

# Subpixel-Precise Extraction of Watersheds

Carsten Steger

Forschungsgruppe Bildverstehen (FG BV), Informatik IX  
Technische Universität München, Orleansstr. 34, 81667 München  
Phone: +49 (89) 48095-211, Fax: +49 (89) 48095-203  
E-mail: stegerc@informatik.tu-muenchen.de

## Abstract

*An approach to extract watersheds and watercourses, as well as their corresponding valleys and hills, from images with subpixel precision is proposed. The critical points of the terrain are essential as the starting points for the construction of these separatrices. They are extracted efficiently with subpixel precision using an approach based on derivatives of Gaussian filters. The separatrices are extracted by integrating their defining differential equation. Finally, the hills and valleys are constructed by an efficient graph search algorithm. Examples show the quality of the results that can be achieved with the proposed approach.*

## 1 Introduction

Watersheds and watercourses are important geomorphological features, which play an important role in hydrological GIS applications. Intuitively, watersheds can be regarded as the lines that separate the area where water drains to different locations. The areas that are enclosed by the watersheds are precisely the regions where water drains to the same place, and are conventionally called basins or valleys. Likewise, watercourses can be regarded as the lines where water accumulates when it drains on the terrain. If water would only run on the surface, the watercourses would be the location of the river beds.

Watersheds also play an important role for the segmentation of images, because the gray value of an image can be regarded as the height of the terrain, and many interesting features, e.g., cell walls in microscopic images, can be described by watersheds.

One of the major categories of approaches to extract watersheds are the ridge detectors, which were first proposed in the early part of the 19th century (see [2, 8] for a historical overview). [1] gives a good overview over the existing classes of ridge detectors. However, it is well known that they do not model the way water runs downhill [2], and can therefore not be used to extract watersheds.

Another theory was proposed in the second half of the 19th century by Maxwell, Jordan, and Cayley (see [4, 6] and references therein). It is based on the observation that for generic surfaces there is a unique slope line through every non-critical point of the surface. Each slope line is the solution of a first-order ordinary differential equation (ODE). Loosely speaking, all slope lines converging at the same maximum are said to form a hill, while all slope lines converging at the same minimum comprise a valley. The lines that separate the hills and valleys are called watercourses and watersheds, respectively. An important aspect of this definition is that these separatrices are given by special slope lines emanating from saddle points, and running to a maximum or minimum, respectively. An implementation roughly following this theory has been described in [7]. However, it suffers from some poor implementation choices that lead to inaccurate results and some surprising cases, in which, for example, slope lines can cross each other.

A different characterization is given in [8]. This definition basically replaces the gradient vector field in the above ODE by its dual 1-form, to obtain another differential equation, which is inexact, i.e., does not have a general integral. To integrate the DE, an integrating divisor must be found, which obeys a first-order partial differential equation (PDE) [8, 2]. The level crossings of the integrating surface of the PDE are the slope lines. Among them, some are singled out as singular solutions, which contain the watersheds and watercourses. Unfortunately, this definition is almost impossible to implement as a computer algorithm. The defining equation of the integrating divisor is a first-order PDE, which could be solved by the method of characteristic strips. However, no boundary conditions can be given. Therefore, no true implementations of this theory have been given, although some erroneously claim to be one [3].

A third way to extract watersheds from the image is to use the fact that water will accumulate at the minima of the landscape. This means that each minimum in the image defines a valley or water catchment basin. Watersheds are the boundaries between different basins. This can be imple-

mented by flooding the landscape from the minima [11].

Since all of the approaches to extract watersheds return the result only with pixel resolution, a subpixel accurate watershed and watercourse extraction algorithm is desirable. This means that either the definition of Maxwell, Jordan, and Cayley, or the definition of Rothe must be used. Since the latter definition can be implemented only with great difficulties, if it can be implemented at all, only the first definition seems to be viable.

## 2 Theory

As mentioned above, the definition of watersheds and watercourses given by Maxwell, Jordan, and Cayley is the definition that is most suitable for an implementation as a computer program. This section will describe the theory in detail, so that its implementation can be easily understood.

The definition of watersheds and watercourses, collectively called *separatrices*, regards the terrain as a surface  $f(x)$ , where  $x \in \mathbb{R}^2$ . This means that overhanging walls in mountains cannot be modeled, but since water falls down vertically at such places, this poses no restrictions on the approach. Each generic surface possesses isolated critical points, i.e., points where the gradient  $\nabla f = 0$  [4, 6]. These are the maxima (peaks), minima (pits), and saddle points (passes) of the terrain.

Every non-critical point  $p$  of the terrain lies on exactly one slope line. The point  $p$  divides the corresponding slope line into an ascending and a descending part. These two parts of the slope line are the solutions of the ODE

$$\dot{x}(t) = \pm \nabla f(x(t)) \quad (1)$$

with the initial condition  $x(0) = p$ . The ascending part of the slope line corresponds to the positive gradient. Critical points do not lie on any slope lines, but we can say that a slope line reaches a critical point  $c$  if

$$\lim_{t \rightarrow \infty} x(t) = c \quad (2)$$

With this, we can define the terms “hill” and “valley” quite intuitively. All points, from which the ascending slope line reaches the same maximum, form the *hill* that corresponds to the maximum, while all points, from which the descending slope line reaches the same minimum, form the *valley* that corresponds to the minimum. Furthermore, we can define that all points, from which the slope line reaches the same minimum and maximum, form the *slope district* of the minimum and maximum. It follows that all hills are disjoint and cover the plane  $\mathbb{R}^2$ , and likewise for valleys. Therefore, there must be curves that separate the hills and the valleys. We can define that curves that separate adjacent valleys are called *ridge lines*, while curves that separate adjacent hills are called *valley lines*. Note that the term ridge

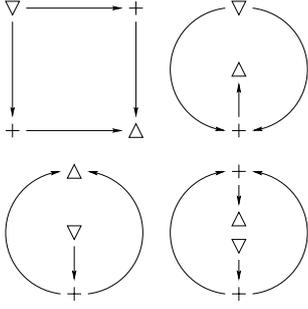
line is defined differently from most ridge detectors. The ridge lines are, as we will see below, very close to what we would intuitively call *watersheds*, i.e., lines that separate regions where water drains to different locations, while the valley lines are very close to what we would call *watercourses*, i.e., possible locations of rivers.

While the above definition of the ridge and valley lines is quite intuitive, it cannot be used for an efficient implementation, since we would have to construct the slope lines for every point of the image. However, the following observation will lead to an efficient algorithm, as we will see below. In every saddle point, four special slope lines “emanate” in the following sense: Every saddle point possesses two preferred directions, its directions of principal curvature, i.e., the directions in which the second directional derivative at the critical point obtains its minimum and maximum value, respectively. The principal directions are perpendicular to each other. The corresponding principal curvatures are of opposite sign, indicating an upward and downward curved direction. Although no slope line leaves the saddle point, since the gradient there vanishes, we can take an infinitesimally small step in the four directions defined by the two principal directions. This will lead to two ascending and two descending slope lines. The ascending slope lines necessarily reach a maximum or another saddle point, while the descending slope lines reach a minimum or another saddle point. The key observation is that the valley lines are exactly the descending slope lines emanating from the saddle points, while the ridge lines are exactly the ascending slope lines emanating from the saddle points.

The above definitions lead to the fact that for generic surfaces, every maximum is surrounded by a “ring” of valley lines, on which only minima and saddle points occur as critical points. Likewise, every minimum is surrounded by a ring of ridge lines, on which only maxima and saddle points occur as critical points.

If we regard valley and ridge lines together, it can be seen that these lines partition the plane  $\mathbb{R}^2$  completely into the slope districts. The critical points, along with the ridge and valley lines, form a graph that describes the borders of the slope districts. Here, the vertices of the graph are the critical points, while the edges are the ridge and valley lines.

It is interesting to note that all slope district boundaries are equivalent to one of the four types of slope districts shown in Figure 1 [4]. The equivalence relation is defined by inserting an arbitrary number of saddle points into the graph, where the slope district boundary makes a right angle turn at each of the inserted saddle points. The slope district type in the upper left part of Figure 1, where there are two paths from a minimum to a maximum via two different saddle points, occurs most frequently for real data. The configuration in the lower left corresponds to a crater, whereas the configuration in the upper right corresponds to



**Figure 1. The four generic types of slope district boundaries. Maxima are indicated by  $\triangle$ , minima by  $\nabla$ , and saddle points by  $+$ . Edges are drawn in the upward direction of the underlying terrain.**

an isolated mountain (imagine a crater that has been turned upside-down). The final configuration has no good real-world interpretation, and was argued to be unstable in [7].

Figure 1 gives a good example why not every ridge line is a watershed and not every valley line is a watercourse. Consider, for example, the ridge line that runs from the saddle point to the maximum in the upper right configuration. There, all water that falls onto this slope district accumulates at the minimum. Hence, the ridge line does not separate two regions where water accumulates at different places, and therefore it cannot be a watershed. The same line of reasoning shows that the valley line in the lower left configuration is not a watercourse, since water will run down the crater walls more or less equally, and hence the water does not really accumulate at a watercourse.

### 3 Extraction of Critical Points

Based on the above discussion, it is clear that the critical points of the terrain play a crucial role for the extraction of separatrices. If the saddle points of the terrain are known, one can start constructing the four special slope lines that emanate from each saddle point by integrating the ODE (1). To know when to stop integrating, one has to know the minima, maxima, and saddle points of the terrain.

Because the slope lines eventually reach the critical points, the location of these points has to be known with high accuracy in order to be able to stop the integration of the slope lines at the right time and place. More importantly, the saddle points and their principal directions have to be known with high accuracy to enable the starting of the integration. These requirements rule out pixel-based approaches to the extraction of the critical points, because the principal directions of the critical points cannot be determined accurately.

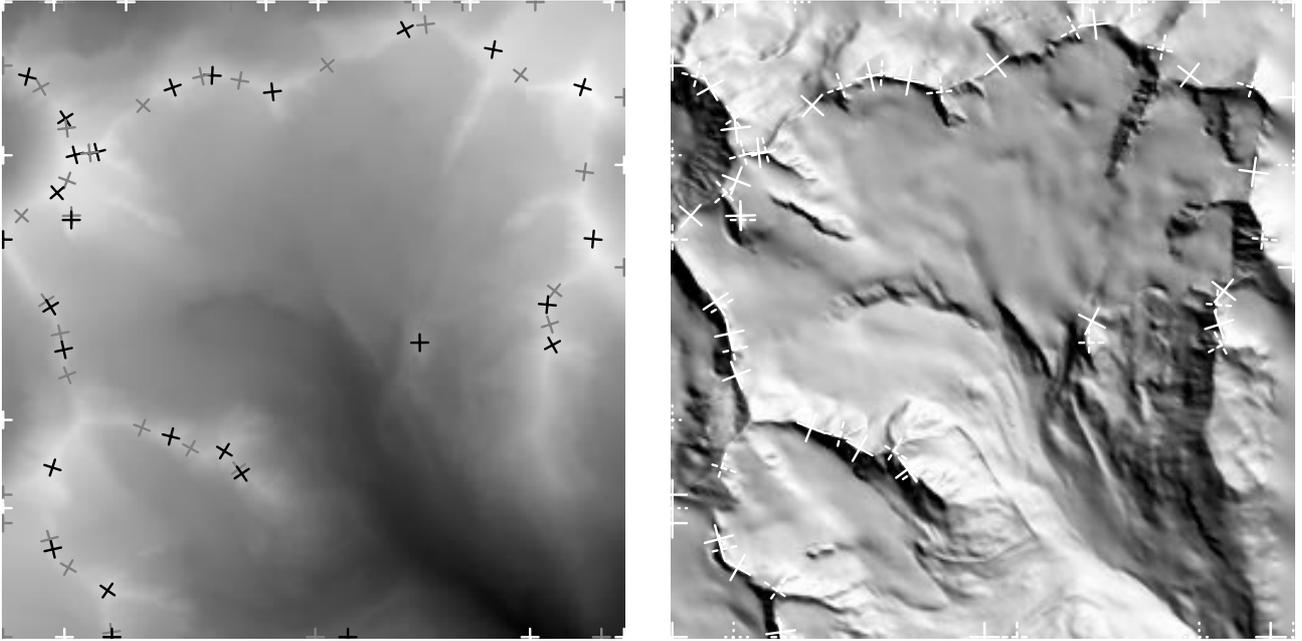
A method for the extraction of minima and maxima with subpixel accuracy was presented in [10]. It can easily be modified to extract saddle points as well. The method works by constructing a second-order Taylor polynomial for every image point. The necessary partial derivatives of the image are obtained by convolving the image with the appropriate partial derivatives of a Gaussian kernel [10, 9]. Smoothing the input data is necessary to remove noise and plateaus. If we denote the Hessian matrix, i.e., the matrix of the second partial derivatives, of the Gaussian-smoothed terrain by  $\mathbf{H}_g f$ , and the gradient of the smoothed terrain by  $\nabla_g f$ , we can extract the critical points by solving the following linear equation for every image pixel:

$$\mathbf{H}_g f \cdot x = -\nabla_g f . \quad (3)$$

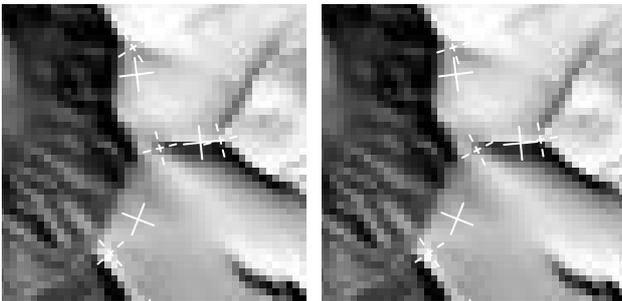
To ensure that the critical point lies within the current pixel,  $x \in [-\frac{1}{2}, \frac{1}{2}] \times [-\frac{1}{2}, \frac{1}{2}]$  must be required.

The above procedure returns the critical points with subpixel resolution. In addition, a classification into maxima, minima, and saddle points, as well as the principal directions at the critical points are needed. Both can be obtained from the Hessian matrix  $\mathbf{H}_g f$ . The eigenvalues  $\lambda_1$  and  $\lambda_2$  ( $\lambda_1 < \lambda_2$ ) of the Hessian give the required classification: A critical point is a maximum if  $\lambda_1, \lambda_2 < 0$ , a minimum if  $\lambda_1, \lambda_2 > 0$ , and a saddle point if  $\lambda_1 < 0$  and  $\lambda_2 > 0$ . Note that because we assume the surface  $f$  to be generic, the case  $\lambda_{1,2} = 0$  occurs with probability zero. Additionally, the eigenvectors  $e_1$  and  $e_2$  corresponding to the two eigenvalues give the two principal directions at each saddle point, and hence the four starting directions  $\pm e_{1,2}$  for the integration of the separatrices. Figure 2 shows the result of extracting the critical points from a digital terrain model (DTM) with 20 m pixel size of the Vernagtferner glacier. The critical points are visualized by crosses which point in their principal directions. One important point to note is that because the image is mirrored at the borders during smoothing and subsequent processing, critical points are created in the correct places for further processing at the border of the DTM automatically.

While the locations of the extracted critical points are quite good, they can be refined even further. The reason for this is that the critical points are *extrapolated* from the Taylor polynomial at each pixel's center. Because terms of order 3 and higher are neglected, the locations can contain extrapolation errors of up to about 1/10 of a pixel, especially if the critical point lies near the pixel's borders. Furthermore, because of this the Taylor polynomials do not agree at the pixel's boundaries, and can therefore not be used for interpolation purposes. However, the construction of the separatrices discussed in the next section requires a consistently interpolated surface. By consistent we mean that the gradient  $\nabla_g f = (g_x, g_y)$  of the ODE (1) must fulfill the



**Figure 2. Critical points extracted from a DTM of the Vernagtferner glacier. The left image shows the DTM visualized by gray levels, while the right image shows the DTM shaded from the north east. Maxima are displayed as black crosses in the left image and as dashed crosses in the right image, respectively, minima as white and dotted crosses, and saddle points as medium gray and solid crosses. The axes of the crosses point to their principal directions.**



**Figure 3. Comparison between the extrapolated and refined locations of the critical points in the vicinity of the peak in the north west corner of Figure 2.**

integrability condition

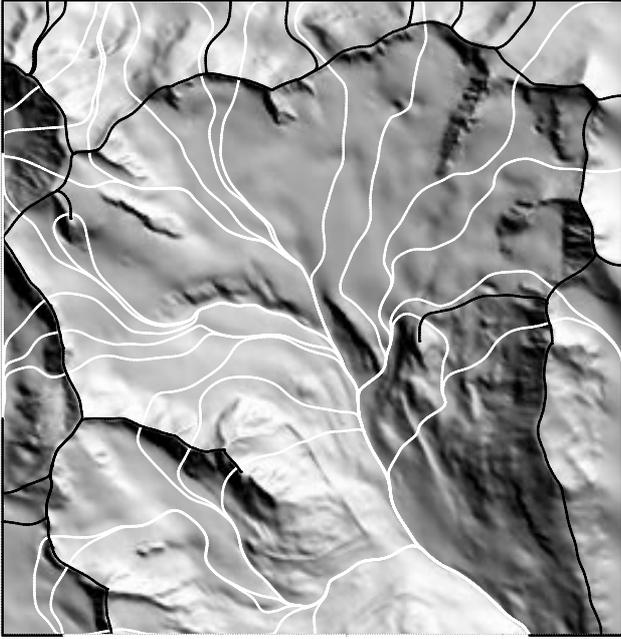
$$\frac{\partial g_x}{\partial y} = \frac{\partial g_y}{\partial x} . \quad (4)$$

Note that this implies that the terrain must have continuous derivatives up to second order. A bicubical interpolation of the terrain fulfills all these requirements, and hence is used from now on to interpolate the partial derivatives of the DTM or image at subpixel positions.

With this, the extrapolated critical points can be regarded as initial guesses for a root finding algorithm that calculates the critical points (recall that their location is given by  $\nabla_g f = 0$ ). A Newton-Raphson-type root finding algorithm [5] will locate the critical points with sufficient accuracy with few iterations. Of course, the principal directions have to be recalculated at the refined locations of the critical points. Figure 3 shows a comparison between the extrapolated and refined locations of the critical points. As can be seen, the location changes are relatively minor, but the changed principal directions indicate that the refinement step is important to get good results.

## 4 Extraction of Separatrices

With the critical points extracted, the construction of the separatrices can be done in a straightforward manner by integrating their defining ODE 1, i.e.,  $\dot{x}(t) = \nabla f(x(t))$  for ridge lines and  $\dot{x}(t) = -\nabla f(x(t))$  for valley lines. The initial condition  $x(0) = p$  needed to start the integration can be obtained by taking a small step away from the saddle points in their principal direction, i.e.,  $\pm e_1$  ( $\lambda_1 < 0$ ) for valley lines and  $\pm e_2$  ( $\lambda_2 > 0$ ) for ridge lines, as discussed in Section 2. In the current implementation, this step is 1/4 pixel. Of course, to get a topologically complete result,

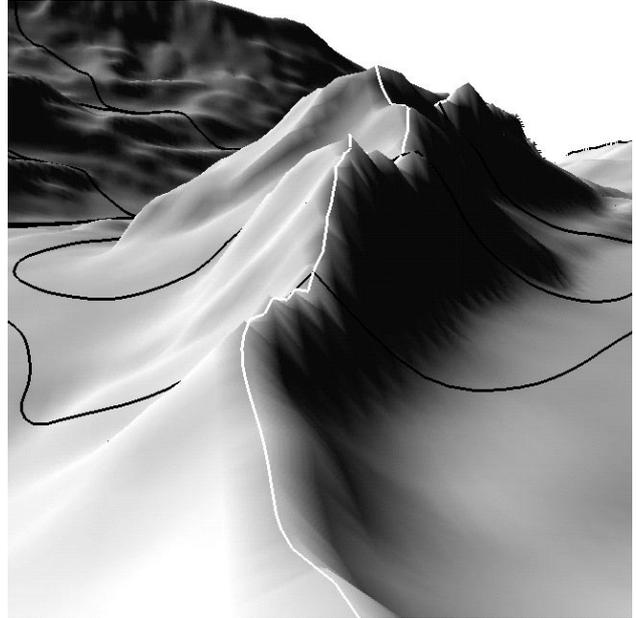


**Figure 4. Separatrices extracted from the DTM in Figure 2.**

the saddle point itself has to be added to each separatrix as well. The integration of the ODE stops if the separatrix gets close enough to another critical point (recall that it cannot get to the critical point itself). To perform the integration, a Runge-Kutta algorithm with adaptive step size control is both robust and efficient enough [5].

With this procedure, all separatrices in the interior of the DTM or image are extracted correctly. At the border of the DTM, some special treatment is necessary since it might happen that a maximum and a minimum are adjacent on the border. In such cases, no separatrix will be constructed between the two critical points. However, to ensure a complete segmentation of the DTM into regions, it is essential that in such cases a separatrix is constructed artificially. Hence, the algorithm examines all adjacent pairs of critical points and inserts a separatrix on the border wherever necessary to ensure that there is a closed ring of ridge and valley lines around the image border.

The results of applying the algorithm described above to the DTM of Figure 2 can be seen in Figure 4, which displays all the extracted separatrices. Figure 5 displays the extracted separatrices on a 3d visualization of the DTM to give an impression of the excellent subpixel accuracy of the algorithm. The keen observer will note that all of the slope districts are of the two types in the upper row of Figure 1, where the right type occurs three times. One surprising fact is that in the south west corner two separatrices approach each other very closely, but then separate again, before con-



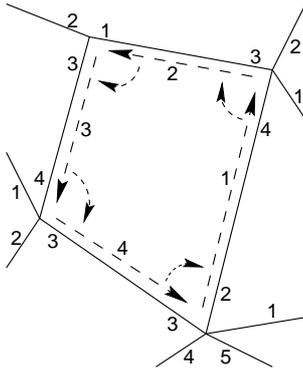
**Figure 5. 3d visualization of the separatrices extracted from the DTM in Figure 2. Watersheds are displayed in white. The view is along the ridge from the east to south of the center of the image.**

verging in the main valley in the south east. This result can only be explained by the fact that polynomial interpolation sometimes introduces oscillations, which might lead to a “phantom ridge” between the two “valleys” in this case.

## 5 Extraction of Hills and Valleys

The separatrices are often useful by themselves. However, most of the times the regions defined by the separatrices, i.e., the hills, valleys, and, to a limited extent, the slope districts, are the objects that need to be returned by the algorithm. Especially the valleys are of great importance, since their borders are exactly the watersheds of the processed DTM or image.

As discussed in Section 2, the critical points form the vertices of a graph, whose edges are the separatrices. From this graph, we can extract the slope districts by extracting the regions of the plane that are enclosed by the separatrices. Conceptually, the regions are given by “minimal” cycles in the graph, i.e., cycles that do not contain edges, which intersect the region enclosed by the cycle such that several smaller regions could be produced. The same can be said for hills and valleys. Here, we only need to insert a subset of the separatrices into the graph. For valleys, the ridge lines and the corresponding vertices need to be inserted, while for hills the valley lines are required.

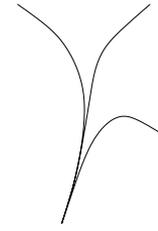


**Figure 6. Algorithm to construct the regions enclosed by the edges of a graph. The solid lines are the edges of the graph. The numbers on the edges indicate the counterclockwise sorting order of the edges in each vertex. The dashed lines are the edges that enclose the region in counterclockwise direction. The numbers on the dashed lines, along with the dotted arrows indicate the order in which the region is constructed.**

Thus, in order to extract the regions defined by the separatrices, a bidirectional graph, i.e., a graph that for every separatrix contains the edge from the saddle point to the critical point it reaches as well as the corresponding back edge, needs to be constructed. To see how the regions enclosed by the graph can be computed, assume for the moment that all edges, i.e., separatrices, are straight line segments. Then, we could easily sort the edges of the graph in each vertex according to the edge direction, i.e., we could order the edges counterclockwise (see Figure 6).

With this, the algorithm to construct the regions is relatively straightforward. To construct a region, take the first unprocessed edge of the graph, and look up the vertex to which it leads, i.e., the tip of the edge. From the edges that leave the vertex, select the one that precedes the incoming edge in the ordering. Mark the incoming as processed, and continue in the same manner until the edge selected at the current vertex is already marked as processed, i.e., until you have formed a cycle. It is easy to see that, by construction, the cycles computed by this algorithm are minimal in the above sense. Obviously, all regions have been constructed if there are no more unprocessed edges.

As can be seen, the crucial part of the algorithm is the counterclockwise ordering of the edges in each vertex. While this is easy to define for straight lines, unfortunately for separatrices this is a rather complicated problem since often several separatrices reach a critical point in the same direction, i.e., are tangential to each other, as shown in Figure 7. Therefore, the ordering cannot be determined locally



**Figure 7. Often the separatrices reach a critical point tangential to each other. Therefore, the ordering of the separatrices in the vertices is non-trivial.**

at each vertex, e.g., based on angles. An angle-based ordering can only be used for separatrices that enter the critical point non-tangentially. For tangential separatrices, the ordering must be done based on the criterion that a vertex “eventually lies to the left” of another vertex. Based on Figure 7, the meaning of this criterion is fairly obvious for a human. In the computer implementation, this criterion can be defined based on the directions of the separatrices at the point at which they move apart by a sufficient distance.

Figure 8 shows the valleys and hills that are extracted from the DTM in Figure 2 with the proposed approach. As can be seen, all regions have been computed correctly. Note that the ridge lines that are no watersheds and the valley lines that are no watercourses have been removed from the result. This was done by successively removing the adjacent edge pairs on the cycles that correspond to the same separatrices.

Another example of the results obtainable with the proposed approach is given in Figure 9. Here, watersheds were extracted from an image of human skin cells. Note again the accuracy and completeness of the segmentation.

## 6 Conclusions

This paper proposes an approach to extract watersheds and watercourses from grid DTMs and digital images with subpixel accuracy. The approach is based on the theory put forward by Maxwell, Caley, and Jordan. For the first time, a consistent and efficient implementation of this theory is developed. It rests on the fact that the critical points of the terrain, most importantly the saddle points, and their principal directions can be found efficiently with subpixel accuracy. The separatrices are then constructed with subpixel accuracy by interpolating the terrain appropriately so that a standard ODE integration algorithm can be used. Finally, the hills, valleys, and slope districts are constructed by an efficient graph search algorithm.

One of the consequences of the algorithm and of the morphology of the earth is that often separatrices, especially

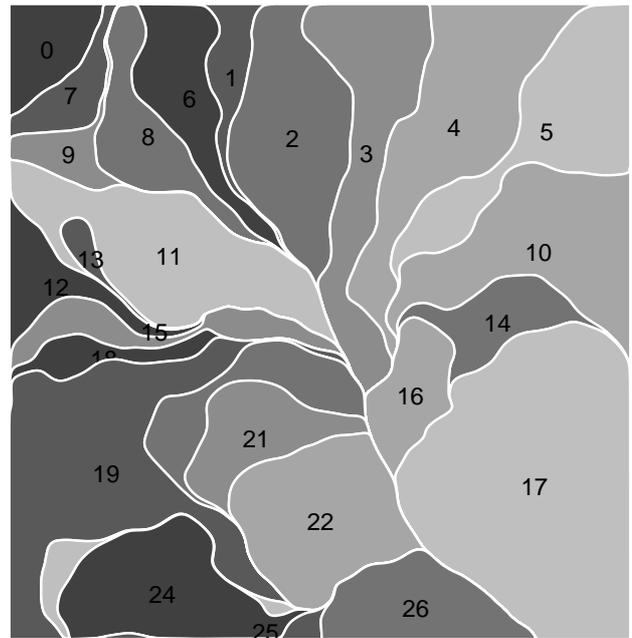
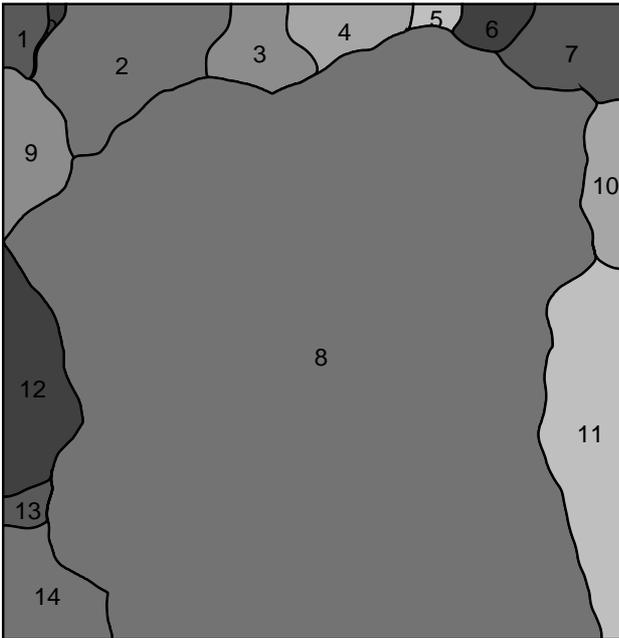


Figure 8. Valleys and hills extracted from the DTM in Figure 2.

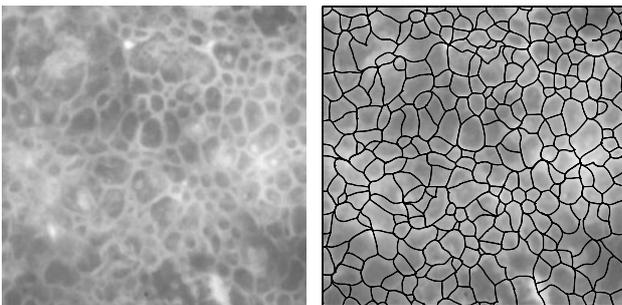


Figure 9. Watersheds extracted from an image of human skin cells.

watercourses, join tangentially long before the corresponding critical point is reached. This leads to elongated region with line-like appendages of nearly zero area. Although this results from the definitions of the separatrices, it is not intuitive from a user's standpoint. Therefore, it might be useful to insert "confluence points" at the appropriate locations into the graph. However, this is a quite complicated problem since it has to be done in a topologically sound manner.

## References

- [1] D. Eberly, R. Gardner, B. Morse, S. Pizer, and C. Scharlach. Ridges for image analysis. *Journal of Mathematical Imaging and Vision*, 4:353–373, 1994.
- [2] J. J. Koenderink and A. J. van Doorn. Two-plus-one-dimensional differential geometry. *Pattern Recognition Letters*, 15(5):439–443, May 1994.
- [3] A. M. López and J. Serrat. Tracing crease curves by solving a system of differential equations. In B. Buxton and R. Cipolla, editors, *Fourth European Conference on Computer Vision*, volume 1064 of *Lecture Notes in Computer Science*, pages 241–250, Berlin, 1996. Springer-Verlag.
- [4] L. R. Nackman. Two-dimensional critical point configuration graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(4):442–450, Apr. 1984.
- [5] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, Cambridge, 2nd edition, 1992.
- [6] J. H. Rieger. Topographical properties of generic images. *International Journal of Computer Vision*, 23(1):79–92, Jan. 1997.
- [7] P. L. Rosin, A. C. F. Colchester, and D. J. Hawkes. Early image representation using regions defined by maximum gradient paths between singular points. *Pattern Recognition*, 25(7):695–711, 1992.
- [8] R. Rothe. Zum Problem des Talwegs. In *Sitzungsberichte der Berliner Mathematischen Gesellschaft*, volume XIV, pages 51–68, Leipzig, 1915. B. G. Teubner Verlag.
- [9] C. Steger. An unbiased detector of curvilinear structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(2):113–125, Feb. 1998.
- [10] C. Steger. *Unbiased Extraction of Curvilinear Structures from 2D and 3D Images*. Dissertation, Fakultät für Informatik, Technische Universität München, 1998. Herbert Utz Verlag, München.
- [11] L. Vincent and P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, June 1991.