1. Watersheds

The watershed terminology is adapted from topographic relief or elevation concepts as they apply to hydrology. The method has also developed from mathematical morphology [Dougherty, 1992, pp. 48] . This is a region formation method that uses the concepts of watersheds and catchment basins. The basic idea is to flood the image data that is considered to be elevation data. As the image floods, the catchment basins will fill with water starting from local minimums. At points where the water from catchment basins meet, create a dam or division. This will segment the image data [Vincent and Soille, 1991; Sonka, Hlavac, and Boyle, 1999, pp. 186]. The method tends to over segment the image. That is, the regions are smaller than regions that correspond to the objects in the image. The method can be applied when the image data are elevation data. Later we will see that it can also be useful as a segmentation method with intensity image data. Whether we are using elevation data or intensity data the gray-level stands for elevation in this method. Recall that p us a pixel i.e. p = (x,y) and N_p are the neighbors of p.

Let us now give some definitions. It is useful to consider the image as a graph in these discussions. The neighbors of a pixel depend upon whether one is considering 4-neighbors or 8-neighbors. The digital image is $G = (D, A_r, g)$ where D is the digital grid, A_r is the set of arcs between pixels, and g(p) is the gray-scale function that is considered to be elevation. A level-component at gray-level h is a connected component of G say lev_h where the pixels all have gray-level g(p)=h. The boundary of lev_h is the set of p in lev_h such that a neighbor, in the graph, is not in lev_h. A neighbor has a different gray-level. The lower boundary is the set of p with a neighbor with a lower gray-level than g(p). The interior is the set of points in lev_h that are not on the boundary. A descending path is a path, a sequence of pixels, such that the gray-level does not rise on the path. A regional minimum or minimum at level h is a level-component of h where the lower boundary is empty. It has no neighbors with a lower gray-level.

Minimums. M is a minimum of g if M is a plateau of pixels with value k. All the pixel in M have gray-level k. And one cannot reach a lower altitude without climbing higher in gray-level values.
$\forall p \; \varepsilon \; M, \forall q \notin M \quad where \quad g(q) \le g(p)$ and for every path
pa=$(p_o, p_1, ..., p_l)$ where $p_o = p$, $p_l = q$ there is an i such that
$g(p_i) > g(p_o) = g(p)$.

Consider the following example.

y								
7	9	9	9	9	9	9	9	9
6	9	8	8	8	8	8	9	9
5	9	7	4	8	3	7	8	9
4	9	7	1	8	7	7	1	9
3	9	7	2	6	2	1	1	9
2	9	8	8	8	2	1	2	9
1	9	8	9	9	8	9	8	9
0	9	9	9	9	9	9	9	9
x	0	1	2	3	4	5	6	7

Figure 1. Minimums

The above data has the following minimums.
minimum_a {(2,4)}
minimum_b {(5,2), (5,3), (6,3), (6,4)}
minimum_c={(4,5)}

Let k a gray-level and $T_k = \{p \mid g(p) \le k\}$, where p is a pixel. These are the pixels with gray-levels $\le$ k. A path in a set A between pixels p, q is a sequence of pixels $pa=\{p_0, p_1, p_2, ..., p_k\}$ where $p_o = p$, $p_k = q$ and each p_i is in A. In addition, each p_i, p_{i+1} pair are neighbors. The length of the path is $len(p_a) = k$. The neighbors may be defined as 4 or 8 neighbors. Another useful term is geodesic distance. The geodesic distance from point p to point q in set A is defined by

$$d_A(p,q) = \inf \{\text{len(pa)} \mid \text{pa is a path between p and q in set A}\}.$$

That is, if pa = $(p_o, p_1,$ $, p_k)$, then each p_i is in A. It is the length of the shortest path between p and q that lies within set A.

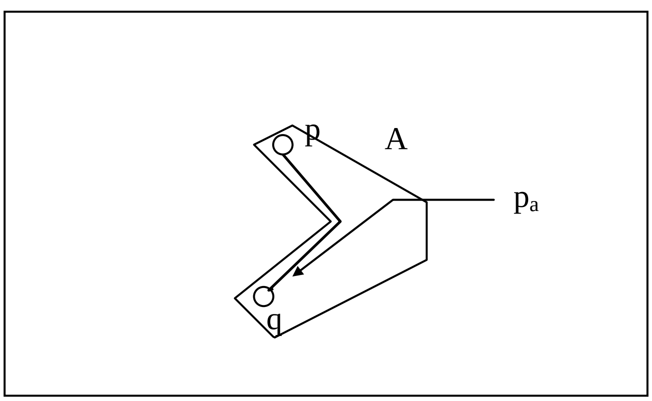


Figure 2. Path within a Set.

Now let us define the geodesic influence zone of a set B with connected components. $B = \bigcup_i B_i$

where each B_i is connected and a subset of A. The geodesic influence zone of subset B_i in set A is defined as $iz_A(B_i)=\{p \in A \mid \forall j \;\; d_A(p,B_i) < d_A(p,B_j) \quad i \neq j\}$ [Vincent and Soille, 1991]. It is the set of points in set A such that the geodesic distance from the point p to B_i in A is less than the geodesic distance from point p to any other component of B.

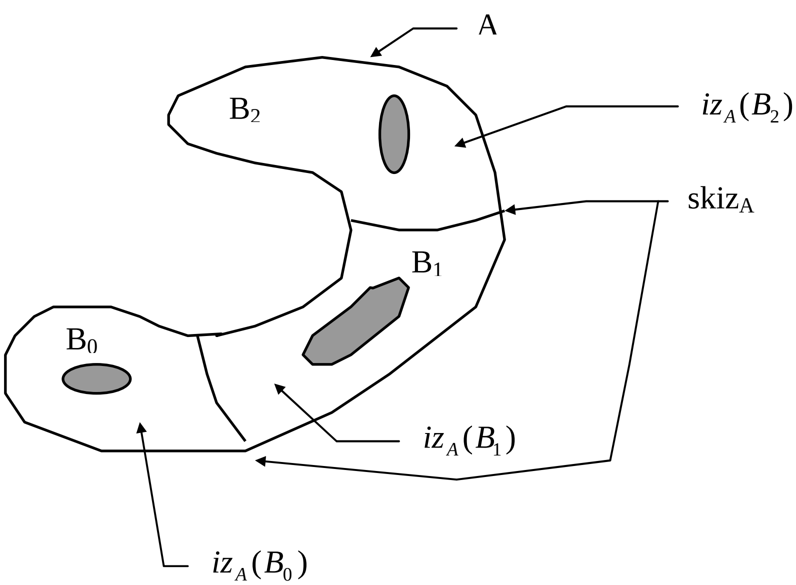


Figure 3. Geodesic Influence Zone

The term $skiz_A$ (B) = (A - iz_A(B)) which are the elements in A which are not in iz_A(B). Here $iz_A(B)=\bigcup_i iz_A(B_i)$. These are the points not in any influence zone. They are points equidistant from two or more connected components.

A catchment basin is defined from the minimum regions [Vincent and Soille, 1991]. Let M be a minimum region. Then a catchment basin C(M) consists of the pixels from which there is a downhill path to M. It is useful to define the following term. Let k be a gray-level, then $C_k(M)=\{p \in C(M) \mid g(p) \leq k\} = \left(C(M) \bigcap T_k\right)$. These are the pixels at gray-level k or less in the catchment basin C(M). The set $C_k(M)$ is useful in calculating C(M).

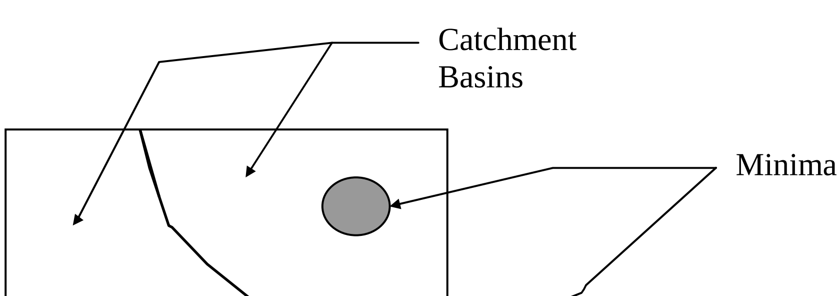
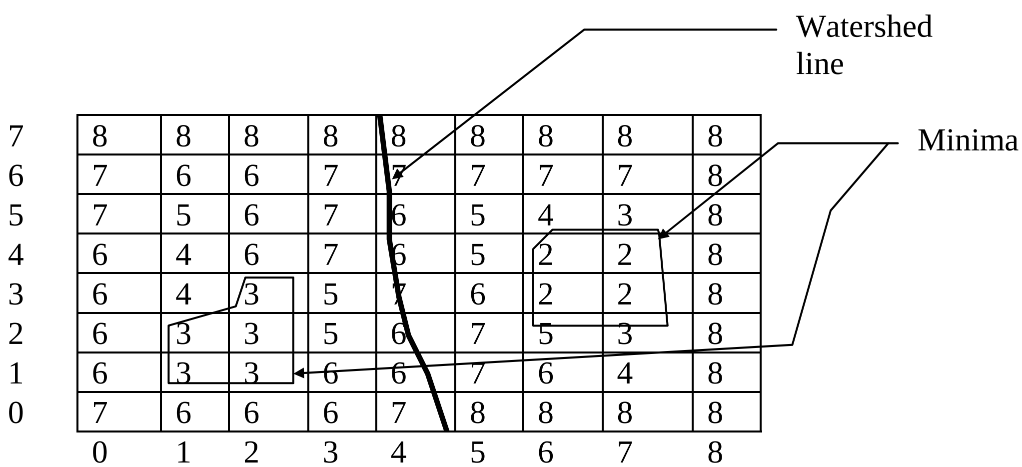


Figure 4. Catchment Basins

Consider the following example. The minimums and catchment basins are indicated.



7	8	8	8	8	8	8	8	8	8
6	7	6	6	7	7	7	7	7	8
5	7	5	6	7	6	5	4	3	8
4	6	4	6	7	6	5	2	2	8
3	6	4	3	5	7	6	2	2	8
2	6	3	3	5	6	7	5	3	8
1	6	3	3	6	6	7	6	4	8
0	7	6	6	6	7	8	8	8	8
	0	1	2	3	4	5	6	7	8

Figure 5. Example Catchment Basins

The following is a method for calculating the catchment basins using an immersion procedure. The basic procedure is to start with the lowest minimum regions and then add the pixels at the next gray-level that is in the catchment basin of each minimum. At each pixel where the catchment basins would merge, then build a boundary between the basins.

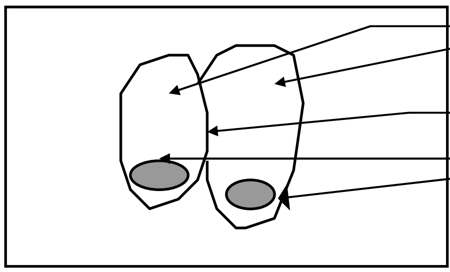


Figure 6. Method for Forming Catchment Basins

Consider some of the details of the processing. Start with the minimum with the lowest gray-level. Let k_{min} be this gray-level corresponding to the lowest minimum. Let S_{kmin} be the set consisting of pixels with gray-level kmin. This is the start of the catchment basins that just contains the lowest minimum sets. The only points in the catchment basins are the points in the minimum sets. We now move up to the next gray-level. At the next gray-level we will expand the catchment basins already formed or else find new minimums that will form the beginning of a new catchment basin.

Now consider the next gray-level and let T_{kmin+1} be the pixels with gray-levels less than of equal to k_{min+1}. Then $S_{kmin} \subseteq T_{kmin+1}$. The points in T_{kmin+1} are the candidate points to go in S_{kmin+1} at the next level. Let B be a connected component of T_{kmin+1}. There are three possible situations as shown in the following figure.

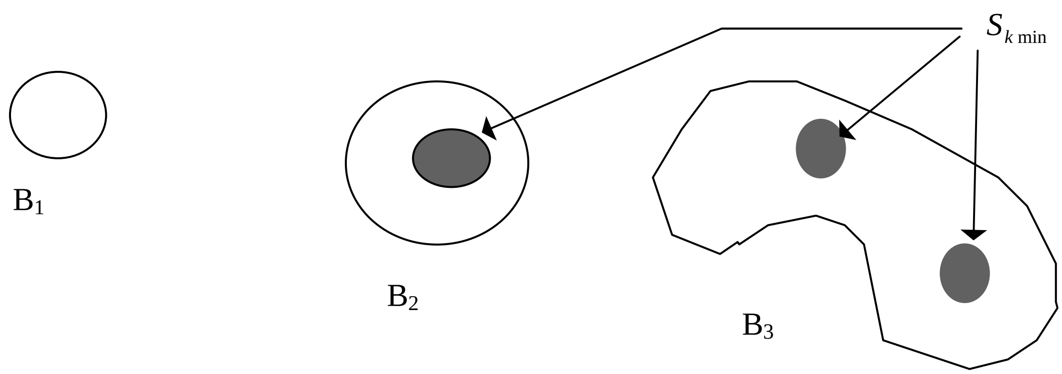


Figure 7. Possible Components in Forming Catchment Basins

The dark areas of the figure are S_{kmin} . The B's are the connected components of T_{kmin+1} . Consider the first case where

1. $B_1 \cap S_{k\,\min} = \varphi$ the empty set. In this case, B_1 is a minimum. Now $\forall p\ \varepsilon\ B_1$ and $p \notin S_{k\,\min} \Rightarrow g(p) \geq (k\min + 1)$. But since $p \in T_{k\,\min+1} \Rightarrow g(p) \leq k\min + 1$. Therefore, g(p) = kmin+1. All the surrounding pixels of B_1 do not belong to T_{kmin+1} and ,therefore , must have a gray-level > kmin+1. We have found a new minimum. This forms the beginning of a new catchment basin.

2. Now consider case 2. In this case $B_2 \cap S_{k\,\min} \neq \varphi$ and is connected. This means that B_2 having a gray level kmin+1 belongs to the catchment basin associated with the minimum $B_2 \cap S_{k\,\min}$.

3. Now consider case 3. Here $B_3 \cap S_{k\,\min} \neq \varphi$ and the intersection is not connected. This implies that B_3 contains catchment basins formed from different minima. Call these different catchment basins $D_1, D_2, ..., D_n$. The question to be resolved is the way to take the points in B_3 and put them in the different catchment basins. Consider each D_i and how to expand it at the next level. The best choice for the next iteration is the geodesic influence zone of D_i inside B_3, namely $Ciz_{B_3}(D_i)$. Add the points in the geodesic influence zone to D_i. Hence, the catchment basin at the next level is $C_{(k\,\min+1)}(D_i) = iz_{B_3}(D_i)$. We now define $S_{(k\,\min+1)} = iz_{(T_{k\,\min+1})}(S_{k\,\min})$. Note that $S_{k\,\min}$ is a set of connected components the D's. And $T_{(k\,\min+1)}$ is a set of connected components the B's. Apply the $iz_{B_j}(D_i)$ geodesic influence zone of D_i in B_j and for all cases to get $S_{(k\,\min+1)}$ the next immersion level.

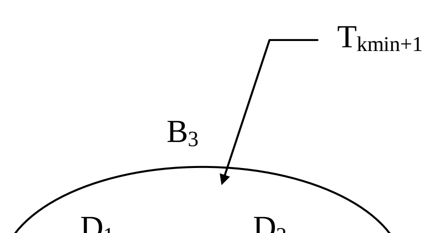


Figure 8. Formation of Catchment Basin at kmin+1

All the cases have been considered. The catchment basins are then $S_{k\max}$ obtained after the iteration of the process. The watersheds lines are the complement of this set e.g. the pixels not in a catchment basin.

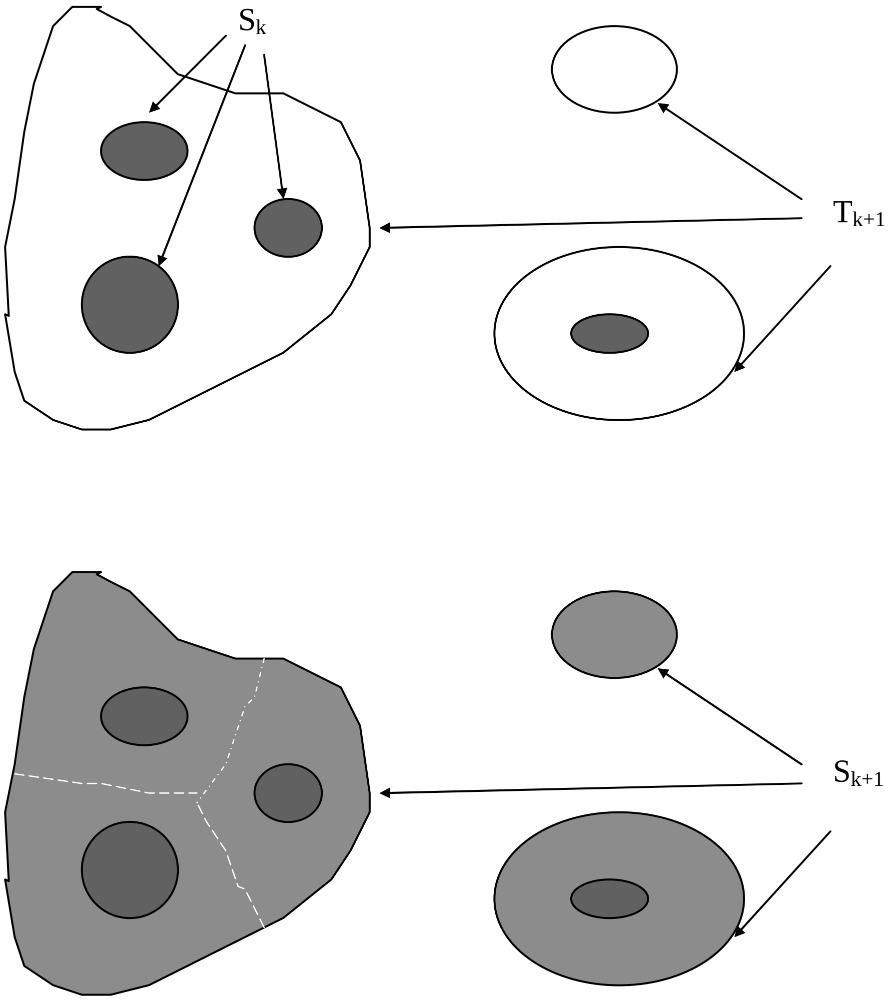


Figure 9 . Formation of Catchment Basin at level kmin+1

1.1 Implementation

Let us now consider the implementation of the method. First construct a data structure which is useful in the calculations. This forms a sorting function. We build a data structure consisting of a histogram and a list of pixels at each gray level. This gives direct access to pixels with gray level k. The histogram with pixels is defined to be $h(k) = \{i, (x_1, y_1), (x_2, y_2), \ldots\}$ here i is the number of pixels with gray level k and $\{(x_1, y_1), (x_2, y_2), \ldots\}$ are the pixels. Next consider the flooding step and the calculation of geodesic influence zones [Vincent and Soille, 1991].

Suppose flooding has progressed to level h. Then every catchment basin whose minimum is $\leq h$ has a unique label. For the next level, one then needs to compute geodesic influence zones of the labeled regions within the T_{h+1} set of pixels. Consider pixels at gray-level h+1. These are the pixels that must be included in the geodesic influence zones of the already labeled regions. One needs to compute the geodetic distances of each of these pixels to its nearest component, *D*, of S_h where *D* is a labeled region. Do the calculation by expanding each component of S_h . In order to do this we define a region where each pixel has gray level h+1. Let $W = \{p \mid g(p) = h+1\}$ be a region with gray-level h+1. Now consider the following steps.

Step 1. Consider all pixels p in W where p has a neighbor in D a component of S_h. These pixels are of geodesic distance one from a component of S_h. If p has two neighbors in different components say D_1 and D_2 of S_h, then p is not a candidate for inclusion in the geodesic influence zone of any D component of S_h. This will leave a boundary between zones. Otherwise, place p in the geodesic influence zone of D where D is in the neighborhood of p. After step one each component of S_h has been updated with all p in W of geodesic distance one from D a component of S_h. The following figure shows this process. Call this new component of D as D'. We now iterate the process.

Step 2. Consider all pixels p in W where p has a neighbor in D' a component of S_h created in step one. These pixels are of geodesic distance one from a component D' of S_h created in step one. If p has two neighbors in different components D'_1, D'_2 of S_h, then p is not a candidate for inclusion in the geodesic influence zone of the D' component of S_h. Otherwise, place p in the geodesic influence zone of D' where D' is in the neighborhood of p. This process also places p in the geodesic influence zone of D.

Step 3. Iterate the process until no more points at gray-level h+1 can be added to the existing catchment basins. After this iteration only the minimums at level h+1 have not been labeled.

Step 4. Label the minimums at gray-level h+1.

Step 5. Increase h by one and iterate the entire process.

It might be noted that the process of including points in the catchment basins of steps 1 and 2 provides for a boundary between the two catchment basins. The criterion says a point is on the boundary between two catchment basins if it has neighbors in at least two different catchment basins.

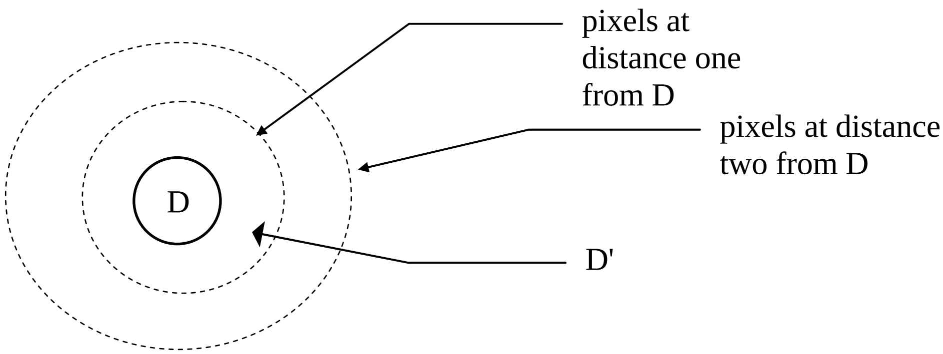


Figure 10. Process for Adding Points to Catchment Basin

Consider the following example. The original image data and the catchment basins at the different levels are shown.

image data

7	8	8	8	8	8	8	8	8	8
6	7	6	6	7	7	7	7	7	8
5	7	5	6	7	6	5	4	3	8
4	6	4	6	7	6	5	2	2	8
3	6	4	3	5	7	6	2	2	8
2	6	3	3	5	6	7	5	3	8
1	6	3	3	6	6	7	6	4	8
0	7	6	6	6	7	8	8	8	8
	0	1	2	3	4	5	6	7	8

gray-level 3 step

7									
6									
5								3	
4							a	a	
3			3				a	a	
2		3	3					3	
1		3	3						
0									
	0	1	2	3	4	5	6	7	8

gray-level 4 step

7									
6									
5							4	a	
4		4					a	a	
3		4	b				a	a	
2		b	b					a	
1		b	b					4	
0									
	0	1	2	3	4	5	6	7	8

gray-level 5 step

7									
6									
5		5				5	a	a	
4		b				5	a	a	
3		b	b	5			a	a	
2		b	b	5			5	a	
1		b	b					a	
0									
	0	1	2	3	4	5	6	7	8

gray-level 6 step

7									
6		6	6						
5		b	6		6	a	a	a	
4	6	b	6		6	a	a	a	
3	6	b	b	b		6	a	a	
2	6	b	b	b	6		a	a	
1	6	b	b	6	6		6	a	
0		6	6	6					
	0	1	2	3	4	5	6	7	8

gray-level 7 step

7									
6	7	b	b	7	7	7	7	7	
5	7	b	b	7	a	a	a	a	
4	b	b	b	7		a	a	a	
3	b	b	b	b	7		a	a	
2	b	b	b	b		7	a	a	
1	b	b	b	b	b	7	a	a	
0	7	b	b	b	7				
	0	1	2	3	4	5	6	7	8

gray-level 8 step

7	8	8	8	8	8	8	8	8	8
6	b	b	b		a	a	a	a	8
5	b	b	b		a	a	a	a	8
4	b	b	b			a	a	a	8
3	b	b	b	b			a	a	8
2	b	b	b	b			a	a	8
1	b	b	b	b	b		a	a	8
0	b	b	b	b		8	8	8	8
	0	1	2	3	4	5	6	7	8

final result

7	b	b	b		a	a	a	a	a
6	b	b	b		a	a	a	a	a
5	b	b	b		a	a	a	a	a
4	b	b	b			a	a	a	a
3	b	b	b	b			a	a	a
2	b	b	b	b			a	a	a
1	b	b	b	b	b		a	a	a
0	b	b	b	b			a	a	a
	0	1	2	3	4	5	6	7	8

Figure 11. Image Data and Catchment Basins

1.2 Binary Images

Consider now the segmentation of binary objects [Vincent and Soille, 1991]. The following image shows overlapping objects in a binary image.

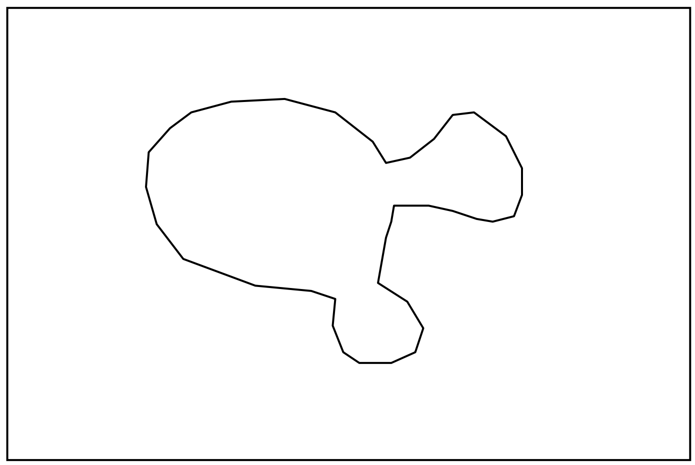


Figure 12. Overlapping Objects in a Binary Image

If we want to use a watershed method, we need to convert the binary image to an elevation image. We can do this by giving each pixel a gray-level that corresponds to its distance from the background region. We need to generate an image where g(x,y) is the distance of the pixel *p* to the background for any pixel *p* inside the object.

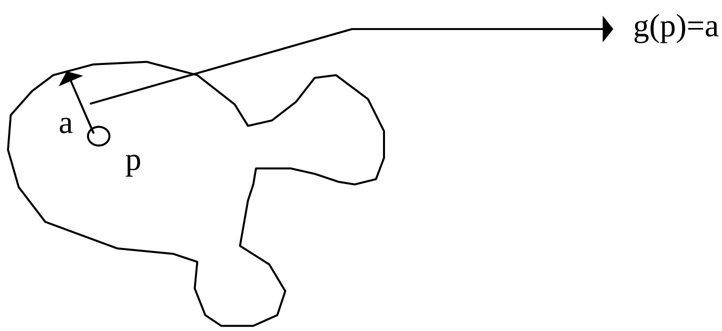


Figure 13. Distance from a point to the Boundary

In the following figure the dark areas would be the pixels with large gray-levels. They are far from the border with the background region.

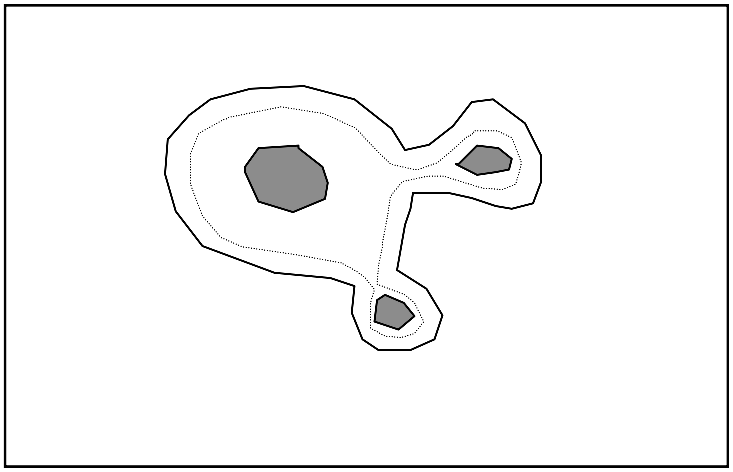


Figure 14. Gray-level Distance Image

The following figure shows a desired segmentation of the image.

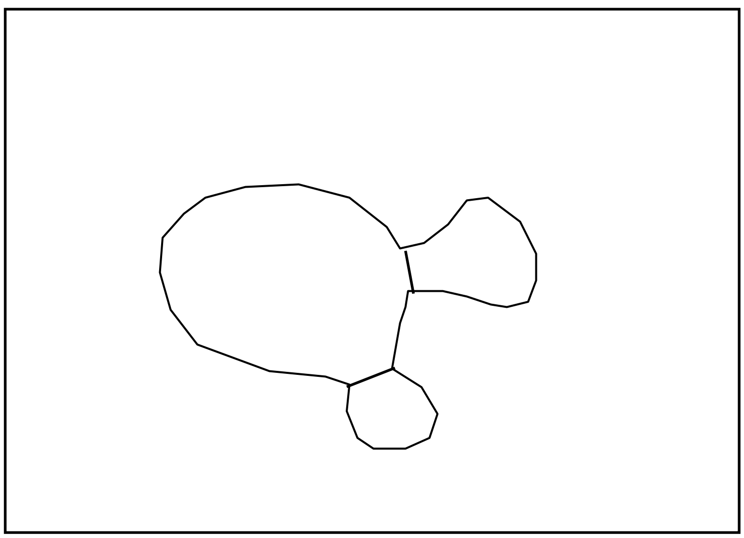


Figure 15. Desired Segmentation of Binary Image

The following example will demonstrate the calculations.

y \ x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
13					1	1	1											
12				1				1					1	1	1	1	1	
11			1						1	1	1	1						1
10		1																1
9		1																1
8		1																1
7		1												1			1	
6			1										1		1	1		
5				1						1	1	1						
4					1				1									
3				1						1								
2				1						1								
1					1				1									
0						1	1	1										

Figure 16. Example of Binary Image

13

y \ x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
12					1	1	1											
11				1	2	2	2	1					1	1	1	1		
10			1	2	3	3	3	2	1	1	1	1	2	2	2	2	1	
9			1	2	3	4	4	3	2	2	2	2	3	2	3	2	1	
8			1	2	3	4	4	3	3	3	3	3	2	1	2	2	1	
7			1	2	3	3	3	3	3	2	2	2	1		1	1		
6				1	2	2	2	2	2	1	1	1						
5					1	1	2	1	1									
4						1	2	1										
3					1	2	3	2	1									
2					1	2	2	2	1									
1						1	1	1										
0																		

Figure 17. Distance Values for Binary Image

We are using 4-connectivity for distance in the above calculations. The following figure has each value subtracted from 5 to get into the form for the watershed algorithm.

13																		
12					4	4	4											
11				4	3	3	3	4					4	4	4	4		
10			4	3	2	2	2	3	4	4	4	4	3	3	3	3	4	
9			4	3	2	1	1	2	3	3	3	3	2	3	2	3	4	
8			4	3	2	1	1	2	2	2	2	2	3	4	3	3	4	
y 7			4	3	2	2	2	2	2	3	3	3	4		4	4		
6				4	3	3	2	3	3	4	4	4						
5					4	4	3	4	4									
4						4	3	4										
3					4	3	2	3	4									
2					4	3	3	3	4									
1						4	4	4										
0																		
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

x

Figure 18. Elevation Values for Binary Image

1.3 Gray-Scale Intensity Images.

Now consider gray-scale images. If we want to use a watershed method on gray-scale images, then we must preprocess the images in order to obtain a desirable segmentation. We need to have large values of the image function at the boundaries of objects for the watershed method to produce a segmentation at the boundaries. The gradient of the image is the obvious preprocessing step.

Let $g'(x,y) = grad[g(x,y)] = \nabla g(x,y) = (g_x, g_y)$ be the gradient of g. We then use the watershed method on g' [Vincent and Soille, 1991]. The gradient will be large at the boundaries of objects and low on the interiors of objects that appear in the image. The minimums are $\{M_k\}$ of g'. Each minimum corresponds to a homogeneous area of g that will tend to occur on the interior regions of objects. The catchment basins are $C(M_k)$ for each minimum. The catchment basins are bounded by crest lines where the gradient function is large which corresponds to the boundaries between objects. Therefore, the watershed segmentation will create boundaries where the gradient function is large as desired. The result depends to a certain extent upon the details of the function used to estimate the gradient.

Consider the following example.

3	2	2	1	1	1	1	1
5	3	2	2	1	1	1	1
7	6	3	2	2	1	1	1
7	7	5	2	1	1	1	1
8	7	7	2	2	1	1	1
8	8	8	3	2	1	1	1
8	8	6	4	2	1	2	2
8	8	8	5	1	1	1	1

original image

Figure 19. Gray-Scale Image

The next image is the gradient of the image. The gradient is obtained using the Sobel gradient operators. The 0's at the boundary are a result of boundary effects and are ignored in the application of the watershed method.

0	0	0	0	0	0	0	0	
0	2.13	1.24	0.707	0.53	0.177	0.	0	
0	2.3	2.15	0.952	0.5	0.25	0.	0	gradient
0	1.43	2.63	1.82	0.5	0.25	0.	0	image
0	0.901	2.69	2.61	0.729	0.395	0.	0	using
0	0.395	2.38	2.65	1.03	0.395	0.395	0	sobel
0	0.5	2.02	2.65	1.5	0.177	0.25	0	operator
0	0	0	0	0	0	0	0	

Figure 20. Gradient Magnitude of the Image

The watershed method works with integer valued data. The next image is the gradient image scaled between 0 and 9.

0	0	0	0	0	0	0	0	
0	7	4	2	2	1	0	0	gradient image
0	8	7	3	2	1	0	0	scaled
0	5	9	6	2	1	0	0	between
0	3	9	9	2	1	0	0	0-9
0	1	8	9	3	1	1	0	
0	2	7	9	5	1	1	0	
0	0	0	0	0	0	0	0	

Figure 21. Integer Scaled Gradient Image

The next images shows the beginning of the watershed algorithm with the first minimum marked.

0	0	0	0	0	0	0	0
0	7	4	2	2	1	0	0
0	8	7	3	2	1	0	0
0	5	9	6	2	1	0	0
0	3	9	9	2	1	0	0
0	1	8	9	3	1	1	0
0	2	7	9	5	1	1	0
0	0	0	0	0	0	0	0

zero minimum of gradient image

Figure 22. First Minimum

The next figure shows the next gray level catchment level pixels added. Note that a new minimum at gray-level 1 is added to start a new catchment basin.

0	0	0	0	0	0	0	0
0	7	4	2	2	1	0	0
0	8	7	3	2	1	0	0
0	5	9	6	2	1	0	0
0	3	9	9	2	1	0	0
0	1	8	9	3	1	1	0
0	2	7	9	5	1	1	0
0	0	0	0	0	0	0	0

catchment basin of 0 minimum for gray-level 1 plus new minimum with a 1

Figure 23. Adding the next Level to the First Minimum

The process continues gray-level at a time until one completes gray-level 6 that is shown in the following figure.

0	0	0	0	0	0	0	0
0	7	4	2	2	1	0	0
0	8	7	3	2	1	0	0
0	5	9	6	2	1	0	0
0	3	9	9	2	1	0	0
0	1	8	9	3	1	1	0
0	2	7	9	5	1	1	0
0	0	0	0	0	0	0	0

catchment basin for 0 minimum and 1 minimum for gray-level 6

Figure 24. Catchment Basins Through Gray-Level 6

The next figure shows the extension to gray-level 7.

0	0	0	0	0	0	0	0
0	7	4	2	2	1	0	0
0	8	7	3	2	1	0	0
0	5	9	6	2	1	0	0
0	3	9	9	2	1	0	0
0	1	8	9	3	1	1	0
0	2	7	9	5	1	1	0
0	0	0	0	0	0	0	0

catchment basin for 0 minimum and 1 minimum for gray-level 7

Figure 25. Catchment Basins for Gray-Level 7

The next figure shows the catchment basins formed by adding gray-level 8. This is the final segmentation.

0	0	0	0	0	0	0	0
0	7	4	2	2	1	0	0
0	8	7	3	2	1	0	0
0	5	9	6	2	1	0	0
0	3	9	9	2	1	0	0
0	1	8	9	3	1	1	0
0	2	7	9	5	1	1	0
0	0	0	0	0	0	0	0

catchment basin
for 0 minimum
and 1 minimum
for gray-level 8

Figure 26. Catchment Basins Through Gray-Level 8

1.4 Extension to Nodes of a Graph.

One can extend these ideas to graphs. In the previous results the pixels were treated the same as if they were nodes on a graph. For example the neighbors of a pixels extends directly to neighbors of a node on a graph. In this case the nodes might represent a region. The distance between nodes could be defined based on region properties such as average gray-level. There is an arc between the nodes if they are adjacent. One can then compute the gradient function and proceed to form the watersheds of the graph which will correspond to watersheds over the graph structure. This will then produce a segmentation.

1.5 References

Dougherty, E. R. (1992). *An Introduction to Morphological Image Processing*, SPIE Optical Engineering Press.

Moga, A. N., and Gabbouj, M. (1997). “Parallel Image Component Labeling with Watershed Transformation.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19, no 5, May, 1997.

Najman, L., and Schmitt, M. (1996). “Geodesic Saliency of Watershed Contours and Hierarchical Segmentation.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18, no. 12, December, 1996, 1163.

Roerdind, J. B. t. m., and Meijster, A. (1999). “The Watershed Transform: Definitions, Algorithms and Parallelization Strategies.” *IWI 99-9-06*, Instituet for mathematics and Computing Science
University of Groningen.

Sonka, M., Hlavac, V., and Boyle, R. (1999). *Image Processing, Analysis, and Machine Vision*, Chapman and Hall Computing.

Vincent, L., and Soille, P. (1991). “Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 6, June 1991, pp. 583-598.