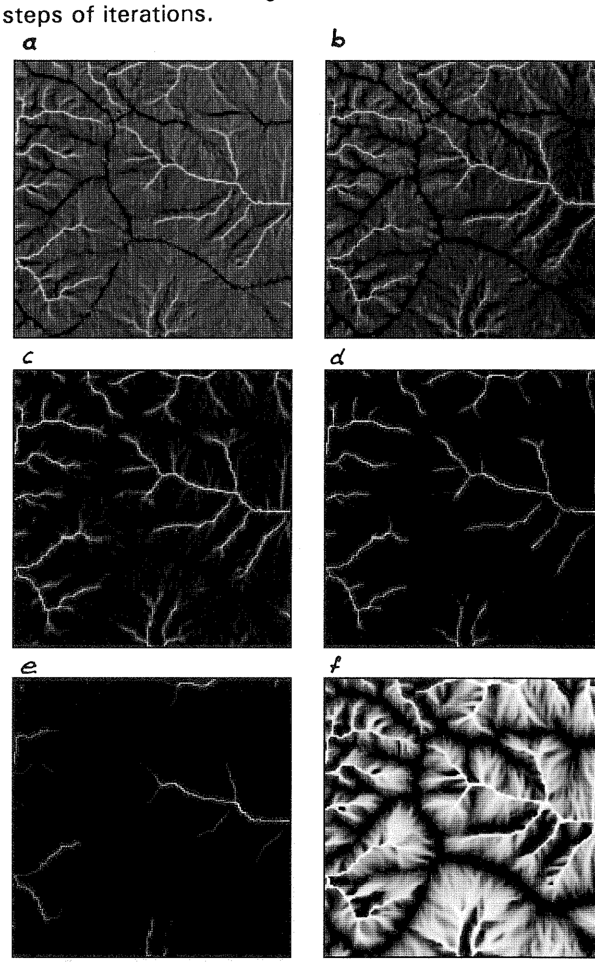

Fig. 5: a) - e) Steps of iteration of the calculation of the DQVs in a part of the model of fig. 4; no. of iterations: a) 1, b) 2, c) 5, d) 10, e) 20.
f) Histogram equalization of picture e).

Note that in step *n* water of all flow paths with lengths of up to n grid-cells (in 4-connectivity) upwards from each cell contributes to the amount of accumulated water in that cell. The system is end-iterated, when the water has reached the edge of the region of interest along the longest flow path in 4-connectivity. In practice, 20-50 iterations will be enough to yield good results.

The pictures show no wide-spread drainage bands as can be observed with single-point drainage (fig. 2a). The reasons for this are: The drainage bands tend to lead into valleys and they converge within valleys. Therefore they are generally not wide-spread as in tilted plains (fig. 2a). The second reason is, that drainage bands of neighboured cells tend to balance one another.

The algorithm is best suited for parallel processing. In each grid-cell, the calculation of the DQVs is independent from one another except that the values have to be added in array **S2**. In a multi-processor environment, each processor can work on one part of the array, a so-called "processing region". The only synchronisation condition is, that two processors should not calculate within their common border region at the same time. This can be obtained, if all processors work on their submatrices in the same row-order.

5.3 Ordered calculation

If only sequential processing is available, the iterative process is very much time-consuming. If, however, the process is done from top to bottom in the DEM, it can be finished in one iteration. To do this, an indexing array is used, containing the matrix addresses of the grid-cells in sorted order. With efficient sorting algorithms like quicksort, implemented for example as a standard function in the programming language C, this sorting is about as time-consuming as a few iterations of the iterative calculation. Furthermore, there are possibilities to accelerate the sorting task for very large models (see chapter 7).

The process now works in one single sum-array, referred to as **S**. At first, rain-fall is simulated by assigning a value of 1 to all grid-cells, as mentioned above. Next, the DQVs are calculated for the highest grid-cell and immediately added to its neighbours. As the algorithm works from top downwards, each cell has already received all water from all cells along all drainage paths that drain through this cell, when its outflow is calculated. The result from working on all cells, the so-called "drainage accumulation" data set, is identical to the result of the end-iterated iterative calculation (except for numerical inaccuracies). Picture 6 shows the result for the model of picture 4 and the same with a histogram equalization (256 gray levels used for data representation).

a b

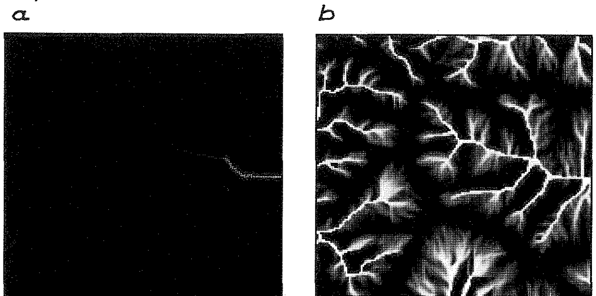

Fig. 6: Drainage accumulated by the ordered calculation in a part of the model of fig. 4. a) Values as calculated: White means much water (maximum: appr. 5850 units), black little water. b) Values after a histogram equalization.

5.4 Obtaining river networks

The drainage accumulation values pretty well represent the area of the catchment to each grid-cell, expressed in numbers of grid-cells. The data set is similar to the "flow accumulation" data set of O'Callaghan and Mark (1984), Jenson and Domingue (1988) and Sperling (1992) except that here each cell may have more than one channel for water runoff (the term "drainage accumulation" is used here for distinction to "flow accumulation"). With the flow accumulation data set a threshold value is used to mark all cells with a higher value as channel cells. This method is, in principle, also applied here. However, since water may spread across flatter areas with the drainage accumulation values, channel lines may become wider than one single cell or may be interrupted. For these reasons and besides to obtain a hierarchical river network structure, channel lines are traced as follows:

Starting from outflow points along the edge of the terrain (detected as maxima in drainage accumulation and minima in height along the edge), channels are persecuted by tracing to all higher (in elevation) neighbours with drainage accumulation values higher than the specified threshold value. If there are more than one paths, a temporary node is installed, and all these paths are traced consecutively, starting with the one with the highest drainage accumulation value. Then a second threshold value is used, that represents a minimum length value for river channels. Channel pieces that are too short are rejected and the temporary node is deleted. If, however, two channels are long enough, the node is accepted and the two new channel segments are inserted. The channel with the larger catchment area (i.e., the higher drainage accumulation value) is expected to be part of the same river line as the lower course, the other one opens a new level in the hierarchy of the river network.

The procedure is implemented as a recursive algorithm, thus it will work the same way on each channel cell or node. Picture 7 shows the river networks obtained by this algorithm, channel sections shaded in different gray levels.

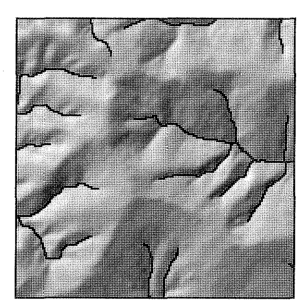

Fig. 7: Channel network obtained by tracing upwards; part of the model of fig. 4.

Another way to obtain the nodes would be to mark all cells with drainage accumulation values higher than the threshold value and a difference in this value to its higher neighbours, that is also higher than the threshold (Jenson and Domingue, 1988). This could prevent from stepping into most short channels, that later on will be rejected. This has not been tested, however.

6. LABELLING OF CATCHMENT AREAS

The next step is to label the catchment areas. The job is, that all sub-watersheds shall be obtained to all river sections. O'Callaghan and Mark (1984) and Jenson and Domingue (1988) assign basin labels for each catchment area, beginning from a starting cell (i.e. an outflow point along the edge of the region), by recursively assigning the label to all neighbours with flow directions to already labelled points. Here, a slightly different algorithm is used utilising the existence of a sorted indexing array. Work takes place upwards. To each grid-cell all deeper neighbours (in 4-connectivity) are examined. The following conditions may occur:

1) No deeper neighbours exist.
   a) The cell lies at the edge of the region: A new "global basin" starts.
   b) The cell lies within the region (this is impossible in a depressionless DEM): A new depression with a "pit-basin" starts.

2) All deeper neighbours belong to the same basin: The cell is assigned to this basin.

3) Deeper neighbours belong to different basins: The cell is marked as a border cell.

4) All deeper neighbours are border cells: The cell is marked as a border cell, too.

Criterion 1) b) has been inserted for use of this algorithm in the elimination of pits: If a cell meets this criterion, a new entry in the pit-table is created (compare to chapter 3.2). Following, whenever a cell has deeper neighbours partly belonging to a pit-basin and a global basin, this cell becomes outflow point of the participating pit-basin. It is recorded in the respective entry in the pit-table and the pit-basin is assigned to the participating global basin to prevent from creating further outflow points for this pit-basin.

Criteria 2) and 3) do not change, if there are also border cells involved. Picture 8 shows the result of this labelling procedure on part of the model of picture 4 with all catchment areas shown in different gray values and the borderlines in black.

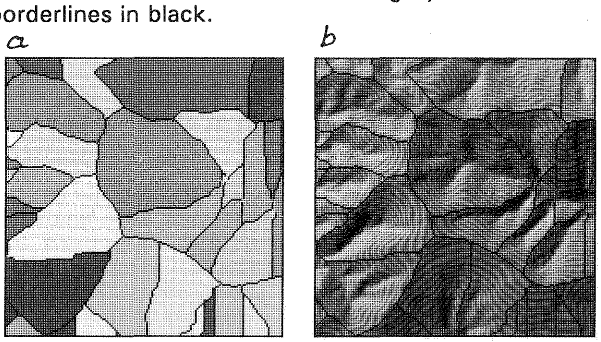

Fig. 8: Catchment areas and borderlines in a part of the model of fig. 4. a) Coded areas; b) border lines on terrain.

Obviously, there are some dependencies on the grid direction, and some borderlines are too broad. Therefore the concept of the DQVs is adapted as the "catchment affiliation values" (CAVs):

The base-cell of a catchment area is assigned a CAV of 1. The same is true for all cells along the marked river sections, which means, that all these cells completely belong to the catchment area. For processing reasons, the label is constructed as follows (*cnr* be the label number of the catchment area, *cav* the CAV in the respective cell, and *label* the value finally assigned):

$$label = \begin{cases} cnr & \text{for} \quad cav = 1 \\ cnr + cav & \text{for} \quad cav < 1 \end{cases} \tag{3}$$

The CAV furthermore is limited to a range of about 0.0001 through 0.9999 to avoid numeric zero for very small values (which would falsely be interpreted as a CAV of 1.0000) and numeric 1.0000 (which would falsely be interpreted as the catchment area with the next higher code-number, $cnr+1$). Each cell i is assigned a CAV from all deeper neighbours j of the same catchment area k as follows:

$${}_k cav_i = \sum_{j=1}^{4} {}_k cav_{ij} \cdot {}_k dqv_j \tag{4}$$

dqv is the drainage quota value as defined in equation (2). The affiliation values are now obtained by working from bottom to top of the DEM. Each cell's CAV is calculated from all its deeper neighbours, thus the process can be finished in one step just like the ordered calculation of the drainage accumulation values.

Within the catchment area all points have CAVs of exactly 1.0, while in the border regions the CAV will be lower than 1. In the last case, the cell belongs to more than one catchment area. Since only one value is provided, the cell is assigned to the catchment area with the highest affiliation value of all participating basins. For better results the complement of cav_{ij} is adjoined as the CAV to the other participating catchment areas as $(1\text{-}cav_{ij})/n$ in the neighbour j to point i, if cell i has n neighbours belonging to other catchment areas. Picture 9a) shows the CAVs for a part of the DEM of picture 4. Picture 9b) shows the zones with CAVs lower than 1 in dark gray as an overlay on the terrain.

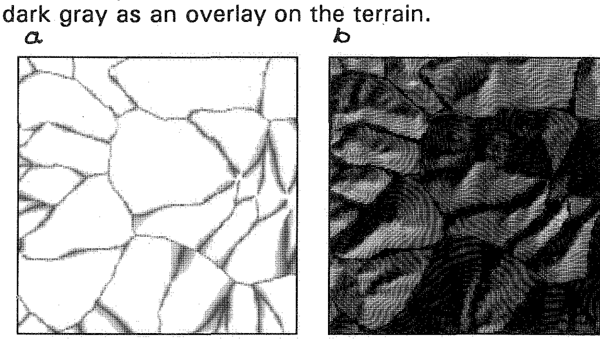

Fig. 9: a) Catchment affiliation values (CAVs) in a part of the terrain of fig. 4. White: CAV = 1, black: CAV = 0, gray: $0.5 \leq \text{CAV} < 1$. b) Zones with CAV < 1 in dark gray on the terrain.

Borderlines are found by examining all cells with neighbours belonging to other basins. The respective cells with the highest height value and the lowest CAV are marked as border cells. Picture 10 shows the final results, river-courses and catchment borders on the whole terrain of picture 4.

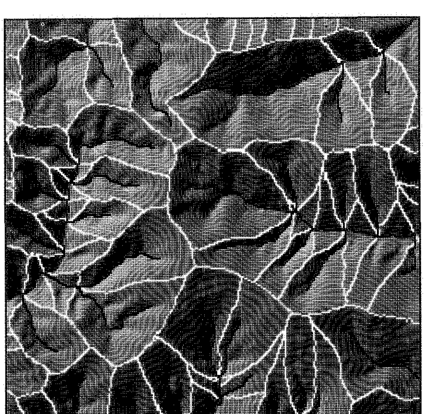

Fig. 10: Terrain of picture 4 with river courses (black) and catchment areas (borderlines in white)

The same algorithm can be used for marking the outflow points of depressions, as mentioned. Within depressions, no CAVs need to be calculated, however, since all cells of the depression fully belong to the depression's basin.

7. WORK IN PYRAMID STRUCTURE

The upper limit of DEM-sizes is caused by the (virtual) memory available. A maximum of 3 floating point matrices are to be held in memory simultaneously for the calculation of the drainage accumulation values in the iterative process (chapter 3.2) or 2 floating point matrices and one integer array of the same size for the ordered calculation. Most usual word-lengths are 32 bits for both floating point and integer values, therefore the indexing array uses the same size as the floating point array. For 1,000,000 grid-cells, e.g., this means about 12 MB of virtual storage. With modern workstations this is no real problem in terms of memory usage.

Table 1 lists computing times for one step of the iterative calculation of the drainage accumulation values and for the sorting of the indexing array in different DEMs of different sizes. After sorting, the calculation of the drainage accumulation values will take additional time of one step of iteration.

No. of points	Iter	Sort
10,201	0.61	1.02
58,081	3.81	7.57
152,348	9.87	20.05
231,361	15.53	37.03
301,101	20.48	45.89
535,200	36.31	86.45
1,204,200	87.13	313.02

Table 1: CPU-times in seconds on a VAX-Station 3100/M76 with 32 MBytes of main memory under the VMS operating system. "Iter" means one iteration of iterative calculation, "Sort" means sorting of indexing array by the ANSI-C function "qsort". These values are mean values and may differ about 2% up or down on repeated calculations.

The ordered calculation will take the time of about 3 to 5 steps of the iterative calculation for model sizes up to over one million of points. This factor includes the time of calculation of the drainage accumulation value in the ordered calculation. Thus, the use of sorting is greatly justified.

Nevertheless it is not commendable to use too large matrices, since terrain may strongly change in terms of roughness within larger regions, thus the grid width should be adapted individually. Furthermore, it is pretty unwieldy to work on very large matrices, and, overall, matrices are always limited in size.

For work on very large DEMs (from several 100,000 points upwards), the pyramid structure, as described in chapter 2, is proposed. When using this structure, the following advantages result:

- Work in different levels corresponds to scale-dependent generalization; it is possible to obtain river networks in a resolution that corresponds to the wanted scale.

- In each level the same algorithms can be used without any changes.

- Data sets of higher levels (finer resolutions) can use information of those of lower levels. This may allow for acceleration as well as - in some cases - for quality improvement. In particular, this can be used for:
  - Pit removal: In lower levels there are generally less pits. Flow directions can then be taken from pitless zones in lower levels - a priori pitless or after removing pits - to use the correct outflow point.
  - Catchment areas: The approximate knowledge of the catchment areas in the higher level allows for work within a single (larger) catchment area. Within such an area fast river-course extraction is as well possible as fast sorting and sub-watershed delineation.
  - River networks: When splitting larger regions, it is possible to impact drainage accumulation values obtained from lower levels (in regard of the changed grid cell size) along the edge of a region section as a constraint. This allows for splitting very long rivers without loss of information about its drainage accumulation values.

- Overview calculations are possible for quick terrain analyses. Detailed work then can be restricted to zones of special interest.

- The structure can easily be adapted for data capturing techniques such as progressive sampling.

Picture 11 shows a series of the test region of picture 4 calculated in different levels.

8. CONCLUSION AND OUTLOOK

Rectangular grid-DEMs in form of matrices can easily be expanded to grid-GISs. Elementary functions on such matrices allow for a huge range of applications. Complex problems often relatively simply can be resolved. Algorithms and visualization techniques of digital image processing can be used. The extraction of river lines and catchment areas can be seen as one example of these features. More complex hydrological modelling could take place with additional data layers, such as roughness, soil conditions, slant (for water flow velocity); rainfall layers would allow for simulation of specific rainfall characteristics, especially in conjunction with a hill exposition layer.

The pyramid data structure proves to be a simple but efficient means for grid GIS and raster data analyses, saving the administrative amount of quadtrees and similar structures. For better accuracy, after the operations have taken place in the grid structure, it is possible to make finer adjustments in regions of special interest, i.e. along the river-courses and the catchment borders, using vector-based algorithms in hybride or vector DEMs.

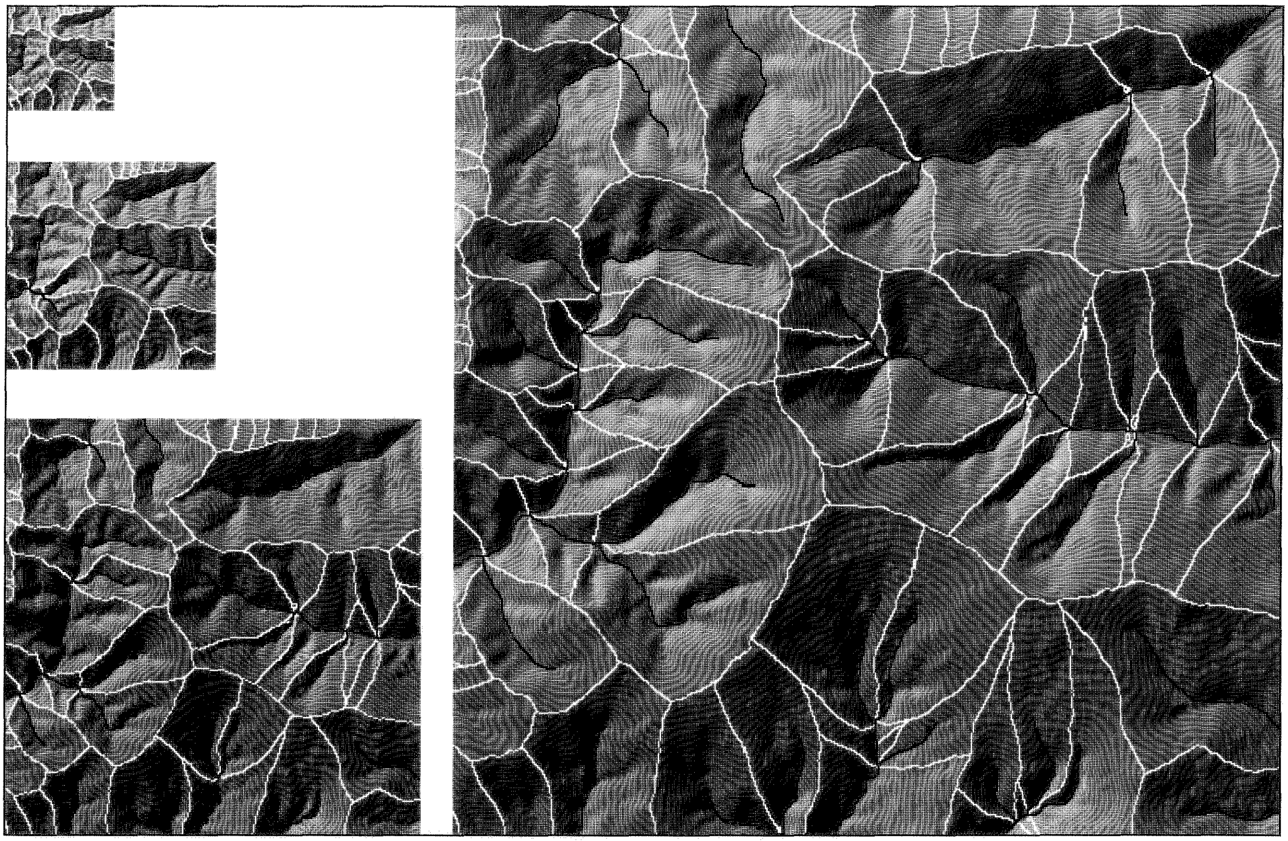

Fig. 11: Test region in 4 levels: 61x61 grid-cells with 200m grid width; 121x121 cells (100m); 241x241 (50m); 481x481 (25m).

9. REFERENCES

Horn, B. K. P., 1982. Hill shading and the reflectance map. Geo-Processing 2(1982):65-144.

Hutchinson, M.F., 1989. A new procedure for gridding elevation and stream line data with automatic removal of spurious pits. Journal of Hydrology, 106(1989):211-232.

Jenson, S.K., Domingue, J.O., 1988. Extracting topographic structure from digital elevation data for GIS analysis. Photogrammetric Engineering & Remote Sensing, 54(11):1593-1600.

Johnson, L.E., 1989. MAPHYD - A digital map-based hydrologic modeling system. Photogrammetric Engineering & Remote Sensing, 55(6):911-917.

Mark, D.M., 1979. Phenomenon-based data-structuring and digital terrain modelling. Geo-Processing, 1(1979): 27-36.

Mark, D.M., 1988. Network models in geomorphology, in Anderson, M.G. (Ed.), Modelling in Geomorphological Systems, John Wiley, Chichester, England.

Molnar, L., Waldhaeusl, P., 1991: Program System SCOP to create, maintain, and apply digital terrain models. Product information of the Inst. of Photogrammetry and Remote Sensing, The Vienna University of Technology, Austria.

Moore, I.D., Grayson, R.B., Ladson, A.R., 1991. Digital terrain modelling: A review of hydrological, geomorphological and biological applications. Hydrological Processes, 5(1):3-30.

O'Callaghan, J.F., Mark, D.M., 1984. The extraction of drainage networks from digital elevation data. Computer Vision, Graphics and Image Processing, 28:323-344.

Palacios-Vélez, O.L., Cuevas-Renaud, B., 1986. Automated river-course, ridge, and basin delineation from digital elevation data. Journal of Hydrology, 86(1986): 299-314.

Qian, J., Ehrich, R.W., Campbell, J.B., 1990. DNESYS - An Expert System for Automatic Extraction of Drainage Networks from Digital Elevation Data. IEEE Transactions on Geosciences and Remote Sensing, 28(1):29-45.

Rieger, W., 1992. Hydrologische Anwendungen des digitalen Geländemodelles. Dissertation, The Vienna University of Technology, Vienna, Austria.

Sperling, B., 1992. Darstellung der Möglichkeiten eines hybriden GIS am Beispiel einer DGM-basierten Oberflächenabflußanalyse. Geo-Information-Systems, 5(1):42-44.