AUTOMATED WATER FINDERS FOR RADAR IMAGERY

Dr. Pi-Fuay Chen, Electronics Engineer and
Mr. Tho Cong Tran, Electronics Engineer
U.S. Army Topographic Engineering Center
Fort Belvoir, Virginia 22060-5546
United States of America
ISPRS Technical Commission I

ABSTRACT:

An automated method for finding and extracting water regions from Synthetic Aperture Radar (SAR) imagery is presented. Three automated water finders are implemented for experimentation with two sets of SAR imagery obtained from the areas of Elizabeth City and Asheboro, North Carolina. The input radar image is first segmented into four categories of terrain features. They are water, field, forests, and built-up areas. Water regions from the segmented image are extracted and the remaining image is eliminated. Noise appearing on both water and non-water regions is smoothed and thus completes the automated water finding task. A second water finder with a relatively higher threshold for the region growing process was designed for extracting shallow water regions from some SAR images. Discrimination of non-water features having water like gray values is accomplished by using a third water finder equipped with an additional capability of computing and examining region properties.

KEY WORDS: Image Analysis, Image Interpretation, Image Processing, Machine Vision, SAR

INTRODUCTION

The problem with automatic extraction of water bodies from radar imagery has been the subject of research for some time. In the past, various pattern classification methods have been applied to samples of both Synthetic Aperture Radar (SAR) and aerial photographic imagery. In many cases, when the samples of the water regions were homogeneous and each sample contained only the water category, a highly successful classification accuracy was obtained (Fox and Chen, 1988). For all past experiments mentioned above, an image sample size of 32 by 32 pixels was used. This sample window size was found to be optimum for these studies, even though the size of the sample window can be varied arbitrarily.

One of the major drawbacks of a statistical pattern classification system is its inability to classify samples containing two or more terrain categories. This problem occurred most often when the sample window was moved across the boundary of different image regions or categories. A totally different approach was proposed by Chen and Hevenor in a recent paper (Chen and Hevenor, 1990). In that paper, they tried to overcome this drawback by first segmenting the entire SAR image into four categories of terrain features where the boundaries were well preserved. Extraction or classification of the terrain features was then easily performed afterwards. In this paper, the technique described in (Chen and Hevenor, 1990) was used as a preprocessor to segment a given SAR image into four terrain categories of water, fields, forests, and built-up areas. The water regions are then extracted, and the remaining image is eliminated. The extracted water regions are smoothed and thus completes the desired automated water finding task. A second water finder was designed for finding shallow water regions that requires a relatively higher threshold value on the region growing operation. A third special water finder was implemented for extracting only the largest water body from all water regions found from an image. Discrimination of non-water features having water like gray values was also accomplished by the use of this third water finder.

SYSTEM DESCRIPTION

Two sets of SAR imagery were used for testing three automated water finders described below. The first set of SAR imagery was X-band, and HH polarization which was taken over the Elizabeth City, North Carolina area with a UPD-4 radar system. Approximately 200 SAR images were digitized, and stored for various feature extraction tasks. Each digitized image consists of 512 by 512 pixels and each pixel contains 8 bits. Each image represents a ground area of approximately 1.6 by 1.6 square miles. The second set of SAR imagery was also X-band, but was VV polarization. This set of imagery was taken over the Asheboro, North Carolina area. Each image also consists of 512 by 512 pixels of 8-bit length, and an entire image covers a ground area of approximately 0.8 by 0.8 square miles. Two SAR images that contain water regions were selected from this set of imagery for testing the automated water finders.

Most of the images selected contain area features such as water, fields, forests, built-up areas, and some linear features such as bridges, roads, railroads, and boundaries between area regions. The required algorithms for performing automatic water finding task were all written in the C programming language and implemented on a SUN 4/330 microcomputer system for the purpose of transferring them to a development laboratory.

METHODOLOGY

There are several algorithms that are required to automatically segment and extract water regions from SAR imagery. The process for the automated water finder for SAR images consists of the following steps:

1. Load the desired digitized SAR image from the disk to the computer. The image will be shown in black and white on the display monitor.

2. Based on its gray level, each pixel of the image is assigned a corresponding pseudocolor for easy viewing.

3. A Sobel edge operator is moved sequentially through the entire image for edge enhancement. Appendix A describes the Sobel edge operator.

4. In order to eliminate the noise produced by the edge operation and the noise appearing on the original image, a lowpass filter is passed through the whole image. The lowpass filter is described in Appendix B.

5. After the lowpass filtering, a technique called "region growing" is employed to merge together pixels which have similar gray values. A commonly used simple region growing technique is explained in Appendix C. For our case, the region growing technique was modified to become a two-pass process as described below. First a threshold value T is set. The selection of a threshold value T requires a lengthy process (Nagao and Matsuyama, 1980) which will be described in detail in a separate subsection of this paper.

   Once the threshold value T is selected, the first-pass of the region growing operation is performed as follows. A control pixel for region n (or category n), called P_{cn}, is selected arbitrarily from the image. Usually the unlabeled upper left pixel in the image is assigned as the control pixel P_{cn}. The next step is to sequentially compare the gray value of each pixel with that of P_{cn} as given by (1):

   $$|G(i,j) - G_{cn}| < T \qquad (1)$$

   where G(i,j) is the gray value of the pixel P(i,j) and G_{cn} is the gray value of the control pixel P_{cn}. The subscript n signifies that the control pixel is for the category n.

   The gray values of all pixels that meet the inequality (1) are then summed and the result is added to the gray value of the control pixel. The resulting sum is then saved in a specially designed memory. For a particular category n this summed quantity is expressed as SUM_n, and is given below:

   $$SUM_n = G_{cn} + \sum_{\substack{\text{all pixels met} \\ \text{by inequality (1)}}} G(i,j). \qquad (2)$$

   Otherwise, the pixel under examination is left unlabeled. At the same time a counter, C_n, is incremented by 1 when a pixel is added to the SUM_n. This counter starts with a content of 1 so that the final count of the counter will indicate the total number of pixels added to the SUM_n. This is expressed for the category n as follows:

   $$C_n = 1 + \text{Number of pixels met by the inequality in (1)} \qquad (3)$$

   This process of comparison continues sequentially for all unlabeled pixels in the image. When the process is completed up to this point, the SUM_n is divided by C_n to obtain an average pixel value for a particular region category n as

   $$A_n = \frac{SUM_n}{C_n} \qquad (4)$$

   The next step is to repeat the entire process described above, or computing (1) through (4) for the remainder of the pixels left unlabeled until all pixels on the image are labeled, and each pixel belongs to a particular region. This completes the first-pass of the region growing process. At this point, a number of average pixel values A_n for potential regions will be computed.

   The second-pass of the region growing process is similar to that of the first-pass except that the control pixel values, G_{cn}, are now replaced by the corresponding average pixel values A_n. Also, when the value of a pixel is compared to a particular average pixel value, and if the absolute value of the difference of the pixels is less than the threshold value, the pixel under examination will be merged to that particular average pixel, rather than adding it into the quantity SUM_n. In other words, the value of the pixel under examination will be set equal to that particular average pixel value that it is being compared with. This process is explained in Appendix C. The second pass of the region growing process will continue until each pixel in the image belongs to a particular region category. The entire region growing process will then be complete.

6. The number of region categories created by the application of the region growing process usually exceeds four. A simple pixel grouping routine is then used to further group pixels in the region grown image into exactly four categories. This routine functions as follows: The gray value of each pixel in the region grown image is sequentially examined. If it is larger than or equal to 100, it is set to 150. If it is less than 100, and larger than or equal to 45, it is set to 65. If it is less than 45, and larger than or equal to 8, it is set to 25, otherwise it is set to 4.

7. After the pixel grouping process, the entire image is segmented into exactly four categories, with each category assuming a different pixel gray value. The first three categories having gray values of 4, 25, and 65 represent water, fields, and forests, while the last category having several boundaries or edges with a gray value of 150 belongs to a built-up area.

8. Having done the image segmentation for a given image as described above, the

extraction or finding of water regions from the image is very straight-forward. Set all the pixels having a gray value of 4 to 0, and the remainder of pixels to 255. Now, we have an inverted binary image of water regions.

9. In order to eliminate noise within the water regions which now appears as white spots of gray value 255 on the inverted image, a square box containing a 1 by 1 pixel is made to scan through the entire image from the top left corner to the bottom right corner in a usual way. Any 255 pixel (or 255 pixels) which can be successfully surrounded by this box represents a separated noise pixel (or pixels), and thus its gray value will be replaced to 0. The size of the square box will be sequentially increased from 1 by 1 pixel to 16 by 16 pixels, and for each increase the same scanning and replacing process will be repeated.

10. The inverted image is now inverted again, so all pixels within the water regions will be represented by a gray value of 255. The noise on the non-water regions will appear as white spots of gray value 255. The process discussed in Item 9 will be repeated for eliminating noise from the non-water regions.

11. The water regions which are represented by a gray value 255 will be set to 180. We selected the gray value 180 to represent the water regions because it shows up as a pleasant dark blue color in the pseudo-color domain. This completes the entire water finding processes.

12. A number of selected SAR images contained shallow water bodies. The water finder described failed to extract water regions from these images because the pixels within the water regions appeared too bright (have higher gray values compared to that of those pixels within a regular water region). A second water finder with a relatively higher threshold for the region growing process was designed to compensate this problem.

13. Discrimination of non-water features having water like gray values is accomplished by using a third water finder equipped with an additional capability. Just like the previous two water finders, the water regions and non-water regions having water like gray values are extracted. An algorithm called connected components (Hevenor and Chen, 1990) is applied next. The purpose of this connected components routine is to provide a unique label for each extracted region from the previous operation. The detail of the connected components is described in Appendix D. Various properties of each region such as area, centroid, elongation, perimeter, compactness, etc can be computed for discriminating non-water regions from water regions. For our case, only the property area needs to be computed since the largest water region always has the largest area among all regions labeled for all SAR images tested. The calculation of area for each region can be made by taking a summation of all pixels within the region. Finally, the gray value of each pixel within the region having the largest area is set to 180, and the remainder of regions are eliminated by setting their pixels to 0.

Selection of an Optimum Threshold Value:

The selection of a threshold value for the pixel based region growing plays a crucial role in the entire process. The selection of the threshold value should be adaptively determined by the image data under analysis. Using an improperly predetermined fixed threshold value for region growing would lead to a serious mistake and would end up in either undergrowing or overgrowing of the image. For our experimentation, the original SAR image of 512 by 512 pixels was first edge enhanced, and smoothed with a lowpass filter. These two steps took place in the first step discussed in Appendix E. The smoothed image was divided into 64 blocks of subimages with each block consisting of 64 by 64 pixels. The threshold determination method discussed in (Nagao and Matsuyame, 1980) was applied to each block, and the minimum threshold value found for the subimages was selected as the threshold value for performing region growing for the entire image. The threshold determination method discussed in (Nagao and Matsuyama, 1980) is provided in Appendix E. The optimum threshold value was 12 for both sets of SAR images taken over the areas of Elizabeth City and Ashboro, North Carolina. With this threshold value the majority of images tested were region grown to yield four categories of area terrain features as described.

RESULTS AND DISCUSSION

The software for the three automated water finders was successfully tested using a group of 18 images selected from two sets of SAR imagery. These two sets of SAR imagery were discussed previously in the section System Description. Test results for the water finders are summarized in Table 1. The name of the specific finder (or finders) used in each image is also indicated in Table 1. For illustration purposes, only the results obtained from three images will be presented. It is seen from Table 1 that for most test images a good to fair result was obtained. Only two images are found to be marginally acceptable. The following observations can be made from the results:

Table 1. Test Results for Water Finders Using Two Sets of SAR Images

Image Names	Figure Number	Finders Used	Results
Unf007	NS	Water	Good
Unf014	NS	Water	Good
Unf026	1a	Water	Good
Unf033	2a	Water1	Fair
Unf034	NS	Water1	Fair
Unf051	NS	Water1	Fair
Unf052	NS	Water1	Marginal
Unf053	NS	Water/Water2	Fair
Unf056	NS	Water2	Good

Unf111	3a	Water2	Good
Unf118	NS	Water/Water2	Good
Unf143	NS	Water1	Marginal
Unf163	NS	Water2	Good
Unf165	NS	Water2	Fair
Unf167	NS	Water1	Fair
Unf182	NS	Water1	Fair
Erim13	NS	Water2	Fair
Erim27d	NS	Water2	Good

NS = Not Shown

1. Very small or narrow water regions, such as the narrow stream on the lower left corner of the first original image, Figure 1a, disappeared after the edge operation and region growing processes. This is illustrated in the resultant Figure 1b. The first finder, named Water, was used for this test.

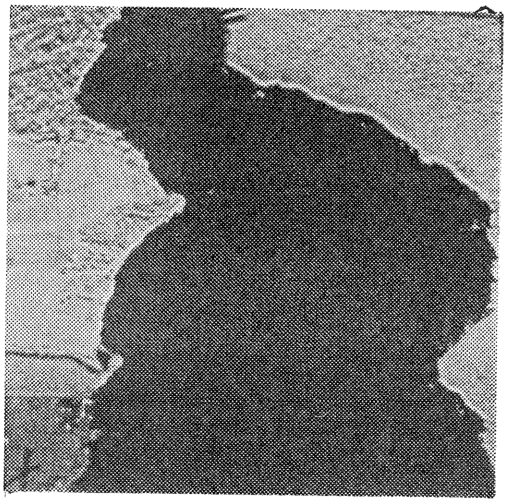

Figure 1a. The Original SAR Image, Unf026.

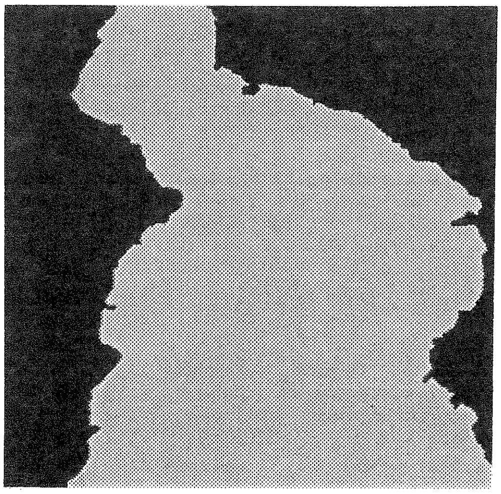

Figure 1b. The Result of Applying the Water Finder to the SAR Image, Unf026.

2. To extract shallow water regions which have relatively high pixel gray values, a second water finder, named Water1, with a higher threshold value of 14 was used. The second original image, Figure 2a, and the corresponding result, Figure 2b, show this point. The narrow and long triangle appearing on the right side of Figure 2a was due to a misalignment of the film while in digitization. The triangle clearly shows an edge of the film. This mistake caused an error in water finding which appears as two near vertical and two horizontal black lines on the right side of Figure 2b. Several small water regions resulting on the left side of Figure 2b are errors due to an imperfect region growing.

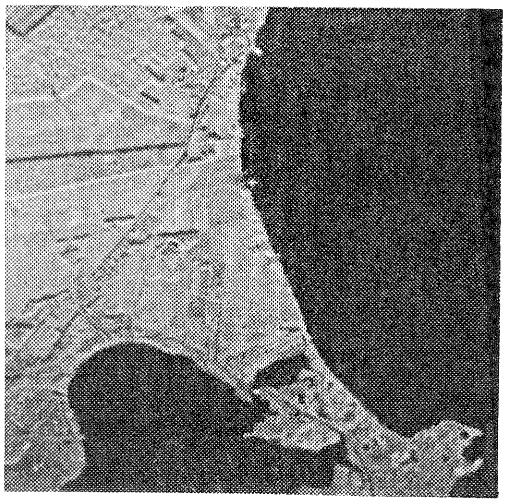

Figure 2a. The Original SAR Image, Unf033.

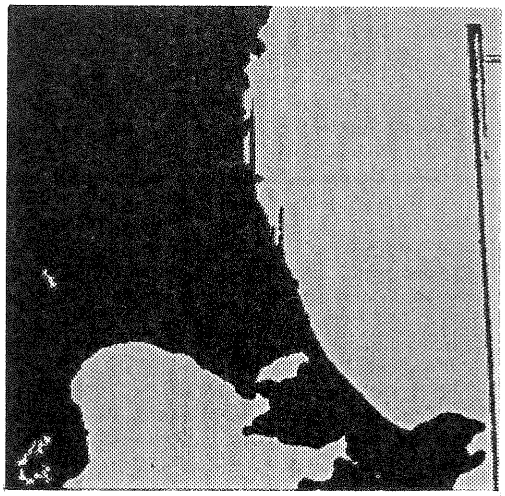

Figure 2b. The Result of Applying the Water1 Finder to the SAR Image, Unf033.

3. Discrimination of non-water dark regions whose pixels have similar gray values to those in water regions appears to be very effective. The airfield runways that appeared on the third original image, Figure 3a, are the example of this discrimination. They are completely removed as shown in the resultant Figure 3b. A third finder, Water2, was used to accomplish this discrimination.

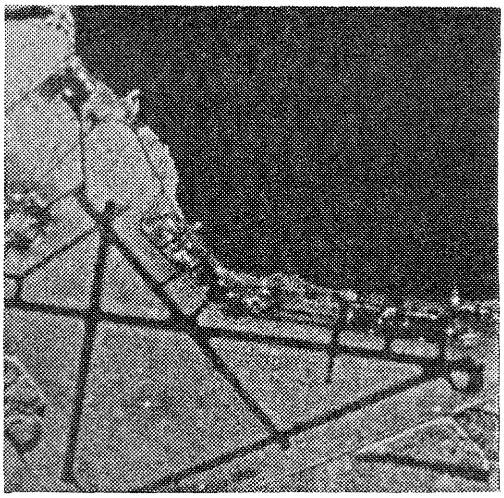

Figure 3a. The Original SAR Image, Unf111.

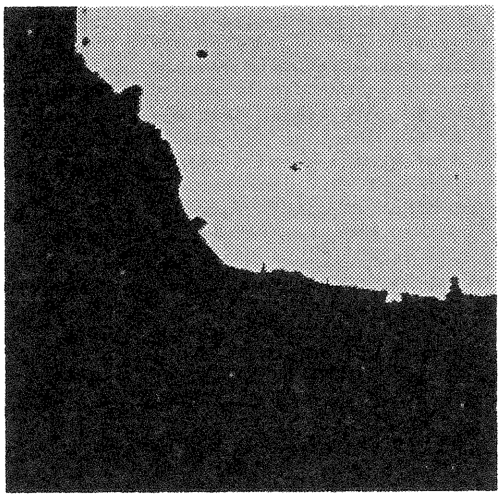

Figure 3b. The Result of Applying the Water2 Finder to the SAR Image, Unf111.

CONCLUSIONS

1. Automated finding of water regions from SAR imagery can be effectively accomplished by properly applying a set of image processing and computer vision algorithms sequentially.
2. Automated water finders can be simply implemented by using the concept of the automated terrain segmenter, which was previously developed, as a preprocessor.
3. Shallow water regions, which appear considerably brighter than ordinary water regions on an SAR image, can be extracted by using a separate water finder with a higher threshold value for the region growing process.
4. Discrimination of non-water features which have approximately the same gray values as that of water regions can be accomplished by adding two more algorithms (connected components and region property computation) on the first water finder as described. The typical non-water terrain features of this nature include airfield runways and shadows.

REFERENCES

+ References from BOOKS:
Nageo M. and Matsuyama T., 1980. A Structural Analysis of Complex Aerial Photographs. Plenum Press, New York, pp. 83-118.

+ References from GREY LITERATURE:
a) Chen, P.F. and Hevenor, R., 1990. Automated Segmentation and Extraction of Area Terrain Features from Radar Imagery. Research Note ETL-0554, U.S. Army Engineer Topographic Laboratories,

b) Fox N. and Chen, P.F., 1988. Improving Classification Accuracy of Radar Image Using a Multiple-Stage Classifier. Research Note ETL-0502, U.S. Army Engineer Topographic Laboratories, Fort Belvoir, VA-USA.

c) Hevenor, R. and Chen, P.F., 1990. Automated Extraction of Airport Runway Patterns from Radar Imagery. Research Note ETL-0567, U.S. Army Engineer Topographic Laboratories, Fort Belvoir, VA-USA.

APPENDIX A SOBEL EDGE OPERATOR

The Sobel operator is a 3 by 3 pixel non-linear edge enhancement mask which is multiplied sequentially with all pixel values in an image to produce a pattern of more pronounced edges.

The weights for the Sobel mask are shown below:

$$\begin{array}{ccc} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \\ \multicolumn{3}{c}{\text{x-direction}} \end{array} \qquad \begin{array}{ccc} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \\ \multicolumn{3}{c}{\text{y-direction}} \end{array}$$

Assume a block of 3 by 3 pixels to be multiplied with the Sobel mask centered at the point (i,j) and having a gray-value distribution as given below:

$$\begin{array}{ccc} A_0 & A_1 & A_2 \\ A_7 & F(i,j) & A_3 \\ A_6 & A_5 & A_4 \end{array}$$

Then, the resultant pixel value G(i,j) which will replace F(i,j), will be

$$|G(i,j)| = \sqrt{X^2 + Y^2},$$

where $X = (A_2 + 2A_3 + A_4) - (A_0 + 2A_7 + A_6)$,

and $Y = (A_0 + 2A_1 + A_2) - (A_6 + 2A_5 + A_4)$.

APPENDIX B LOWPASS FILTERS

The most commonly used lowpass filter consists of a mask of 3 by 3 pixels. It is multipled sequentially with all pixel values on a given image to produce a smoothed pattern.

The weights for the lowpass mask are shown below:

$$\begin{array}{ccc} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{array}$$

Assume a block of 3 by 3 pixels to be multiplied with the lowpass mask centered at the point (i,j) and having a gray-value distribution as given below:

$$\begin{array}{ccc} A_0 & A_1 & A_2 \\ A_7 & F(i,j) & A_3 \\ A_6 & A_5 & A_4 \end{array}$$

Then, the magnitude of the resultant G(i,j) which will replace F(1,j), will be

$$G(i,j) = \frac{1}{9} \left(\sum_{k=0}^{7} A_k + F(i,j) \right)$$

For more sophisticated lowpass filters, the size of the mask can be increased to 5 by 5 or 7 by 7 pixels, and have values consisting of all 1's. However, the processing time of using these filters will be increased accordingly.

APPENDIX C SIMPLE REGION GROWING

The simple region growing method, based on pixel gray value, consists of the following steps:

Step 1: If all pixels in a given image are labeled, then end. Else take an unlabeled pixel and assign a new unused region number.

Step 2: If the absolute difference of gray value between the new labeled pixel and its neighboring pixels are less than the threshold value, merge the neighboring pixels and assign them the same region number.

Step 3: Iterate Step 2 until no pixels adjacent to the new labeled region can be merged.

Step 4: Go to Step 1.

APPENDIX D CONNECTED COMPONENTS

The purpose of the connected components routine is to provide a unique label for each component of 1 pixel in the binary image that represents water regions. Each label is a number that is assigned to every pixel in a given connected component. This labeling operation can be performed by scanning the entire binary image with a 3-pixel by 3-pixel array and considering the following pattern:

$$\begin{array}{ccc} C & B & E \\ D & A & \end{array}$$

If we scan along the image from left to right and from top to bottom, and if pixel A is the pixel presently being considered and it has a value of 1, then a label must be assigned to A. The pixel at D, C, B and E have already been labeled and can be used to label A. If the pixels at D, C, B and E are all 0, then A is given a new label. If pixels C, B and E are all 0 and D = 1, then A is given the label of D. Each possible construction of 0's and 1's for the pixels D, C, B and E must be considered when providing a label for A. If two or more pixels in the set D, C, B and E are equal to 1 and they all have the same label, then A is also given the same label. The real difficulty comes when two or more of the pixels D, C, B and E have different labels. This can occur when two or more separate components, which were originally assigned different labels are found to be connected at or near pixel A. For these cases the pixel A is given the label of any one of the pixels D, C, B or E, which has a value equal to 1. An equivalence list consisting of pairs of equivalent labels is formed. After the binary image has been completely scanned, the equivalance list is restructured in such a way as to contain a number of lists. Each of these lists contains all of the equivalent labels associated with a particular connected component. A new label is then given to each of the new lists and is assigned to each of the appropriate pixels.

APPENDIX E THRESHOLD DETERMINATION METHOD

The following is the adaptive threshold determination algorithm used for our experimentation.

Step 1: Differentiate the smoothed image using the operator

$$d(i,j) = \max_{\substack{-1 \le k \le 1 \\ -1 \le m \le 1}} |G(i,j) - G(i+k,j+m)|,$$

where G(i,J) and d(i,j) denote the gray value and the differential value at a point (i,j).

Step 2: Divide the differentiated image into M blocks of 64 by 64 pixel subimages and make a histogram $h_n(d)$ of the differential values $d(i,j)$ in the n-th block of subimage ($n = 1,2, \ldots, M$).

Step 3: The Valley-Detection Algorithm. For each histogram $h_n(d)$, find the minimum value, d_n which satisfies the following inequalities:

$$h_n(d_n) < h_n(d_n + k) \text{ for } k = 1, 2, \ldots, N,$$

$$d_n > d_n^* ,$$

where d_n^* denotes the differential value for which histogram $h_n(d)$ has the maximum population. N is set initially to 9. The value of N will be reduced from 9 to 8, and from 8 to 7 sequentially until a satisfactory result is obtained.

Seep 4: Find the minimum value among the d_n for all blocks of subimages, and make it the threshold value T for all areas of the image. That is

$$T = \min_{1 \leq n \leq M} d_n .$$