
Masters Theses

Student Theses and Dissertations

2009

Analysis of lesion border segmentation using watershed algorithm

Hanzheng Wang

Follow this and additional works at: https://scholarsmine.mst.edu/masters_theses



Part of the [Electrical and Computer Engineering Commons](#)

Department:

Recommended Citation

Wang, Hanzheng, "Analysis of lesion border segmentation using watershed algorithm" (2009). *Masters Theses*. 7370.

https://scholarsmine.mst.edu/masters_theses/7370

This thesis is brought to you by Scholars' Mine, a service of the Missouri S&T Library and Learning Resources. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact scholarsmine@mst.edu.

ANALYSIS OF LESION BORDER SEGMENTATION USING
WATERSHED ALGORITHM

by

HANZHENG WANG

A THESIS

Presented to the Faculty of the Graduate School of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

2009

Approved by

Randy H. Moss, Advisor

R. Joe Stanley

William V. Stoecker

© 2009

Hanzheng Wang

All Rights Reserved

ABSTRACT

Automatic lesion segmentation is an important part of computer-based skin cancer detection. A watershed algorithm was introduced and tested on benign and melanoma images. The average of three dermatologists' manually drawn borders was compared as the benchmark. Hair removing, black border removing and vignette removing methods were introduced in preprocessing steps. A new lesion ratio estimate was added to the merging method, which was determined by the outer bounding box ratio. In post-processing, small blob removing and border smoothing using a peninsula removing method as well as a second order B-Spline smoothing method were included. A novel threshold was developed for removing large light areas near the lesion boundary. A supervised neural network was applied to cluster results and improve the accuracy, classifying images into three clusters: proper estimate, over-estimate and under-estimate. Comparing to the manually drawn average border, an overall of 11.12% error was achieved. Future work will involve reducing peninsula-shaped noise and looking for other reliable features for the classifier.

Index terms—Malignant Melanoma, Watershed, Image Processing, Segmentation, Neural Network.

ACKNOWLEDGMENTS

This thesis could not have been written without Dr. Moss who not only served as my advisor but also encouraged and challenged me throughout my academic program. He and the other committee members, Dr. Stoecker and Dr. Stanley, guided me through the dissertation process, never accepting less than my best efforts. I also would like to thank Xiaohe Chen for his knowledge and previous work on the watershed algorithm. Jason Hagerty and Thomas Mark Daniel Szalapski gave me great help on programming the algorithm. My friends Wang Renzhong and Wang Chong helped me with data optimization and image design. I thank them all.

Also I would like to thank my parents and Zhang Yi for their continuous support on my study and unselfish care in my life.

Especially, I would like to thank everyone who was concerned and cared about my situation when I encountered the tough-visa-waiting period in China.

TABLE OF CONTENTS

	Page
ABSTRACT.....	iii
ACKNOWLEDGMENTS	iv
LIST OF ILLUSTRATIONS	vii
LIST OF TABLES	viii
SECTION	
1. INTRODUCTION	1
2. WATERSHED ALGORITHM FOR LESION SEGMENTATION.....	5
2.1. PREPROCESSING: HAIR REMOVAL	6
2.2. PREPROCESSING: VIGNETTE REMOVAL	8
2.3. PREPROCESSING: BLACK BORDER REMOVAL	11
2.4. LUMINANCE AND BLUE PLANE IMAGES	13
2.5. WATERSHED ALGORITHM.....	13
3. TRANSFORM FROM MATLAB® TO VISUAL C++ VERSION.....	17
3.1. CODE TRANSFORM	17
3.2. ADAPTIVE AREA SIZE	19
4. POST-PROCESSING: OBJECT MERGING	22
4.1. OBJECT HISTOGRAM	22
4.2. OBJECT MERGING	23
4.3. WATERSHED BORDER.....	25
4.4. OUTER BOUNDING BOX	26
4.5. EDGE OBJECT THRESHOLD	30
4.6. LESION RATIO ESTIMATE (LRE)	33
4.7. LRE CORRECTION BASED ON NEURAL NETWORK	35
4.8. NOISE CONTROL---BORDER SMOOTHING.....	38
4.8.1. Peninsula Removal	38
4.8.2. Border Sampling and B-Spline Border Smoothing	39
5. RESULTS.....	43
6. CONCLUSION AND FUTURE WORK.....	49

APPENDIX..... 51
BIBLIOGRAPHY 55
VITA 58

LIST OF ILLUSTRATIONS

Figure	Page
2.1: Lesion border segmentation using watershed algorithm flowchart	5
2.2: Hair removal images	7
2.3: Vignette removal images	10
2.4: Black border crop program applied on two images	12
2.5: Original image with sample area part shown as a white square	15
2.6: Magnification of area in white	15
2.7: Flooding procedure	16
3.1: Example lesion image	18
3.2: Watershed segmentation of the image in Figure 3.1	19
3.3: Modified watershed segmentation adding “Area”	20
3.4: Modified watershed segmentation and the original image	21
4.1: Object histogram on blue plane	22
4.2: Merging method from the global peak	23
4.3: Object merging 1	24
4.4: Object merging 2	24
4.5: First pass border	25
4.6: Watershed border	26
4.7: Outer bounding box boundaries	28
4.8: Outer bounding box	29
4.9: Improved global minimum searching	30
4.10: Edge threshold effect	32
4.11: Linear relationship between actual lesion ratio and lesion ratio estimate	34
4.12: Sampling interval impact	40
4.13: Smoothing methods	41
5.1: Final watershed border and actual border	43
5.2: Final mask layover	44
5.3: XOR operation image	45
5.4: Results and error values	46

LIST OF TABLES

Table	Page
5.1: XOR truth table.....	45
A.1: Image Table	51

1. INTRODUCTION

Skin cancer is the one of the most common cancers in humans, and malignant melanoma is the most deadly form of skin cancer. The number of melanoma cases is increasing significantly. There were 62,480 cases of melanoma estimated in 2008 in the U.S. and about 8,420 deaths the same year [1]. But on the other hand, it is curable if detected early. Lesion segmentation in the early stage of skin cancer is an important part of computer-based diagnosis of malignant melanoma [2].

Automatic image segmentation is useful in lesion segmentation applications. There are a variety of algorithms to solve this problem, including grayscale threshold, split and merge, Sobel and Robert method, edge detection, stabilized inverse diffusion, active contours, fuzzy c-means clustering, region growing, and color histogram threshold. To summarize those methods, automatic image segmentation applications can be classified into four classes: threshold techniques, boundary detection approaches, region growing methods, and hybrid techniques [3].

Threshold techniques [4] often perform poorly because the way they make decisions is based on the local pixel information, such as pixel value, luminance, etc., and the method ignores the related information from connected spatial areas. Compared to the threshold method, the edge and the boundary detection approach [5][6] uses spatial information of an image and focuses around outline detection, but boundary-based methods are sensitive to images with much noise and often produce irregular edges or additional edges. In that case, the boundaries found may not form a closed region. Sobel and Roberts are two of the most popular examples of boundary detection techniques. With region-growing methods, there is no problem in forming a closed region, because

the region-growing method uses the similarity of regional image data to form a region. The seed region growing [7], region splitting, and region merging approaches [8]-[10] are some of the broadly used applications. The key problem for region growing is how to select the initial starting point to obtain higher accuracy of segmentation and how to choose at what point the region growing process should be completed.

With the combination of boundary detection and region growing methods together, the hybrid techniques use a mixed development to provide another segmentation of images, which gives lower border detection errors. Such methods include histogram thresholding and fusion followed by border detection [11]-[13]. Meanwhile, varieties of histogram thresholding methods are applied to different wavelengths to make use of the threshold reflectance-graph-based image followed by polynomial fitting and region growing. Ganster, et al. uses six domain-knowledge-based rules to perform fusion of three different segmentation techniques [14].

On the other hand, sharp gradients are one of the crucial points for those methods to find lesion edges. But the way dermatologists define lesions is not the same as those methods. A dermatologist uses physiology to define the lesion parts which are different from normal skin. Thus, dermatologists tend to find wider boundaries than the automated methods. Furthermore, some of the applications do not take topological information as part of the segmentation process, bringing about holes or gaps in the segmentation outcomes [15].

Based upon the topology of the image, the watershed algorithm is an image segmentation algorithm that divides an image into a large number of different regions. In the image, locations with higher pixel values correspond to “hills”, and the lower ones

represent the “valleys”. The flooding variant of the watershed algorithm corresponds to immersing a lake from the lowest point (bottom of the “valley”), so that the raindrop path will be determined. Through looking for the maximum gradient (steepest path) of the gray value of the neighbors, during the successive flooding of the luminance value relief; watersheds with adjacent catchment basins are built and the basins should emerge along the edges. After that, the image is segmented by creating closed contours on those different area blobs. Normally, this will result in an over-segmentation of the image, especially for a noisy original image. Either the original image must be pre-processed or the regions must be merged on the basis of a similarity criterion afterwards [15].

Xiaohe Chen [15] used the topographical information and post-processing to find the most dermatologist-like segmentation. In his research, a watershed-based approach was investigated for automatically segmenting skin lesions in dermoscopy images utilizing both luminance and color information. Watershed lines are the separate lines of “domains of attraction” of water drops. The flooding variant of the watershed algorithm is equivalent to concentration of the relief in a lake flooded from holes at the minima. The flooding variation is much more efficient than the original falling rain drop approach for many other applications [15]. After application of merging techniques, the watershed algorithm is now a helpful mathematical morphology-based tool for image segmentation [16]. Chen [15] compared his proposed watershed-based segmentation technique to dermatologists’ manual borders on a set of 100 dermoscopy images. A Random Forest algorithm was applied to correct the segmentation accuracy. He compared his watershed border to three dermatologists’ borders and a mean error of 5.30% was achieved.

In this research we use a similar watershed algorithm, which is based upon a topographical representation of an image gray-level map. Based on Dr. Stanley's watershed algorithm, the simple steps of this application are: growing regional minima; labeling regional minima; creating watershed; removing watershed lines; and creating the watershed image from the labeled image.

In this work, the same 100-image set was used as was used by Chen [15]. Three dermatologists' borders are compared as the reference. Also, we use a new Outer Bounding Box method to estimate the rough lesion ratio, which mainly focuses on the lesion area instead of the whole image. An improved Lesion Ratio Estimate formula is proposed to set up the relationship between the Outer Bounding Box ratio and the estimate of the lesion ratio. In addition, in the merging application, a novel Edge Watershed Object Threshold is applied to remove the "big bright" areas in the skin at the lesion boundary, which is a useful way to prevent an over-estimate. A supervised neural network process is added to improve the accuracy of lesion ratio estimate. All the work is completed in Visual C++ except the neural network application and error computing.

The remainder of this thesis is organized into the following sections: 1) description of the watershed algorithm for lesion segmentation, 2) conversion from MATLAB® to a Visual C++ version, 3) post-processing and object merging, 4) results and 5) conclusion and future work.

2. WATERSHED ALGORITHM FOR LESION SEGMENTATION

Figure 2.1 illustrates the flowchart of the watershed algorithm. Each individual component will be discussed in turn.

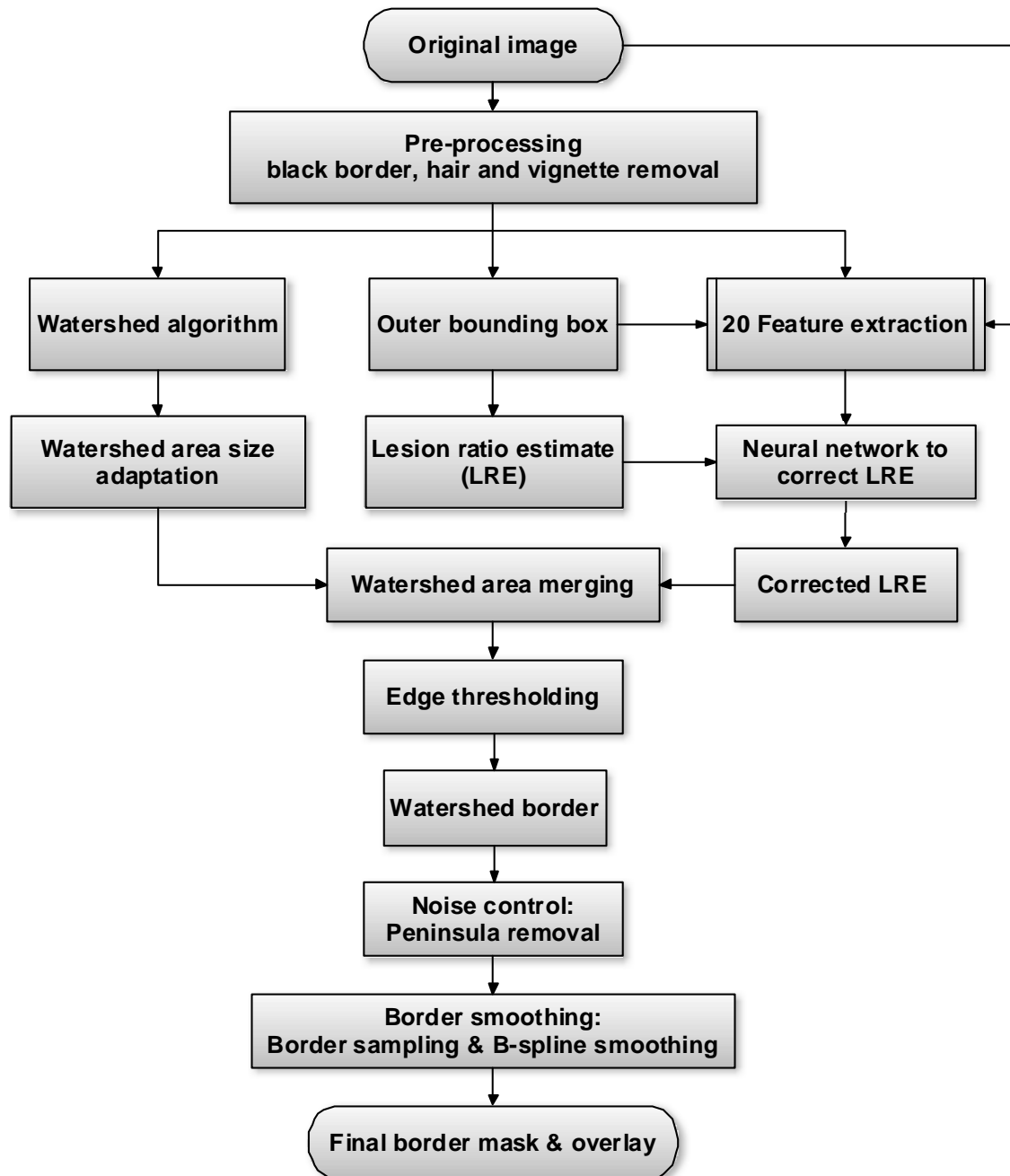


Figure 2.1 Lesion border segmentation using watershed algorithm flowchart

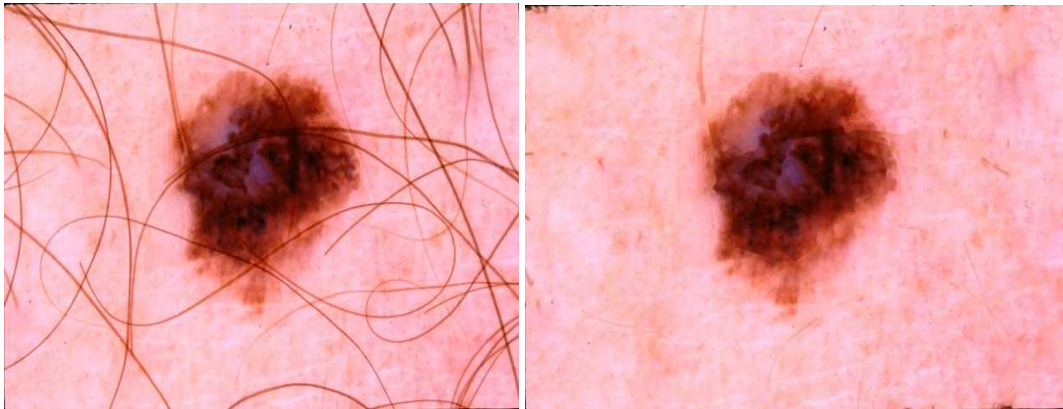
2.1 PREPROCESSING: HAIR REMOVAL

Images with hair or hair stubble within the lesion area will affect the watershed segmentation, because most of the hairs contain similar luminance value as the lesion area. In our image set, some images include hair around or across the lesion area. In this step, the goal is not to suggest a way to reinstate hidden information due to the presence of hair but to replace the concerned hair images pixels in a specific way so that the hair does not influence the processing afterwards. It is clear that there is no effective way to “repair” the image, but it is possible to detect these components and to reduce the (negative) visual effect they cause, which consequently can adversely affect the medical diagnosis.

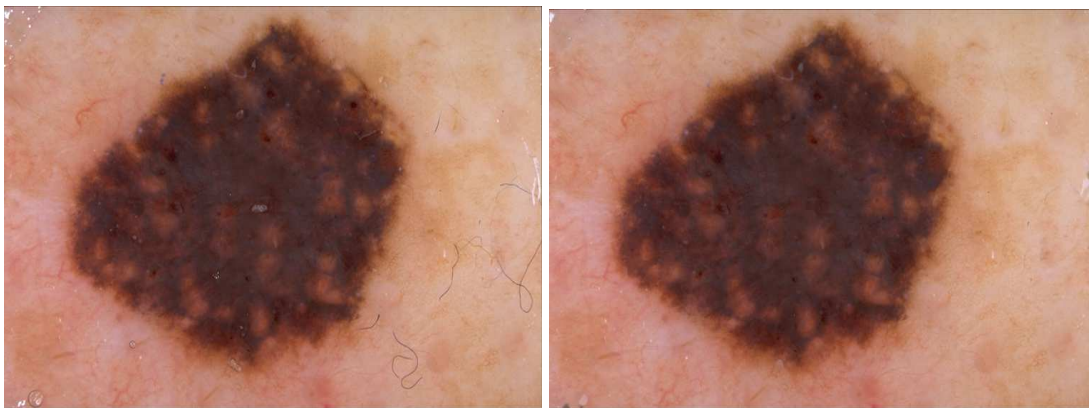
One way to solve the hair problem is using linear or elongated feature detection, which is a very important issue in the areas of image analysis, computer vision, and pattern recognition [17]. There are a number of techniques in linear feature detection, including application of directional morphological filters, the tracking or stick growing method, pixel classification using a neural network with supervised training, S-Gabor filter and deformable splines, and mathematical morphology [18].

In this research, a hair removal method with the morphological closing operator is introduced to remove hairs if they exist. This is based on Thomas Mark Daniel Szalapski’s hair removal program. First, areas are marked where the local standard deviation is high and local average intensity is low. Second, the pixels in these areas are replaced with the local average based on non-marked pixels. In the skin area, it may be possible that parts of pigmented structures have a similar morphology and as high contrast as the hair parts, but even if parts of such structures are considered as hair, the

introduced modification is usually indiscernible and with no effect on the following process. In the lesion area, the hair with sharp contrast will be removed as well as some other features inside the lesion area, but even if those features are falsely eliminated, it will not affect the following steps for border detection. Figure 2.2 shows example images before and after hair removal is applied.



(a)



(b)

Figure 2.2 Hair removal images. (a) and (b) show the hair removal program takes hair away, (c) shows bubbles are removed or weakened.



(c)

Figure 2.2 Hair removal images (cont.) (a) and (b) show the hair removal program cleans the hair parts, (c) shows bubbles are removed or weakened.

2.2 PREPROCESSING: VIGNETTE REMOVAL

The vignetting effect refers to a position dependent loss of brightness in the output of an optical system, due mainly to the blocking of a part of the ray bundle by a limit in the effective range of the aperture stop; therefore gradual fading out of an image at points near its periphery results [19]. It is an artifact that is prevalent in photography. Vignetting distortion becomes small as the f-number gets larger, i.e., with small aperture. For the limiting case of a pinhole camera, it becomes zero [20]. Other reasons for vignetting are geometric in nature. For instance, light arriving at oblique angles to the optical axis may be partially obstructed by the field stop or lens rim. Although not distasteful to the average viewer at low levels, it can significantly damage computer vision algorithms that rely on accurate intensity data to analyze a scene [21].

In our 100-image set, a number of images are influenced by the vignetting effect. Most of the vignetting problems occur in the 30-lesion image set. Because for the image

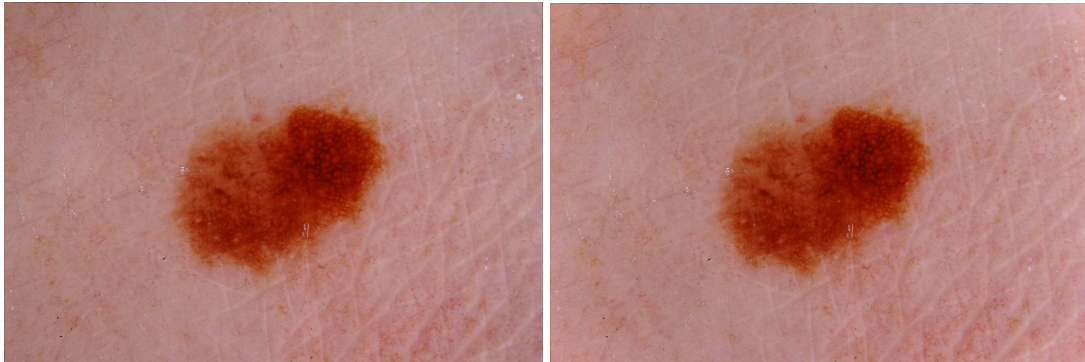
to be used for diagnosis, the patient and the doctor want the photographic image to be clear enough to show the details of the lesion area. In order to get a legible vision of lesion area for small lesions, most of the cameras tend to zoom in and change the focal length. In that case, the vignetting problem becomes serious in the lesion image.

Although vignetting does not significantly affect the lesion area since vignetting always happens in the image periphery, and the lesion area is typically near the center of the image and thus some distance from the image border, vignetting still causes problem in the watershed algorithm and merging method, because after an image projection histogram is created and the global minimum is found, the vignetting area can turn out to be the minimum due to the accumulation of low pixel values horizontally or vertically.

A variety of techniques have been proposed to determine the vignetting effect in an image. Some of the methods require precise scenes for calibration, which typically must be consistently lit [19]. Others methods apply image sequences with overlapping views [20] [21] or image sequences captured with a projector at different exposures and different aperture settings [19].

In our research, three steps are used to solve this problem: first, define a certain set of concentric circular regions, the width of the radius based on a tunable parameter; second, start the approach with the center of image; third, adjust brightness of the next circular region so that the average intensity is the same as the center and continue this for every region. This is based on Thomas Mark Daniel Szalapski's vignetting removal program.

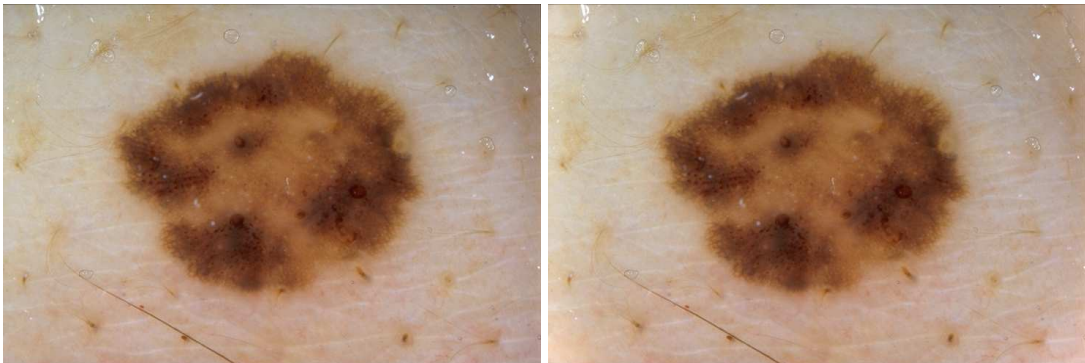
Figure 2.3 illustrates images before and after the anti-vignette method is applied. Most improvement is made at image corners.



(a)



(b)



(c)

Figure 2.3 Vignette removal images

The image luminance after vignetting removal may be different in each corner, because we use a same weight for an entire concentric ring; the darkest part is improved while lighter parts will be even lighter.

2.3 PREPROCESSING: BLACK BORDER REMOVAL

Similar to the vignetting effect, black borders at image boundaries also cause a problem in the image projection histogram. The black frame problem is typically produced by slide digitization. When the image projection histogram is applied, the black border part will show a sharp drop in the projection curve. In that case, this will give a false result for the global minimum and cause outer bounding box generation error [22].

To solve this problem, the simple way is to crop the black border from the most interior point of each black rim. If the border is parallel to the image edge, it is cut completely without bothering the useful image area. On the other hand, if the border is slanted, first the dark point which is the nearest to the image center is found and a crop line is marked parallel to the image edge. The cropping operation is applied to the new line. In that case, some of the image area is cut off as well as the black border area. Also, the preprocessing does not alter the images that do not contain a black frame. This is based on Jason Hagerty's black border removal program.

This may result in a decrease in image resolution and an increase in actual lesion ratio temporarily, but since this preprocessing step is only used to get the projection histogram and the outer bounding box to take in the lesion area, and the border is drawn on the original image, the image resolution decline and lesion ratio increase will not affect the further steps. Also, a compensation step of resizing the image to the original size by padding is applied before error calculation.

Figure 2.4 shows the original image, black border image mask and black border removed image.

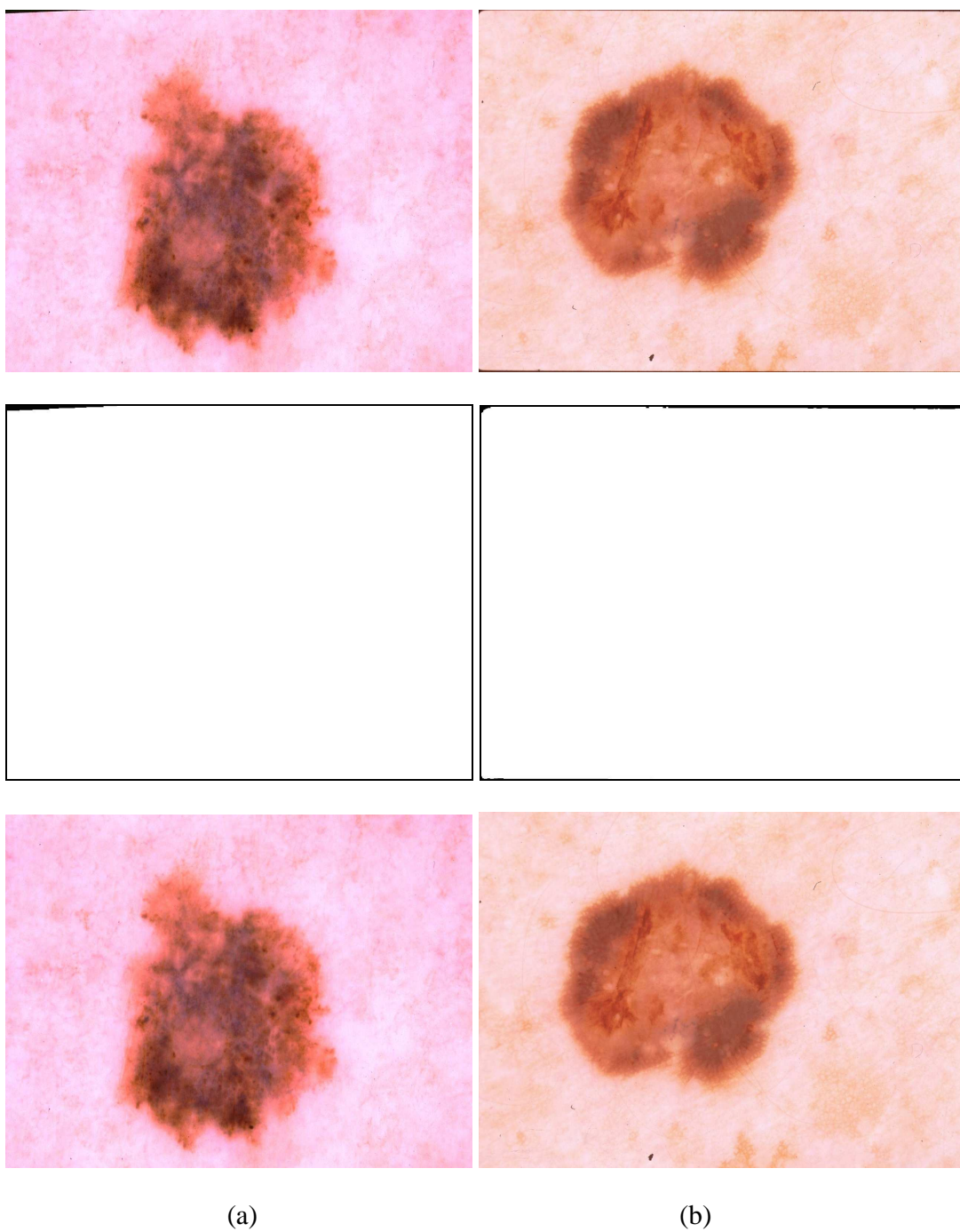


Figure 2.4 Black border crop program applied on two images, (a) and (b). The top row shows the original images with black border, the middle row shows the black borders to be removed, the bottom row shows the resulting images.

2.4 LUMINANCE AND BLUE PLANE IMAGES

In our program, at first the luminance image is mainly used as a testing source. In the luminance image, the red, green and blue planes are combined into a single plane using this formula:

$$L = 0.30R + 0.59G + 0.11B$$

where L is the luminance plane, R is the red value, G is the green value, and B is the blue value [15].

Noticing that on images with a red lesion rim, larger errors occur using the luminance plane than using the blue plane, and then the blue plane image was chosen for use throughout the 100-image set. Usually the blue plane image performs better when a red color occurs at lesion boundary, such as blood vessel, infection, etc.

2.5 WATERSHED ALGORITHM

The watershed algorithm, which is based on the topology of the image, is an image processing segmentation algorithm that splits the image into different shapes of small areas [15], using the maximum gradient as the flooding path. After the continuous flooding of the grey value relief, watersheds with adjacent catchment basins are constructed; the basins should emerge along with the edges [15]. An over-segmentation of the image might occur after this application, especially for noisy images. So we applied the preprocessing steps, such as hair removal, black border removal and vignetting removal to minimize the image noise.

A hierarchic watershed transformation converts the result into a graph display (i.e. the neighbor relationships of the segmented regions are determined) and applies further

watershed transformations recursively. A problem is that the watersheds will increase in width. The marker based watershed transformation performs flooding starting from specific marker positions which have been either explicitly defined by the user or determined with morphological operators. Interactive watershed transformations allow the user to determine include and exclude points to construct artificial watersheds. This can enhance the result of segmentation [23].

On a watershed line there is an equal chance for the water in rainfall simulations to fall into more than one regional minimum when a rainfall simulation is performed. A set of adjacent pixels with the same gray level is called a plateau [15].

The straightforward steps of the watershed algorithm are as follows:

1. Use a rainfall simulation to reach the regional minimum by following the raindrop paths. The rainfall simulation starts from a point with its gray value given as well as its neighboring points. The next rainfall path is given by finding the maximum gradient (steepest path) of the gray value of the neighbors. When the rainfall path reaches a regional minimum, these regional minima become the flooding start points. We use the same label for the pixels in the rainfall path and the regional minima.

2. Repeat the rainfall simulation on all neighboring pixels to see if there is another point that could reach the same regional minimum. Label all the pixels with the same tag when the regional minimum is reached; otherwise, a new flooding process will start with a new regional minimum proposed. Stop the procedure when all the regional minima are processed. The flooding procedure is launched at a regional minimum and proceeds as the water level increases. The flooded area pixels are given the same label.

3. When the different labels of lakes are about to merge, a “dam” is built to prevent the merging. The dam boundaries are finally built when the flooding procedure

reaches the global maximum. The dam boundaries here are referred to as watershed lines. Figures 2.5, 2.6 and 2.7 show a 3-dimensional illustration of the flow stages of the flooding process.

An example image with a small sample square region (shown in white) is from the original image, illustrated in Figure 2.5. The reason using such a small region is because the whole image has thousands of “peaks” and “valleys” and is too complicated for showing a flooding procedure.

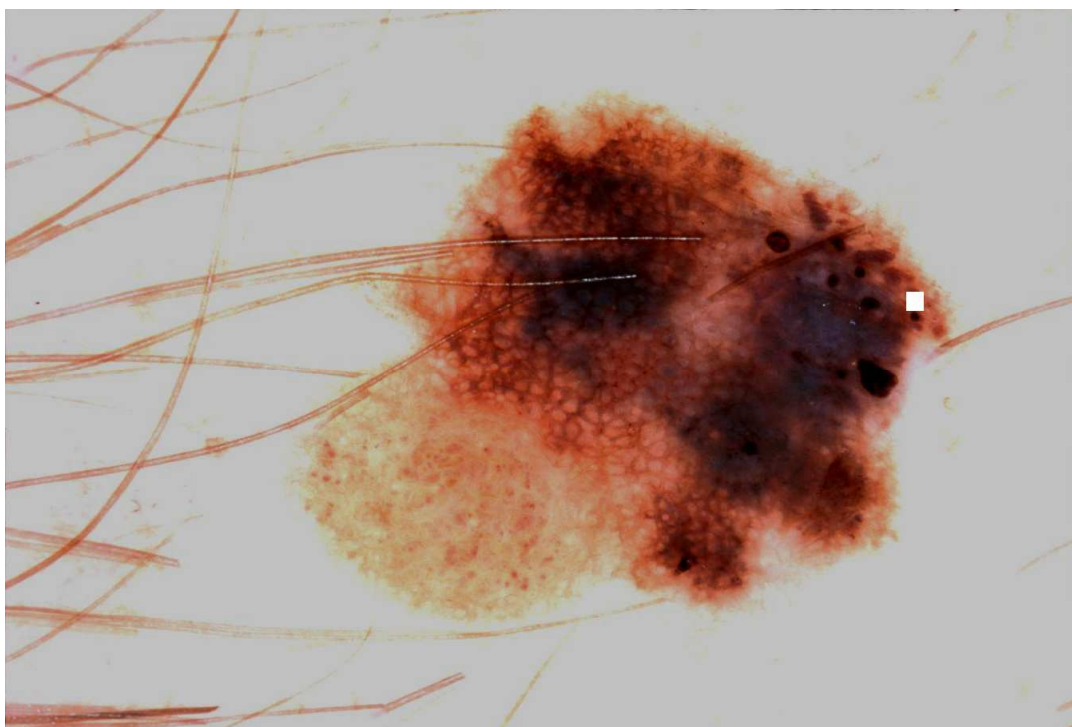


Figure 2.5 Original image with sample area part shown as a white square

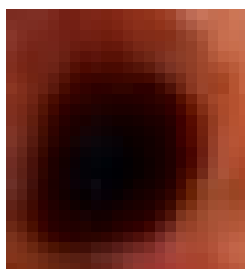
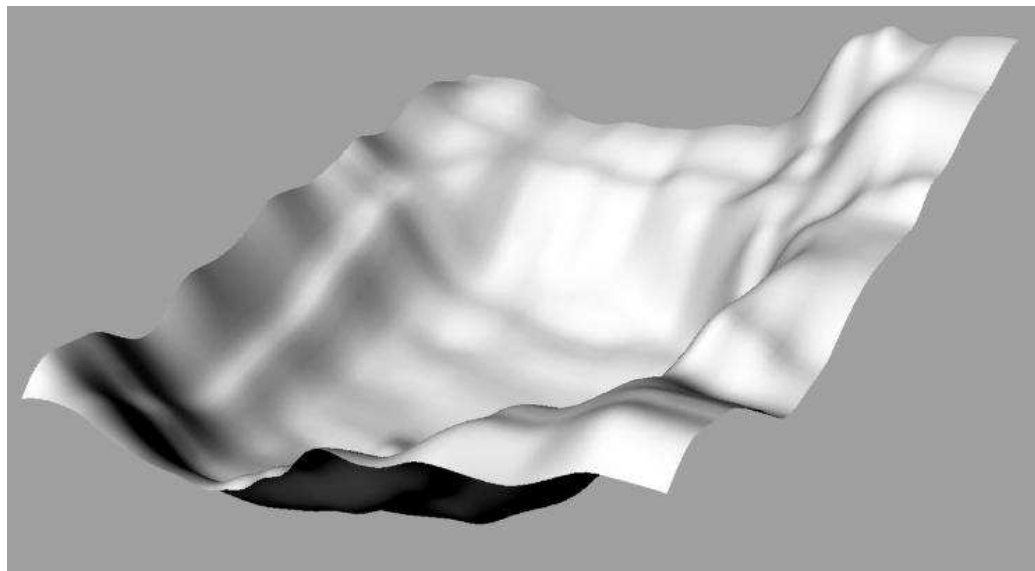
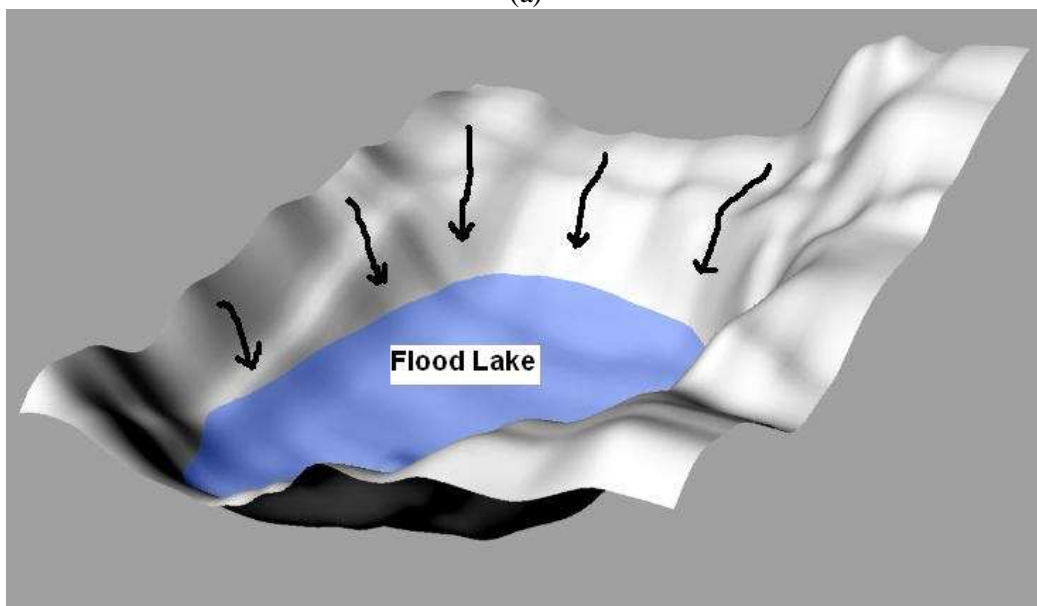


Figure 2.6 Magnification of area in white in Figure 2.5

Figure 2.7 shows the procedure before and after the flooding process. In the mesh figures, high altitude stands for brighter areas with higher pixel values in the image; low altitude shows the darker areas with lower pixel values in the image.



(a)



(b)

Figure 2.7 Flooding procedure

If the flooding procedure continues, the “lakes” will be separated by the “dams” built before different water basins are about to merge.

3. TRANSFORM FROM MATLAB® TO VISUAL C++ VERSION

3.1 CODE TRANSFORM

In Xiaohe Chen's program [15], MATLAB® was used as the tool to carry out the watershed algorithm. MATLAB is a numerical computing environment and programming language which allows easy matrix manipulation, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs in other languages. The two basic packages are Matlab and Simulink, adding toolboxes as supplements for different functions [24].

The watershed algorithm in the MATLAB® version uses built-in functions in a MATLAB® Toolbox. In the Visual C++ version, the functions are realized from basic commands, and a few functions from the OpenCV library. OpenCV is a computer vision library originally developed by Intel. It is free for commercial and research use under a BSD license. The library is cross-platform, and runs on Windows, Mac OS X, Linux, PSP, VCRT (Real-Time OS on Smart camera) and other embedded devices. It mainly focuses on real-time image processing, as such, if it finds Intel's Integrated Performance Primitives on the system, it will use these commercial optimized routines to accelerate processing. Released under the terms of the BSD license, OpenCV is open source software [25]. The Visual C++ transformed watershed algorithm is based on Dr. Stanley's watershed code in C.

Figure 3.1 shows a lesion image that will be used as an example as the watershed algorithm is discussed.

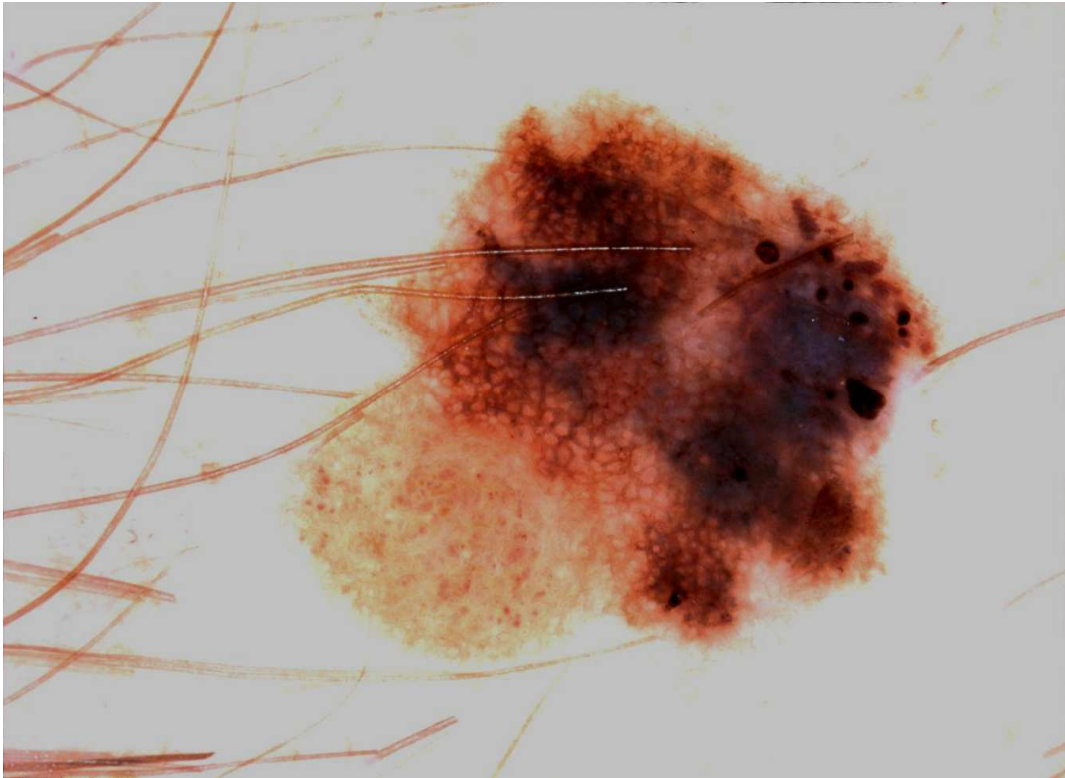


Figure 3.1 Example lesion image

After running the first step of the watershed algorithm for creating watershed blobs, the original image is segmented into thousands of little blobs. The size of the blobs varies due to average blob pixel value. So if the pixels values do not change much, then a large blob is generated, otherwise, small blobs are formed. Also, the shape of the blob can be influenced by a certain constant value, such as long thick hair, which was not removed completely in a preprocessing step. Figure 3.2 shows the watershed segmentation of the image in Figure 3.1.

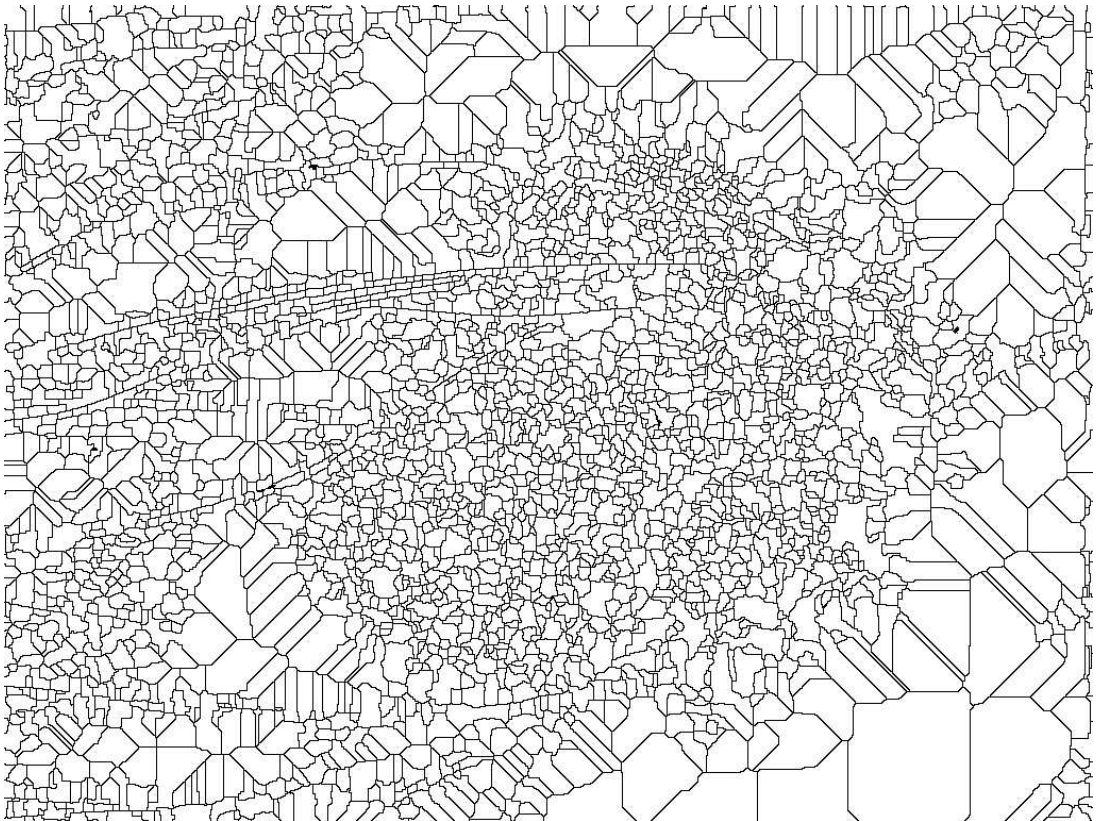


Figure 3.2 Watershed segmentation of the image in Figure 3.1

3.2 ADAPTIVE AREA SIZE

For different images, the blobs' sizes may affect the total lesion shape. So area size control is necessary in watershed segmentation.

In the program, a parameter "area", which stands for the minimum area size of the objects, is introduced to the watershed algorithm before the lesion ratio estimate. Through changing the area size, the result of the watershed segmentation is being controlled and improved in the similarity to the original image, which means by controlling the size of the area, the watershed blobs can be either larger or smaller to adjust to a specific image. If the 'area' is changed to a higher value, in the algorithm, the watershed blob border will be eliminated and neighboring blobs will merge if the total area is greater than the

minimum area. Otherwise, the watershed blob will break into small pieces and new blob borders will be created. Another image with a smaller lesion is used to show this method, because an image with a small lesion can significantly show the effect on the lesion shape from the watershed area size change. Results are shown in Figure 3.3 and Figure 3.4.

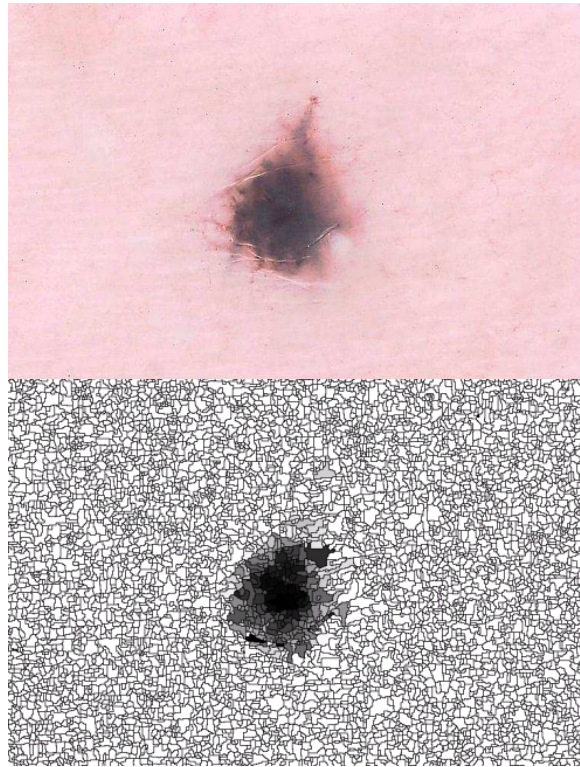


Figure 3.3 Modified watershed segmentation adding “Area”.

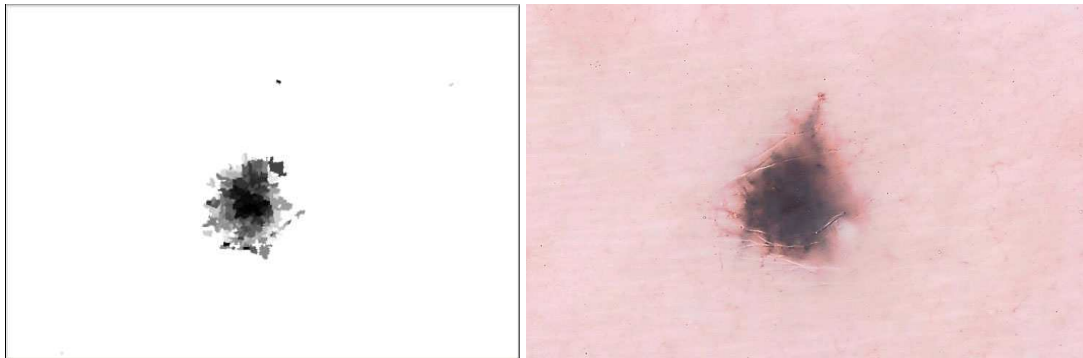
In most of the cases, an area size of “5” gives the best border result. But it may adversely influence the program running speed if the image resolution is higher than 1024×768 . So an adaptive area size is introduced for images with high resolution:

$$F = I \times 5 / (1024 \times 768)$$

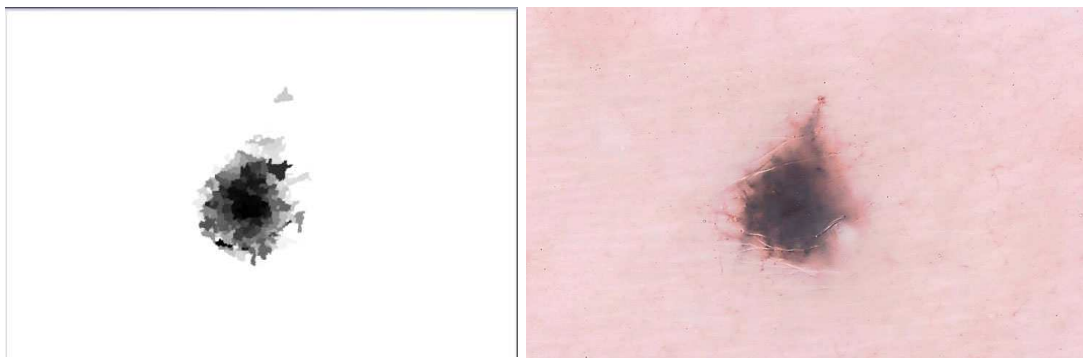
F ----Final area size

I ----Total image size

Figure 3.4 shows a comparison of the modified watershed segmentations. The minimum area size is 5 for Figure 3.4 (a) and the adaptive area size is used for Figure 3.4 (b). The original image is on the right hand side for comparison.



(a)



(b)

Figure 3.4 Modified watershed segmentation and the original image. (a) Object minimum area size of 5; (b) Adaptive minimum area size

This application is mostly effective on small lesion images with high contrast between lesion area and skin area. In that case, the change of area size is obvious and results in different shapes of the lesion segmentations.

4. POST-PROCESSING: OBJECT MERGING

4.1 OBJECT HISTOGRAM

An object histogram is used to show the numbers of objects in different pixel values from the watershed segmentation image. In the histogram, the y-axis represents the number of objects and the x-axis shows the gray-level value of each object. Usually, there are two peaks in the histogram: the higher one represents the skin and the lower one represents the lesion.

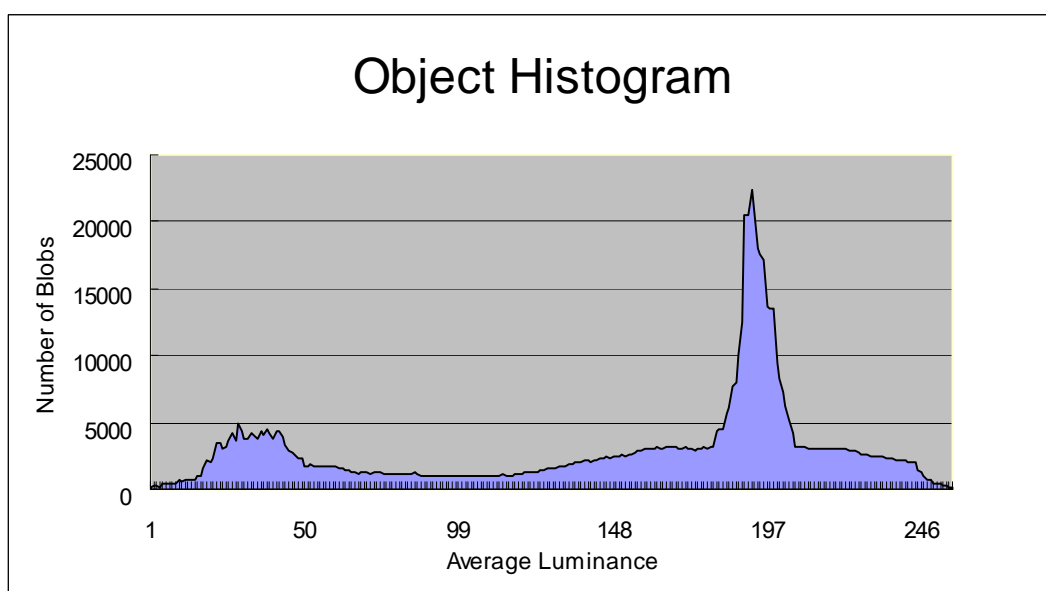


Figure 4.1 Object histogram on blue plane

The histogram in Figure 4.1 is the information of the watershed objects from the sample image shown in Figure 3.2. In Figure 4.1, two peaks can be noticed. The global peak is made of skin blob numbers, which are mostly with high pixel value; the second peak is made of lesion blob numbers, which are mostly with low pixel value. The objects' numbers are used in the merging application to form the lesion area.

4.2 OBJECT MERGING

The merging method is based on looking for the global peak for the skin. (See Figure 4.2) After locating the global maximum, the merging application is started at the peak point of the maximum. The merging goes in two directions. It stops when the covered area represents the estimated skin area.

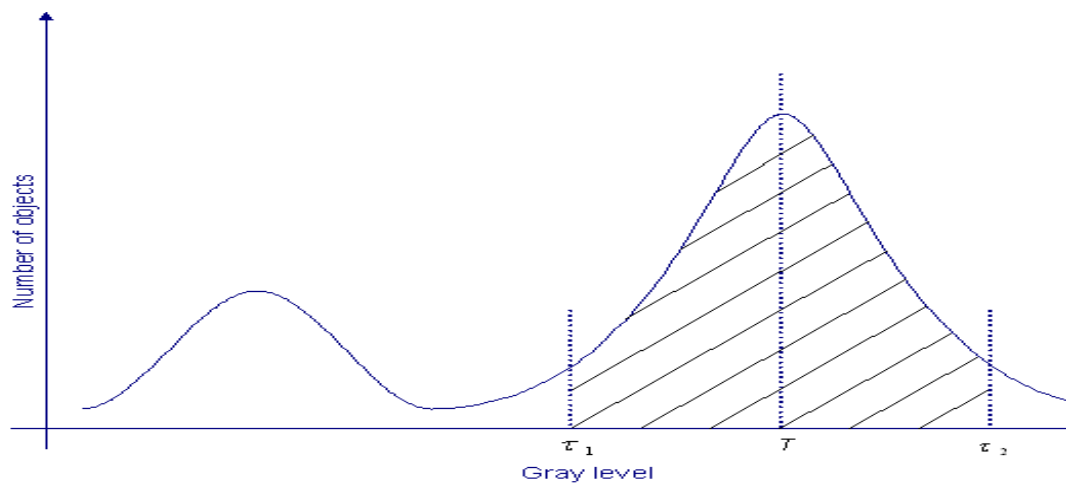


Figure 4.2 Merging method from the global peak [15]

The image area, which is not reached before the merging application stopped, will be ignored and marked with the same label. Figure 4.3 shows the resulting image after the merging application; blob areas equal to the estimated lesion area are merged together.

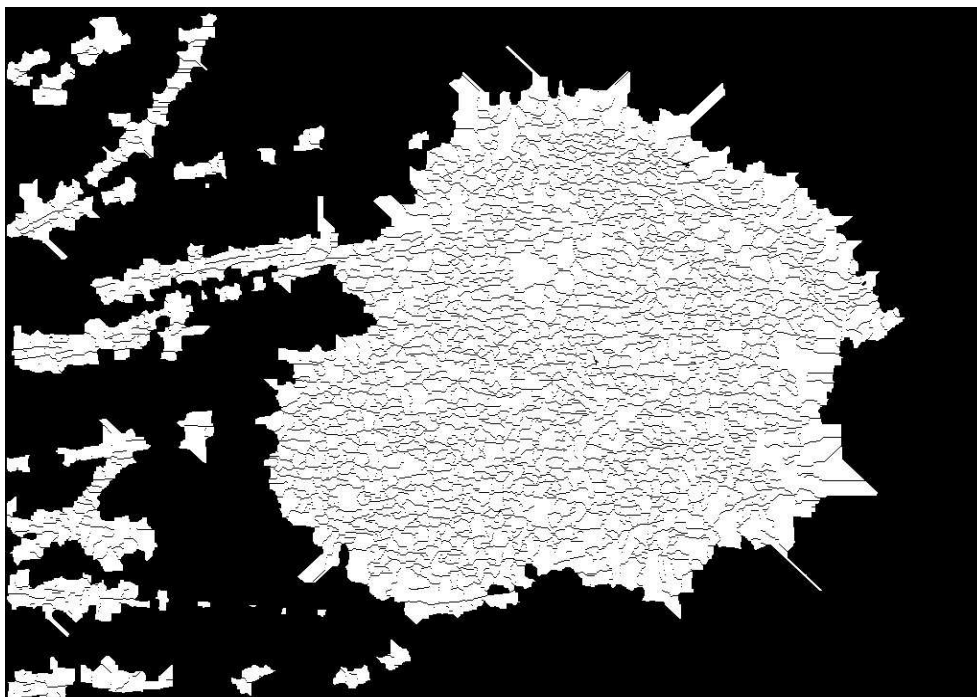


Figure 4.3 Object merging 1

After removing all the watershed lines, the lesion area is merged. In Figure 4.4, the sum of the white areas equals the lesion ratio estimate.

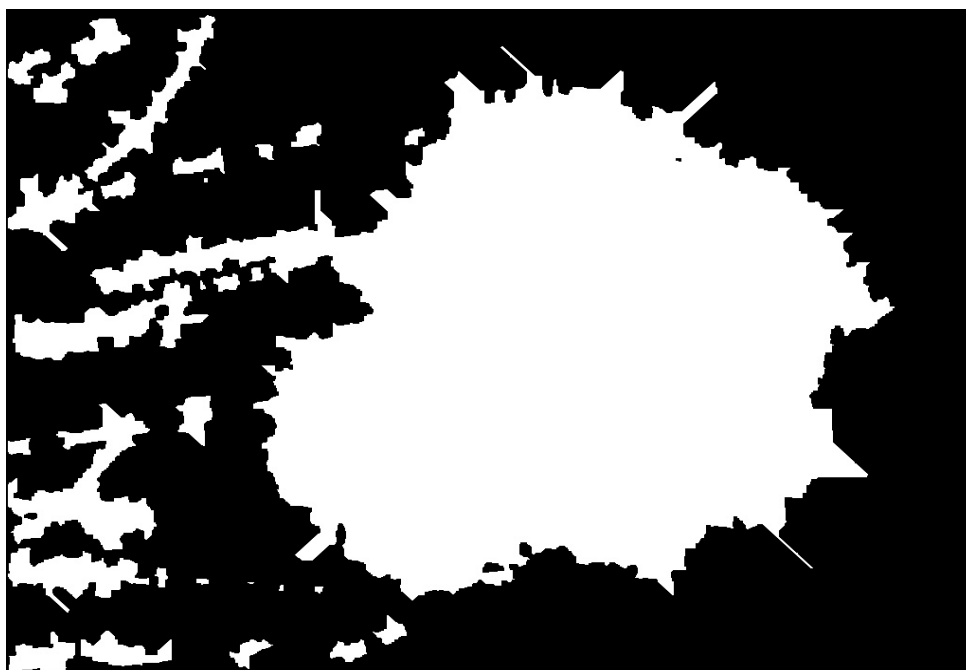


Figure 4.4 Object merging 2

The largest blob is retained and its border found. This border is considered as the first pass border of watershed before post-processing as shown in Figure 4.5.

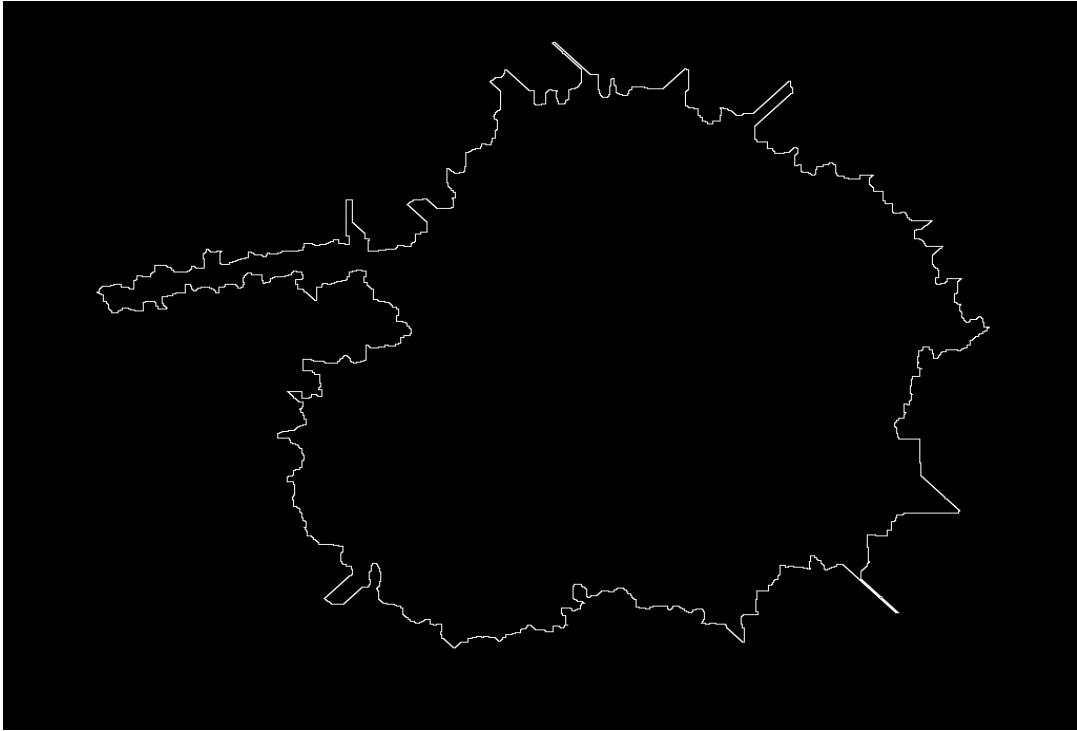


Figure 4.5 First pass border

4.3. WATERSHED BORDER

Figure 4.6 shows the lesion border after the watershed application but before post-processing. The primary watershed border is shown by a thin red line around the lesion area.

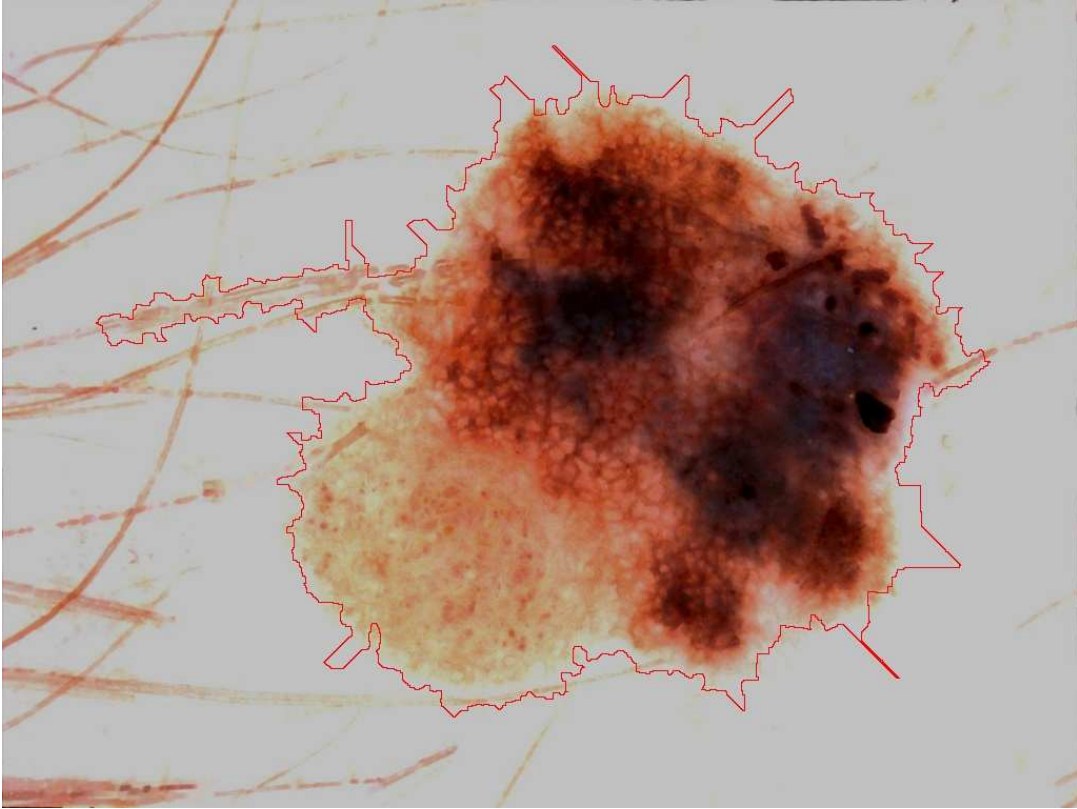


Figure 4.6 Watershed border

4.4. OUTER BOUNDING BOX

The lesion ratio estimate is based on the outer bounding box ratio, which is given by the ratio of the area of the outer bounding box to the whole image area. This is based on Xiaohe Chen's outer bounding box application in MATLAB® [15]. In order to get the boundaries for outer bounding box, first we need the image projection function: Let I_{ij} be the blue plane value of the pixel at the position (i, j) , n be the number of rows and m be the number of columns. The projection of the j^{th} column is expressed as:

$$p_j = \sum_{i=1}^n I_{ij}$$

Let p_f be the best quadratic fit line created from the projection curve:

$$p_f = a_2x^2 + a_1x + a_0$$

The coefficient vector $A = [a_2 \ a_1 \ a_0]$ is determined by minimizing the mean square error between p_f and p_j . To get the final subtraction equation, the means of the curves $\overline{p_f}$ and $\overline{p_j}$ are used to normalize the curve, giving the final equation of the bounding curve B_j :

$$B_j = [p_j - (p_f - \overline{p_f})] \cdot \frac{\overline{p_j}}{p_j}$$

After the final subtraction curve is formed, the curve is divided into two parts based on the global minimum, and the local maximum is found in each direction. These two maxima stand for the two boundaries of outer bounding box vertically. The horizontal outer bounding box boundaries are found in the same way based on the horizontal projection and its best fit quadratic curve [15]. Figure 4.7 shows the procedure to find vertical bounding box boundaries.

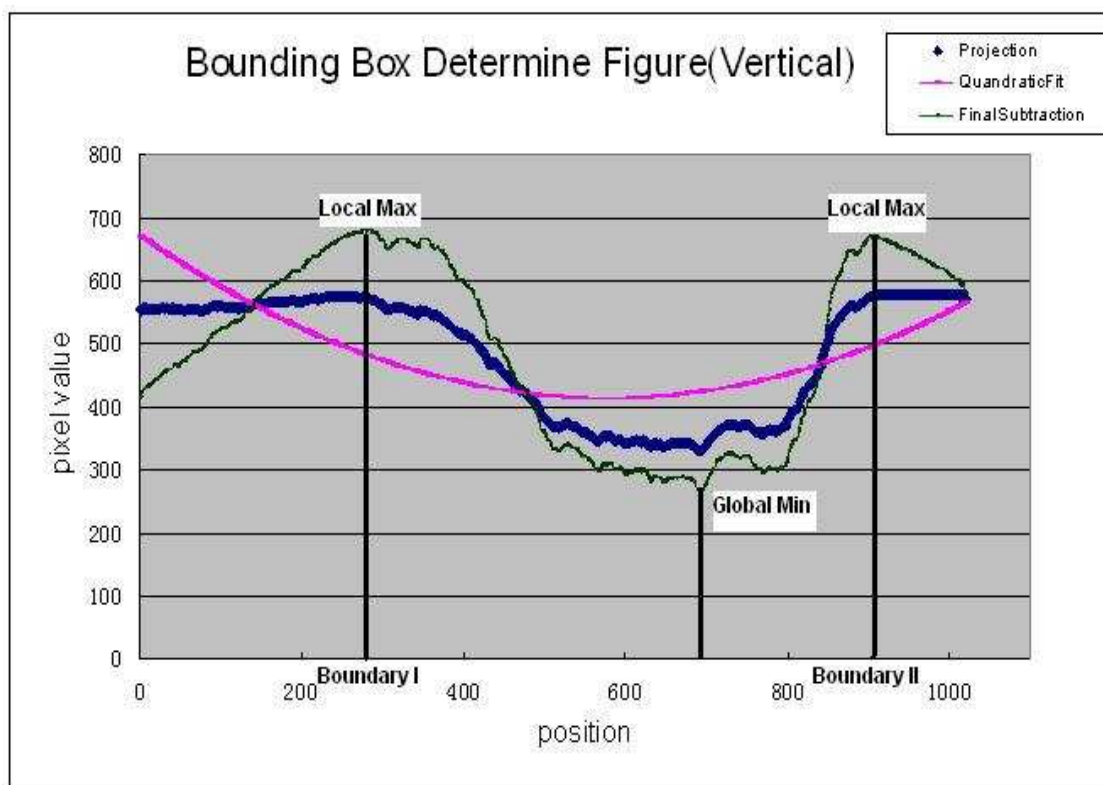


Figure 4.7 Outer bounding box boundaries

Generally, the lesion area is included in the outer bounding box, but according to specific conditions, some parts of the lesion may extend out of the bounding box. Figure 4.8 shows the outer bounding box superimposed on a lesion image. In this image, the vertical relative maxima are at $X=278$ and 908 and the horizontal relative maxima at $Y=90$ and 633 . Based on the X and Y coordinates, the outer bounding box ratio, which is the ratio of the outer bounding box area to the image area, is 0.5012 for this image.

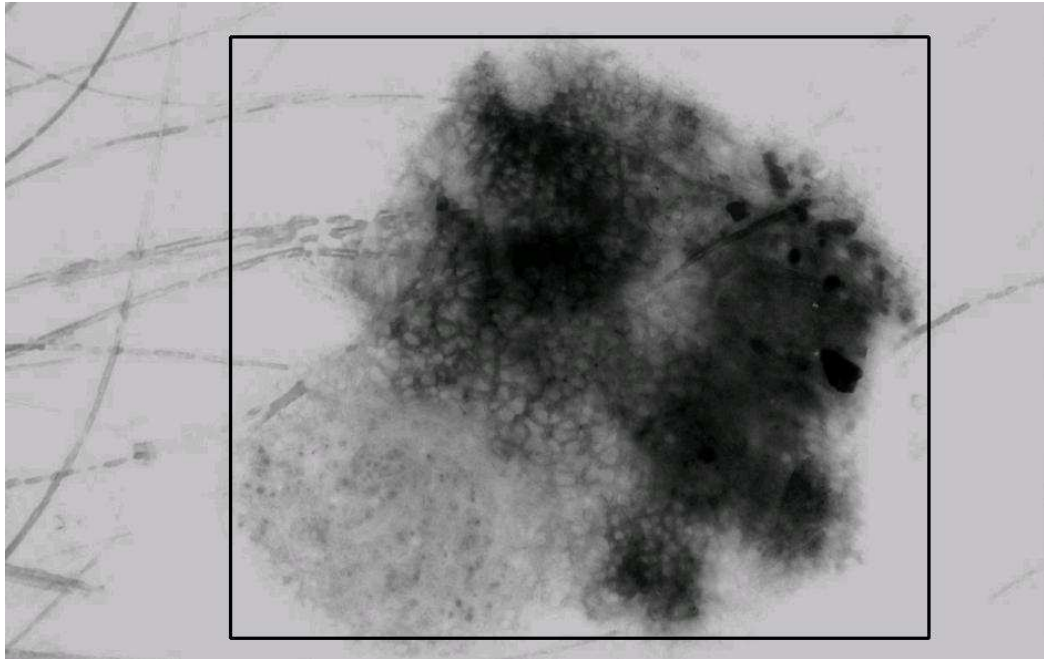


Figure 4.8 Outer bounding box

Different from the original outer bounding box finding method [15], instead of processing the whole image, 30 lines in from the image edge in each direction are ignored, so that unresolved black borders and bright bubbles in these areas can be neglected in the projection. Also, we start looking for the global minimum from 25% of the image horizontally and vertically and stop at 75% of the image in each direction, because the accumulation of the darkest parts in lesions is within this ratio of image and that is where the global minimum is located. Using the new outer bounding box processing method, the program running time in this part can be up to 75% less theoretically. Figure 4.9 presents an illustration using this approach:

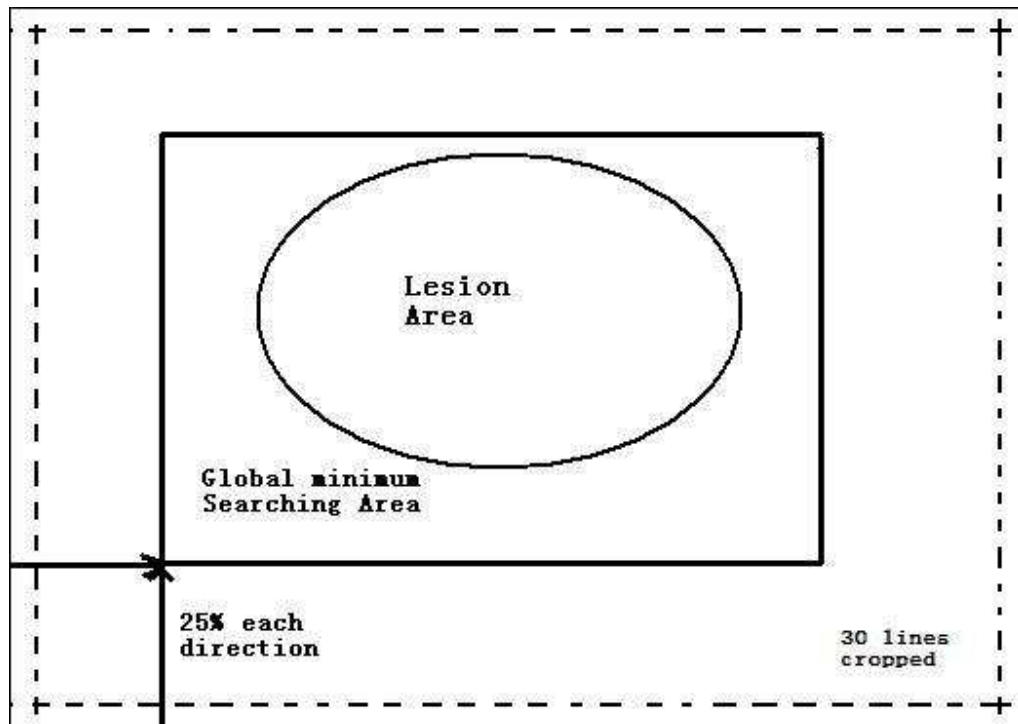


Figure 4.9 Improved global minimum searching

4.5. EDGE OBJECT THRESHOLD

One factor that can affect the merging process and produce error in the watershed border is the “large bright edge blobs”. As we already know, the watershed blobs are divided into different areas with a variety of shapes, which are determined by the average pixel value in the particular area. Generally, there is a sharp pixel value drop at the lesion boundary. If the pixel value drop is not in a sharp gradient, after the watershed approach, large bright watershed objects can exist with part of the object being inside of the lesion and the other larger part being outside of the lesion. After the merging application, this object will produce either false negatives or false positives. Most of the time, a false positive error will occur.

One method to solve this problem is to use an “edge object threshold (*EOT*)”. It is defined as:

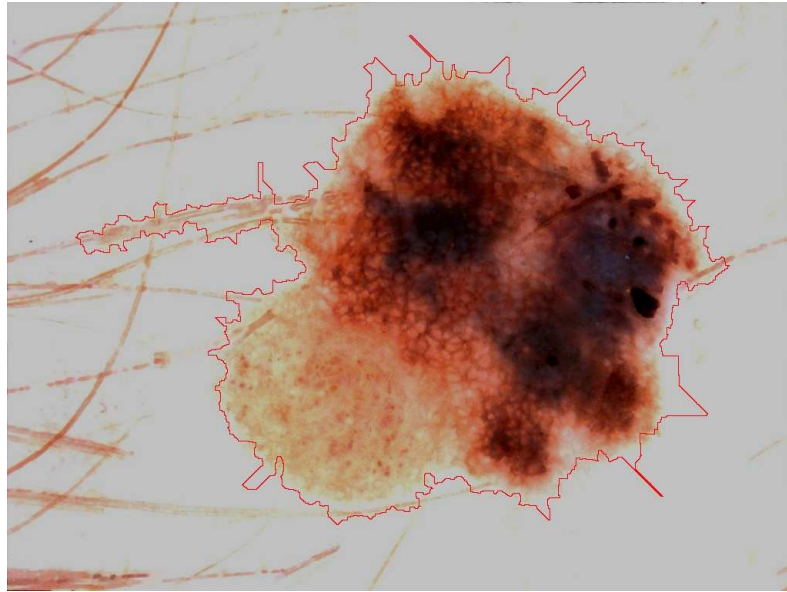
$$EOT = R \times A / B$$

R -----Average watershed object blue value

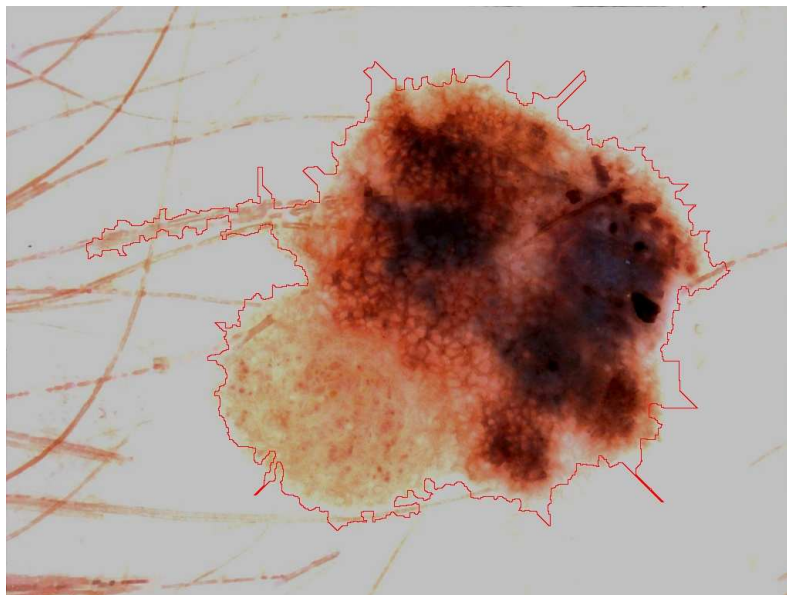
A -----Watershed object area size

B -----Average lesion blue value

Since those large areas are always lighter than the inside lesion part, meaning those objects have a higher pixel value than other blobs inside the lesion area, the product of the area average blue value and the size of the area will be much larger than the same product for the neighboring blobs. The average lesion blue value in the denominator is used to normalize the threshold value. After testing throughout the 100 image set, a number of ‘50,000’ was selected as the optimal threshold to cut off those large bright areas at the lesion boundary. There are some slight false negatives due to the cut-off application, which will result in smaller lesion areas, but those errors will be corrected in the peninsula removal method in following steps. In our sample image, several threshold values are used to show the process of removing those blobs gradually, noticing that some peninsula-shaped noise is removed. The three different thresholds (see Figure 4.10) are: 1000, 1500 and 3000. Note that a threshold of 50,000 gives the same result as a threshold of 3000 for this image.

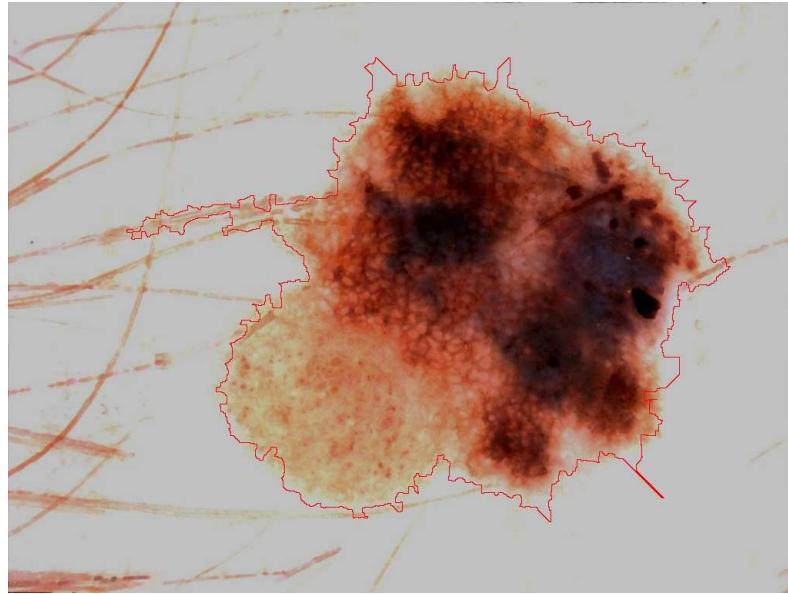


(a)



(b)

Figure 4.10 Edge threshold effect. Threshold Value: (a) 1000; (b) 1500; (c) 3000



(c)

Figure 4.10 (cont.) Edge threshold effect. Threshold Value: (a) 1000; (b) 1500; (c) 3000

4.6. LESION RATIO ESTIMATE (LRE)

Based on 100 images obtained from Stoecker & Associates, including 70 benign images and 30 malignant melanoma images, with lesion ratios ranging from 0.0528 to 0.5362, Figure 4.11 shows the relationship between the outer bounding box ratio, the actual lesion ratio and the lesion ratio estimate.

The pink dots show the actual lesion ratio, which are based on the average of three dermatologists' manually drawn borders; the dark blue line is the previous lesion ratio estimate from Xiaohe Chen's application [15]:

$$LRE = B \cdot 0.7799 - 0.0436$$

where B is the outer bounding box ratio. The weights in the formula are based on his lesion ratio results. In Figure 4.11 for his linear fit, there is some departure in the low lesion ratio area. Due to the difference of the watershed algorithm results using

MATLAB® and Visual C++, the bounding box size and shape may vary. After applying the best linear fit on actual lesion ratio dots, we get the brown line shown as the new lesion ratio estimate:

$$NewLRE = B \cdot 0.8633 - 0.1919$$

This formula tries to improve the error in the low lesion ratio area and get the best fit line in the whole lesion ratio range. Note that there are around ten images whose actual lesion ratio is a significant distance from the best-fit line. This may be due to outer bounding boxes that are significantly larger than the lesion area due to noise in the background skin, such as black border, low contrast, lesion area, etc. Thus, further work needs to be done on outer bounding box correction. Figure 4.11 shows the relations of outer bounding box, LRE and $NewLRE$.

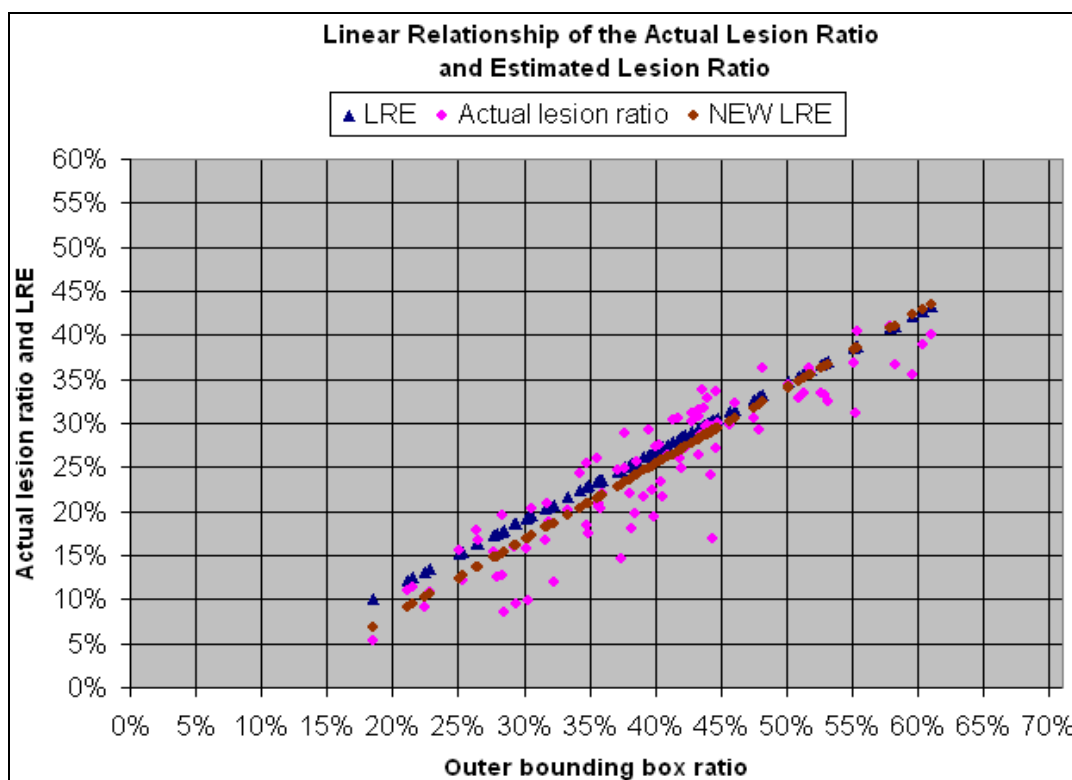


Figure 4.11 Linear relationship between actual lesion ratio and lesion ratio estimate

In the appendix, statistical tables are shown for the actual lesion ratio and lesion ratio estimate.

4.7. LRE CORRECTION BASED ON NEURAL NETWORK

In Xiaohe Chen's dissertation [15], a Random Forest data mining algorithm (the Weka implementation) was applied to optimize the segmentation, which used color statistics from these first estimates. Forty-four features were acquired from the original images and used as the inputs to the Random Forest Algorithm [15].

Another classification method is introduced in this research using a supervised neural network. Generally a biological neural network is composed of groups of chemically connected or functionally associated neurons. A single neuron may be connected to many other neurons and the total number of neurons and connections in a network may be large. Apart from electrical signals, there are other forms of signals that arise from neurotransmitter diffusion, which have an effect on the electrical signal, so neural networks are extremely complex. Artificial intelligence and cognitive modeling try to simulate some properties of neural networks. While similar in their techniques, the former has the aim of solving particular tasks, while the latter aims to build mathematical models of biological neural systems.

Artificial neural networks have been applied successfully to speech recognition, image analysis and adaptive control, in order to construct software agents (in computer and video games) or autonomous robots [26]. Currently, most of the employed artificial neural networks for artificial intelligence are based on statistical estimation, optimization and control theory. The cognitive modeling involves the physical or mathematical

modeling of the behaviors of neural systems; ranging from the individual neural level, through the neural cluster level to the complete organism [26].

In supervised learning, we are given a set of example pairs $(x, y), x \in X, y \in Y$ and the aim is to find a function f in the allowed class of functions that matches the examples. In other words, we wish to infer how the mapping implied by the data and the cost function is related to the mismatch between our mapping and the data [26].

Due to correlation between different features, a feature-reduction procedure was carried out to eliminate those less significant features; after that, a total of seventeen features were saved for further use. Starting with the updated seventeen features, some of the features were eliminated and some other useful features were added. For example, first-iteration lesion area is in the seventeen nonlinear features, but it is not crucial for the lesion area size; mean value of watershed rim is linear related, so we added it as an input feature. This results in a set of twenty features. These features are: mean value of watershed rim in blue and red; average value in three planes of the image (red, green and blue); three planes (red, green and blue) of the high peaks of the object histogram; lesion ratio estimate of blue plane; pixel histogram standard deviation in luminance and blue planes; histogram variances in luminance plane; and the inner and outside average color in four planes (red, green, blue and luminance). The supervised neural network algorithm does not indicate which features are the strongest discriminators.

Based on Dr. Stanley's supervised back-propagation neural network, a neural network was used to try to classify the lesion ratio estimate into one of three clusters: proper estimate, over estimate, under estimate. The first class represents the case where the lesion ratio estimate $\hat{\gamma}_B$ is within 5% of the actual lesion ratio; the second class

represents the case where the lesion ratio estimate $\hat{\gamma}_B$ is at least 5% above the actual lesion ratio; the third class represents the case where the lesion ratio estimate $\hat{\gamma}_B$ is at least 5% below the actual lesion ratio. The final lesion ratio estimate is:

$$\gamma = \hat{\gamma}_B + 0.025 \cdot p \quad p = \begin{cases} 0 & \text{proper estimate} \\ -1 & \text{over estimate} \\ 1 & \text{under estimate} \end{cases}$$

γ -----final lesion ratio estimate

$\hat{\gamma}_B$ -----lesion ratio estimate

For the lesion ratio estimate, the average error value is 14.08%. The average lesion ratio is 0.25 to the whole image, so 2.5% of image area is around 10% of the lesion area. The value of 0.025 indicates adding or subtracting 2.5% of the whole image.

Ninety images were used as the training set, ten images as the testing set, with a different test set every time and taking the average so that each image is used as the input and output. The RMSE is around 1.4 and the training stops at 10 epochs. A three-bit binary number is used for clustering:

$$Cluster \left\{ \begin{array}{ll} 0 : & 001 \\ -1 : & 010 \\ 1 : & 100 \end{array} \right.$$

The most significant output is assigned as 1, otherwise, is assigned as 0.

Compared to the target, the supervised neural network is able to correctly classify 70% of the images into the three clusters.

4.8. NOISE CONTROL---BORDER SMOOTHING

4.8.1. Peninsula Removal. Peninsula-shaped noise is one of the key types of noise on the primary watershed border. One way to remove that kind of noise uses mathematical morphology [27], which is a technique for the analysis and processing of geometrical structures, based on set theory, lattice theory, topology, and random functions. Mathematical morphology is most commonly applied to digital images, but it can be employed as well on graphs, surface meshes, solids, and other spatial structures.

Erosion and dilation are two important morphological tools. While erosion shrinks image objects, dilation expands them. Generally, erosion decreases the sizes of objects and removes small anomalies by subtracting objects with a radius smaller than the structuring element. In our sample binary image, a 3×3 pixel matrix is used as the radius to process erosion. Erosion completely removes objects smaller than the structuring element and removes perimeter pixels from larger image objects. After that, a dilation operation is applied to the resulting image. Dilation generally increases the sizes of items, filling in holes and connecting areas that are separated by spaces smaller than the size of the structuring element. In our binary eroded image, dilation connected areas that are separated by spaces smaller than the structuring element, a 3×3 pixel matrix, and added pixels to the perimeter of each image object.

In this step, first we define the radius as:

$$R = k \times \sqrt{A}$$

k----- Tunable parameter (1/20 in this project)

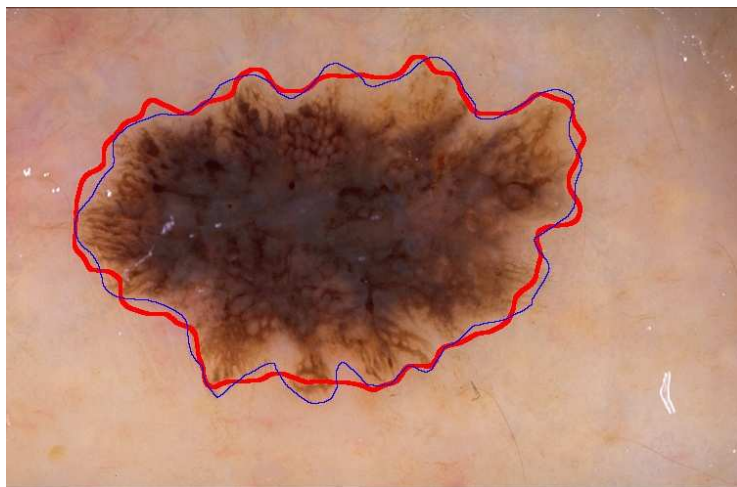
A----- Lesion area

Then, using an erode mask of radius R , the isolated regions are removed; using a dilate mask of radius R , the holes both inside and outside the lesion are filled. According to the peninsula-shaped objects, first the connecting part between the peninsula and the main lesion is cut off by the erosion operation, and then the isolated “island” is removed by the dilation operation. Also, for the second step, a black frame is added around the image (width equal to structuring element radius) to prevent the image edge from being dilated. This is based on Thomas Mark Daniel Szalapski’s peninsula removal program.

The draft border is smoothed while the main shape of the border is not changed. These two steps are applied to smooth the region borders which are usually irregular due to peninsula shaped noise. Again, the application is due to the duplications of structures, such as pigmented networks on the first pass watershed borders. After the successive dilation step, the small peninsulas are being filled.

4.8.2. Border Sampling and B-Spline Border Smoothing. Second-order B-Spline closed curve fitting [28] is applied to the peninsula-smoothed border of the lesion mask. First, a list of border pixels $[X, Y]$ is generated. Second, a smoothing procedure records the starting point then traverses along the boundary and records all the pixels in the path. When the distance from the starting point exceeds P pixels in either the x or the y direction, the point on the lesion mask is recorded. An average control point is calculated as the average of all the pixels’ coordinates along the path from the starting point. After an iterative calculation, all the pixels with P -pixel distance on the mask are marked [15]. The average x -coordinate and y -coordinate of these pixels are used as a control point for B-Spline smoothing.

Three tentative values for P were tested: 8, 16 and 32. The numbers 8 and 16 work better in jagged borders, since 32 ignores the sharp change within a longer distance. On the other hand, the number of 32 wins overall in the 100-image set, because most of the lesion rims are not jagged and the longer distance makes the border smoother and reduces the error. Figure 4.12 shows the different impacts, the red border is the final watershed border; the blue border is the actual manually drawn border.

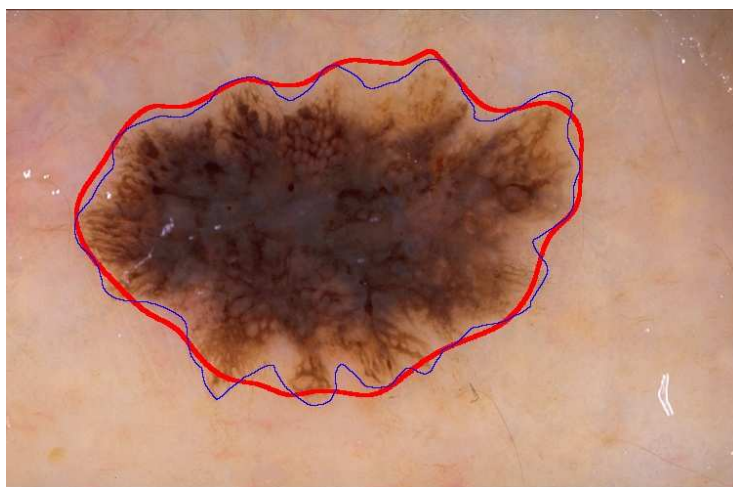


(a)



(b)

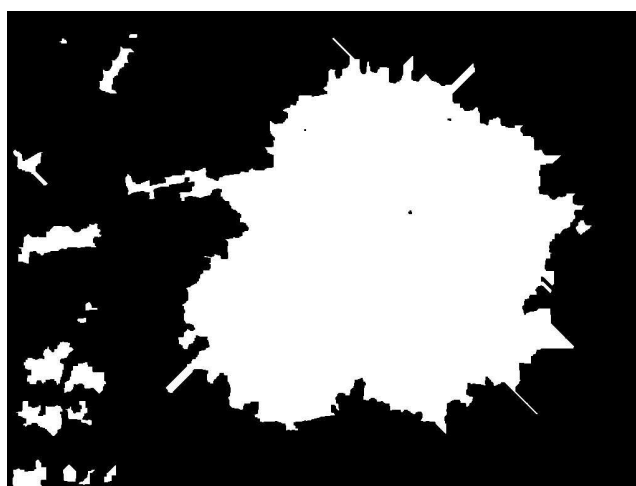
Figure 4.12 Sampling interval impact. (a) interval of 8 (b) interval of 16 (c) interval of 32



(c)

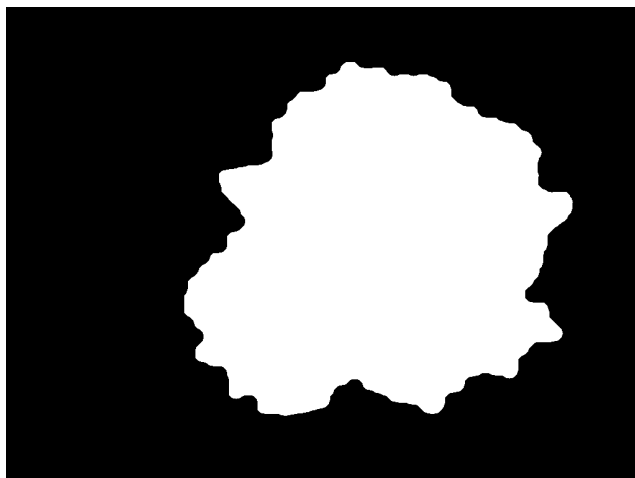
Figure 4.12 (cont.) Sampling interval impact. (a) interval of 8 (b) interval of 16 (c) interval of 32

Figure 4.13 shows the steps of the smoothing method. Based on the binary image mask, the peninsula shaped noises are removed, while holes inside the lesion are filled. After that, the jagged border generated from peninsula removal is smoothed by a B-Spline fitting curve.

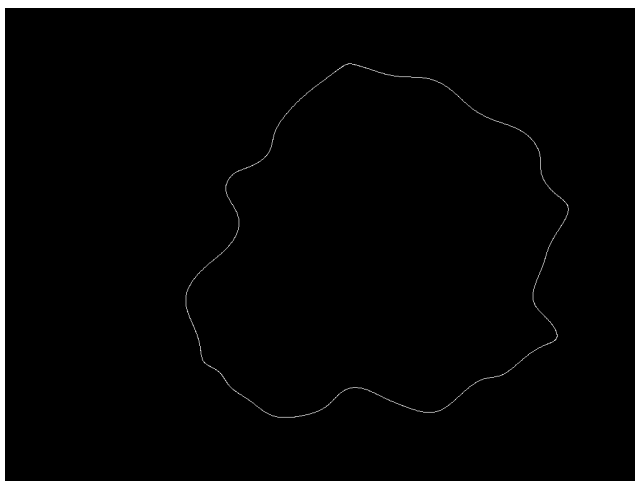


(a)

Figure 4.13 Smoothing methods. (a) watershed mask; (b) peninsula removal; (c) B-Spline



(b)

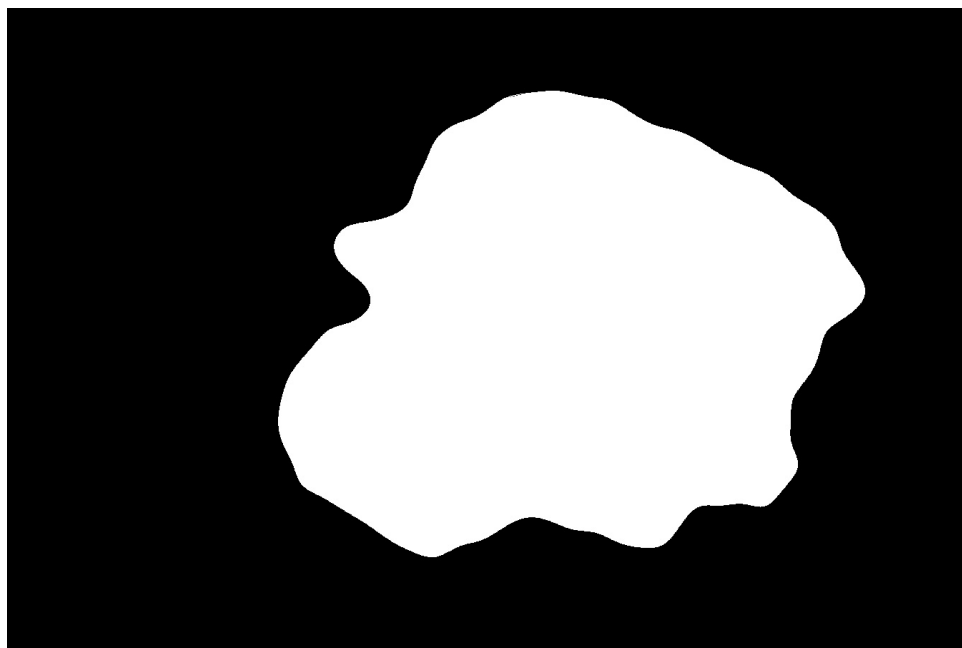


(c)

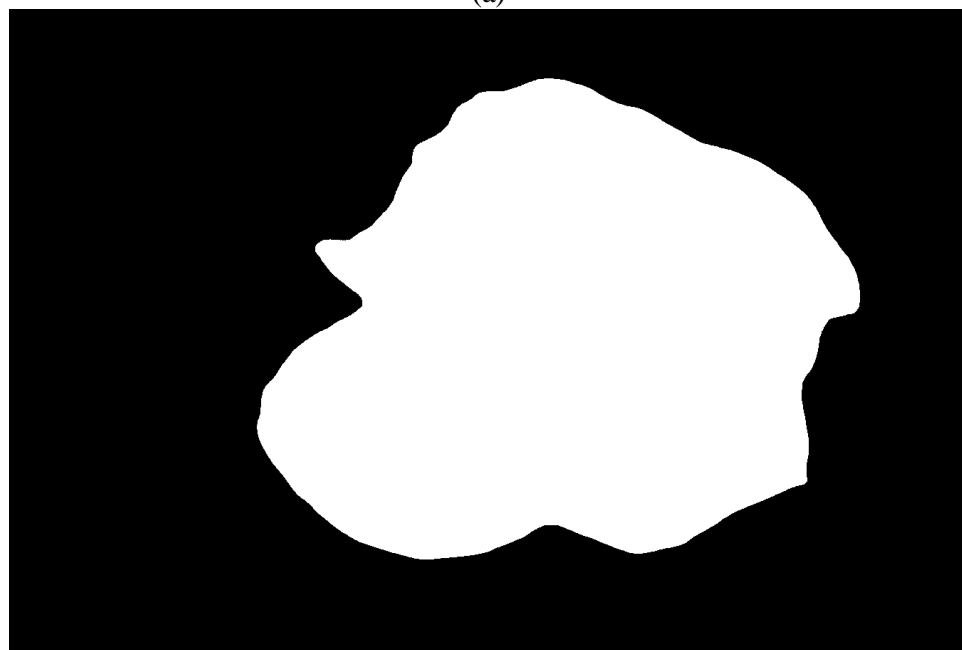
Figure 4.13 (cont.) Smoothing methods. (a) watershed mask; (b) peninsula removal; (c) B-Spline

5. RESULTS

Figure 5.1 shows a comparison between the watershed final result mask and the average border mask of three dermatologists' borders.



(a)



(b)

Figure 5.1 (a) Final watershed border and (b) actual border

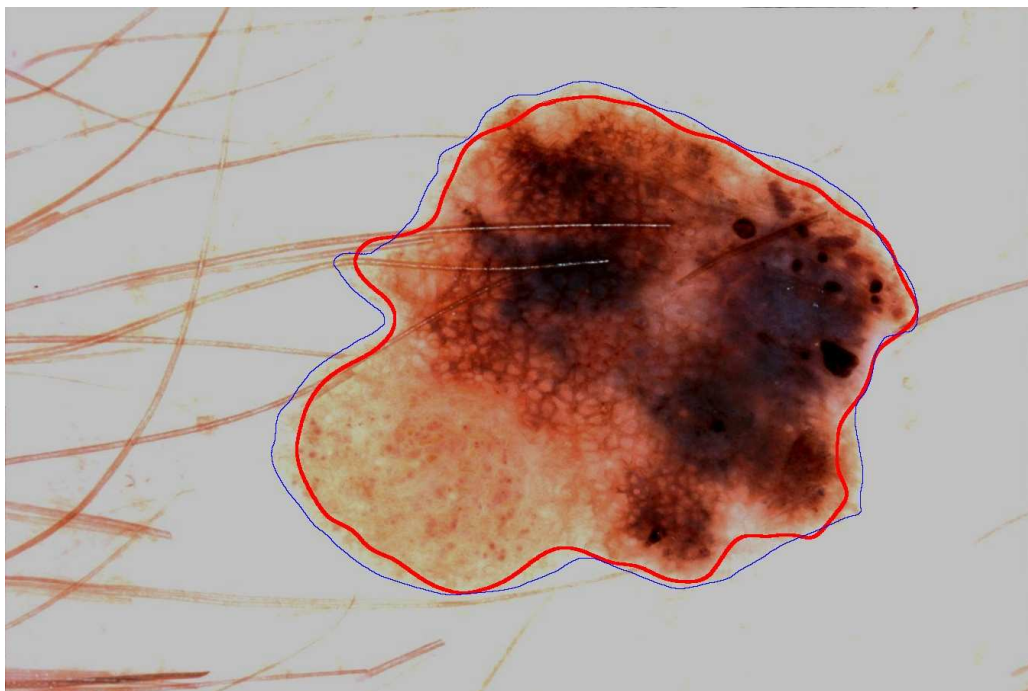


Figure 5.2 Final mask layover

In Figure 5.2, the thick red line is the watershed border and the thin blue line is the average of three dermatologists' manually drawn borders. Using an XOR operation to calculate the error, this image gives a total error of 7.591% with 6.948% false negative and 0.6420% false positive. The XOR operation is looking for the difference between input and target image based on the logical operation:

$$p \oplus q$$

p -----input image

q -----target image

Table 5.1 is the truth table for XOR operation :

Table 5.1 XOR truth table

p	q	XOR
F	F	F
F	T	T
T	F	T
T	T	F

Figure 5.3 shows the XOR operation between the watershed mask and manually drawn border; the white area stands for the mask difference error.

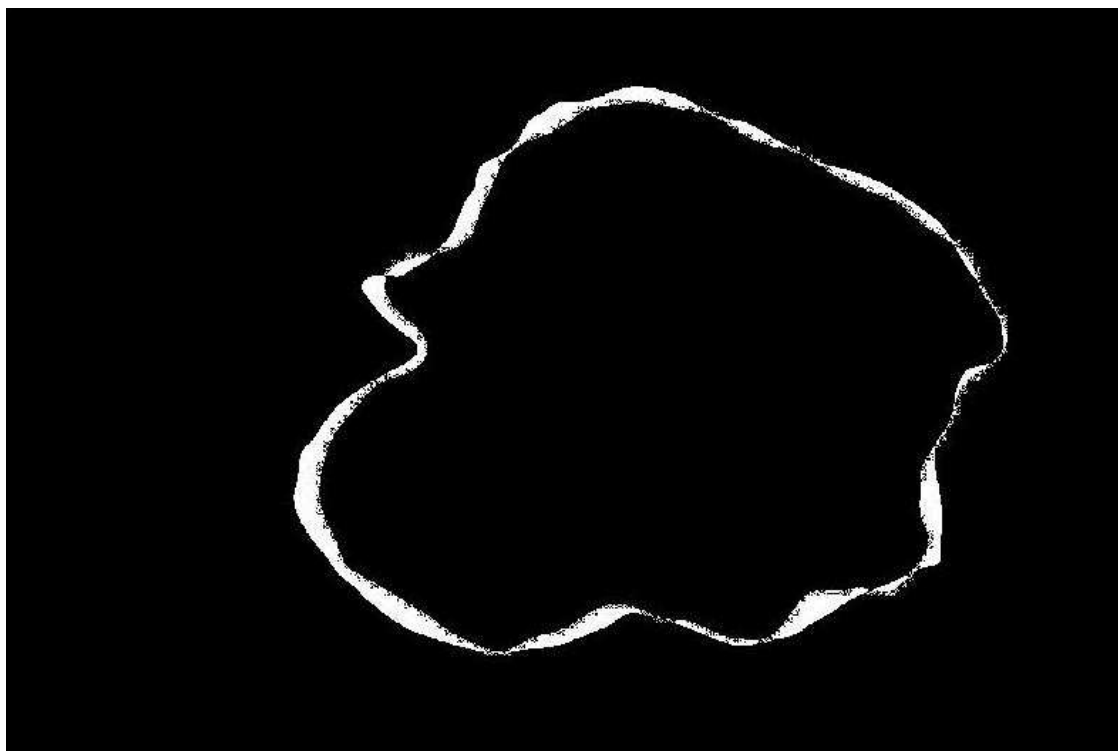
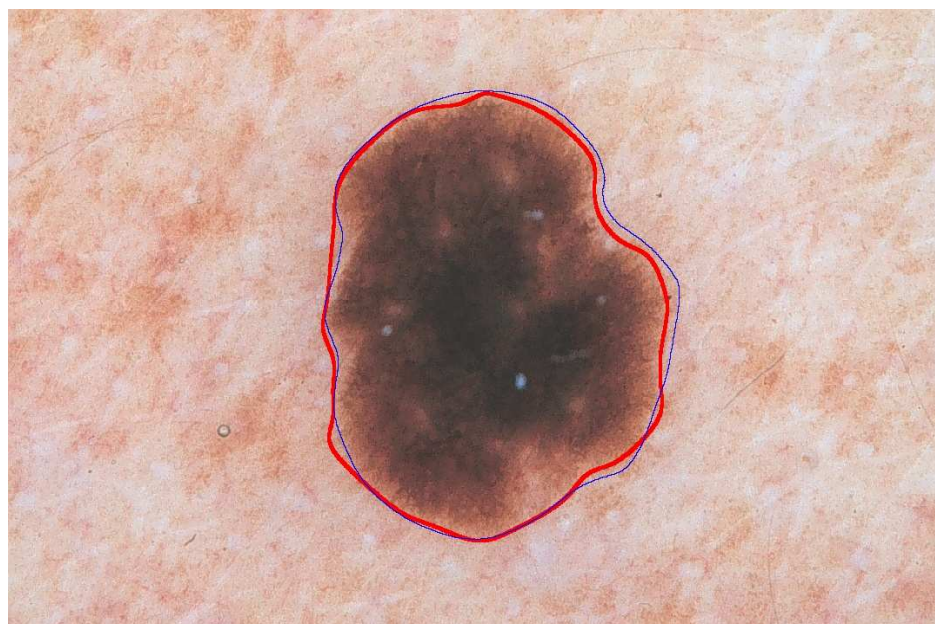
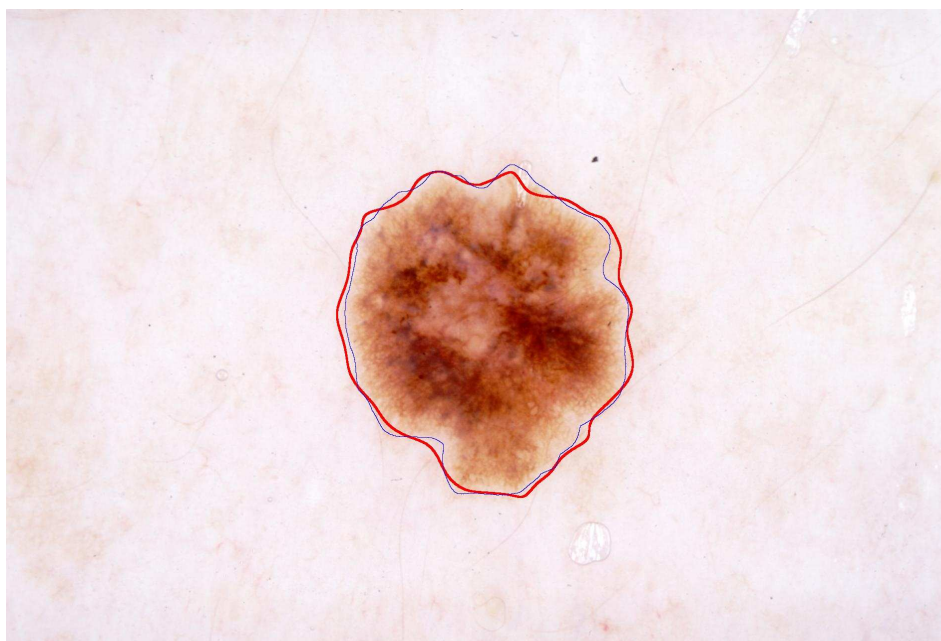


Figure 5.3 XOR operation image

Five other results are shown in Figure 5.4:

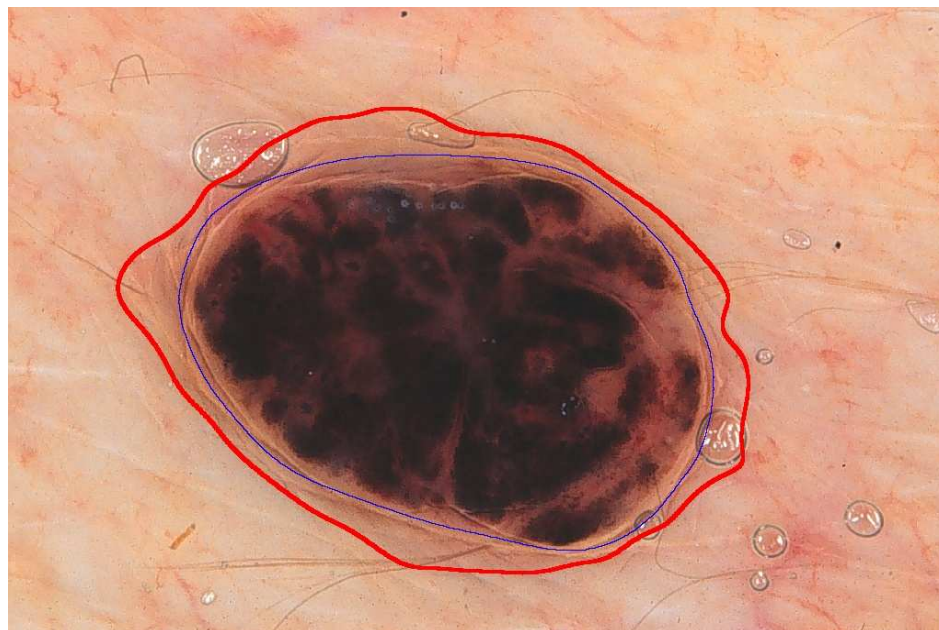


(a)

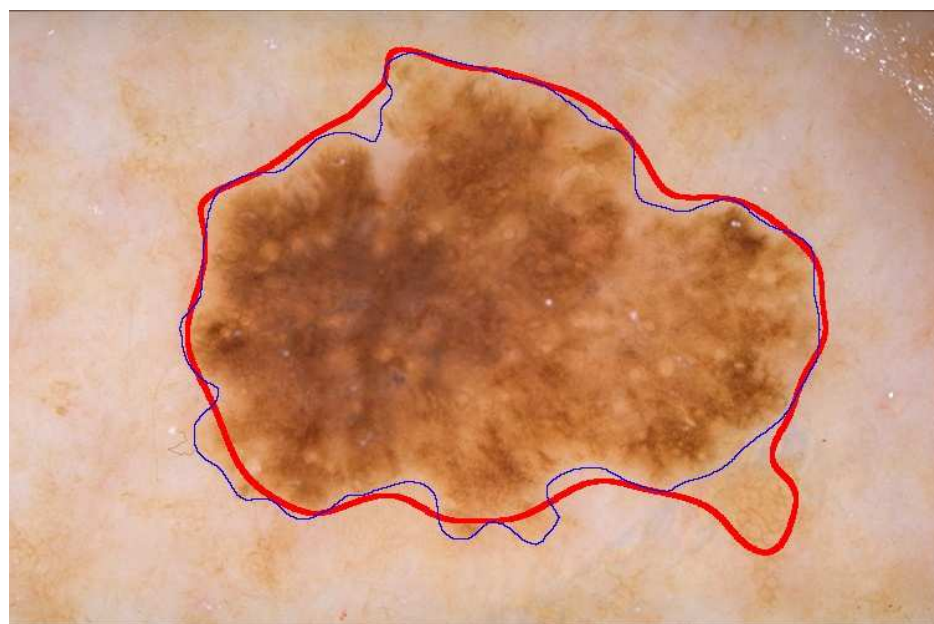


(b)

Figure 5.4 Results and error values

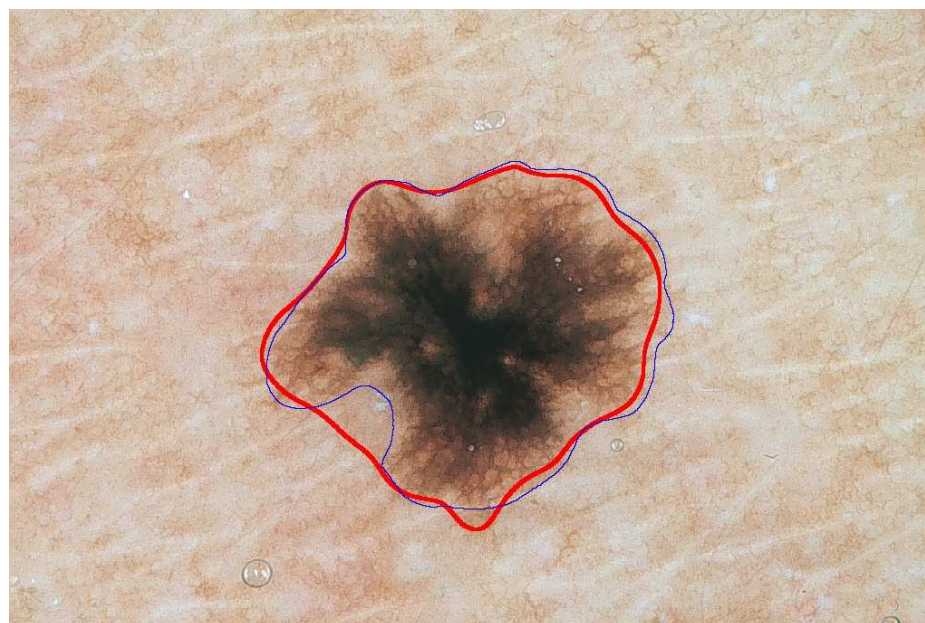


(c)



(d)

Figure 5.4 (cont.) Results and error values.



(e)

Figure 5.4 (cont.) Results and error values: (a) 5.584% in total, 2.862% false positive, 2.721% false negative; (b) 7.773% in total, 6.466% false positive, 1.307% false negative; (c) 29.72% in total, 29.72% false positive, 0.0000% false negative; (d) 11.44% in total, 9.382% false positive, 2.061% false negative; (e) 10.61% in total, 6.152% false positive, 4.451% false negative.

Figure 5.4 (c) (d) and (e) show the three most common shapes of errors: (c) error due to lesion rim structure or color; (d) outside peninsula shape error; (e) inside peninsula shape error. For the 100-image set, the average total error is 11.12%, average false positive is 9.204% and average false negative is 1.912%.

A table for the 100-image set error is shown in the appendix.

6. CONCLUSION AND FUTURE WORK

The watershed algorithm provides a useful method for lesion segmentation. The lesion ratio estimate is important additional information added to the watershed algorithm to control the lesion size. This can be determined iteratively, allowing correction of errors made at the first iteration. In the preprocessing, hair, black border and vignetting removal removes some of the noise affecting our lesion segmentation, thus helping the watershed algorithm to segment the image similarly to the way dermatologists do. The post-processing gives an acceptable final segmentation result. This research also completes the transform from MATLAB® code to Visual C++ code for the watershed algorithm and, in addition, modifies the watershed algorithm by allowing the size of the object area to be specified.

In this research, the outer bounding box [15] method gives a good first approximation for the lesion ratio estimate based on the range of 0.0528 to 0.5362 of the actual lesion ratios. The primary watershed method is robust and able to detect and draw borders similar to the border obtained by dermatologists. In post-processing, edge thresholding solved the problem with big blobs at the lesion border and gave a closer result to the actual lesion ratio. Furthermore, as a supervised neural network was applied for clustering to improve the accuracy, an accuracy of 70% was achieved in classifying the estimate comparing to the average of three dermatologists' manually drawn borders. Three clusters: proper estimate, under estimate and over estimate classifies all 100 images and improved the accuracy up to 2.5%. Noise removal of border smoothing using peninsula removal and B-Spline methods provide a satisfactory final border.

Future work will involve looking for more reliable features for the classifier, while other networks will be employed for comparison. Also, peninsula-shaped noise is another key problem to be solved or reduced in the next step.

APPENDIX
IMAGE TABLE

This table illustrates the result and 100-image set information, including image name, lesion ratio estimate, bounding box ratio, actual lesion ratio, total error, false positive and false negative. The average total error is 11.12%, average false positive is 9.204% and average false negative is 1.912%.

Image Name	LRE	Bounding Box Ratio	Lesion Ratio	Total Error	FP	FN
kk0498ad.tif	0.0680	0.1853	0.0528	0.1252	0.0679	0.0573
zr0798ad.tif	0.1545	0.2854	0.0856	0.1843	0.1836	0.0007
Rk0399ad.tif	0.1017	0.2243	0.0915	0.1323	0.1288	0.0035
4084d.tif	0.1619	0.2940	0.0954	0.1476	0.1460	0.0016
95-13-15.tif	0.1699	0.3032	0.0992	0.3144	0.3132	0.0012
As0598ad.tif	0.1055	0.2286	0.1081	0.0641	0.0555	0.0086
hm1096ad.tif	0.0903	0.2110	0.1101	0.0620	0.0067	0.0553
Ld1196ad.tif	0.0944	0.2158	0.1133	0.0777	0.0647	0.0130
6089d.tif	0.1864	0.3223	0.1193	0.2363	0.2358	0.0005
Fj1200ad.tif	0.1263	0.2527	0.1214	0.0911	0.0895	0.0016
6688d.tif	0.1489	0.2790	0.1256	0.2062	0.2062	0.0000
mj0798ad.tif	0.1527	0.2834	0.1270	0.0720	0.0286	0.0434
6714d.tif	0.2305	0.3735	0.1466	0.1941	0.1931	0.0010
6025d.tif	0.1475	0.2773	0.1541	0.0991	0.0950	0.0041
96-3-21.tif	0.1242	0.2504	0.1558	0.0599	0.0240	0.0359
6023d.tif	0.1685	0.3017	0.1577	0.1060	0.0615	0.0445
6687d.tif	0.1613	0.2933	0.1604	0.0977	0.0493	0.0484
6706d.tif	0.1816	0.3168	0.1667	0.0736	0.0576	0.0160
6005d.tif	0.1373	0.2655	0.1675	0.0891	0.0799	0.0092
Fj0297ad.tif	0.2916	0.4442	0.1683	0.2186	0.2154	0.0032
4082d.tif	0.2097	0.3493	0.1753	0.1179	0.1179	0.0000
Bg0899ad.tif	0.1363	0.2643	0.1779	0.0941	0.0450	0.0491
mk1096ad.tif	0.2378	0.3819	0.1809	0.1593	0.0945	0.0648
kh0598ad.tif	0.2080	0.3474	0.1836	0.2073	0.0190	0.1883
4030d.tif	0.1838	0.3193	0.1871	0.0790	0.0361	0.0429

6054d.tif	0.2523	0.3987	0.1932	0.1567	0.1492	0.0075
Mm0899ad.tif	0.1523	0.2829	0.1950	0.1124	0.0131	0.0993
6091d.tif	0.2403	0.3848	0.1977	0.0610	0.0060	0.0550
Kr1196ad.tif	0.1956	0.3330	0.2005	0.0891	0.0809	0.0082
6707d.tif	0.2169	0.3577	0.2022	0.1191	0.1180	0.0011
6031d.tif	0.1719	0.3055	0.2027	0.0725	0.0505	0.0220
6612d.tif	0.2165	0.3572	0.2053	0.1088	0.0704	0.0384
gc0697ad.tif	0.1824	0.3177	0.2086	0.0766	0.0716	0.0050
6094d.tif	0.2159	0.3565	0.2089	0.0558	0.0286	0.0272
6709d.tif	0.2587	0.4061	0.2169	0.1594	0.1507	0.0087
6077d.tif	0.2461	0.3916	0.2170	0.1454	0.1438	0.0016
6599d.tif	0.2180	0.3590	0.2195	0.1311	0.1306	0.0005
6074d.tif	0.2362	0.3800	0.2210	0.0905	0.0658	0.0247
4083d.tif	0.2514	0.3977	0.2247	0.2000	0.1999	0.0001
6085d.tif	0.2573	0.4045	0.2331	0.0547	0.0426	0.0121
Ra0199ad.tif	0.2895	0.4418	0.2408	0.0676	0.0673	0.0003
Ma0899ad.tif	0.2037	0.3424	0.2433	0.0576	0.0322	0.0254
4074d.tif	0.2285	0.3711	0.2467	0.0962	0.0896	0.0066
6051d.tif	0.2336	0.3771	0.2484	0.0666	0.0470	0.0196
6084d.tif	0.2706	0.4199	0.2489	0.0810	0.0804	0.0006
6024d.tif	0.2082	0.3477	0.2553	0.0592	0.0197	0.0395
6627d.tif	0.2411	0.3858	0.2567	0.1805	0.1425	0.0380
6079d.tif	0.2699	0.4191	0.2599	0.0897	0.0092	0.0805
95-13-17.tif	0.2147	0.3551	0.2603	0.1536	0.0074	0.1462
Fn1200ad.tif	0.2620	0.4100	0.2646	0.1750	0.1731	0.0019
4058d.tif	0.2820	0.4331	0.2648	0.1641	0.1641	0.0000
6050d.tif	0.2703	0.4196	0.2719	0.1845	0.1845	0.0000
Wa0499ad.tif	0.2727	0.4223	0.2721	0.0887	0.0880	0.0007
6028d.tif	0.2930	0.4458	0.2722	0.2004	0.2004	0.0000
6102d.tif	0.2537	0.4003	0.2736	0.0439	0.0165	0.0274
6093d.tif	0.2565	0.4035	0.2748	0.0351	0.0285	0.0066
6642d.tif	0.2827	0.4340	0.2872	0.2082	0.2082	0.0000
Bm0300ad.tif	0.2338	0.3773	0.2892	0.0620	0.0322	0.0298
6605d.tif	0.3218	0.4792	0.2926	0.2972	0.2972	0.0000
6617d.tif	0.2486	0.3945	0.2931	0.0770	0.0112	0.0658

Lj0899ad.tif	0.2897	0.4421	0.2967	0.1509	0.1505	0.0004
zp0298ad.tif	0.2869	0.4388	0.2970	0.1145	0.1114	0.0031
6622d.tif	0.2876	0.4396	0.2977	0.1444	0.1416	0.0028
6061d.tif	0.3028	0.4572	0.2979	0.0875	0.0859	0.0016
4063d.tif	0.2949	0.4480	0.3001	0.1112	0.1112	0.0000
RI0298ad.tif	0.2776	0.4280	0.3028	0.1064	0.0493	0.0571
6677d.tif	0.2648	0.4131	0.3042	0.0785	0.0613	0.0172
hn0996ad.tif	0.2680	0.4169	0.3061	0.0495	0.0089	0.0406
4023d.tif	0.3178	0.4746	0.3064	0.0940	0.0749	0.0191
6053d.tif	0.2819	0.4330	0.3076	0.1366	0.1281	0.0085
6015d.tif	0.2769	0.4272	0.3109	0.0631	0.0319	0.0312
6049d.tif	0.3850	0.5524	0.3114	0.1384	0.1368	0.0016
6667d.tif	0.2824	0.4336	0.3145	0.0359	0.0145	0.0214
lj1196ad.tif	0.2860	0.4377	0.3166	0.0759	0.0148	0.0611
6637d.tif	0.3060	0.4609	0.3219	0.0981	0.0882	0.0099
4044d.tif	0.3667	0.5313	0.3251	0.1784	0.1782	0.0002
6643d.tif	0.3475	0.5090	0.3290	0.1671	0.1671	0.0000
mm0598ad.tif	0.2877	0.4397	0.3293	0.0759	0.0064	0.0695
6075d.tif	0.3647	0.5289	0.3330	0.1463	0.1457	0.0006
6044d.tif	0.3514	0.5135	0.3341	0.0853	0.0813	0.0040
6644d.tif	0.3620	0.5258	0.3351	0.1802	0.1748	0.0054
jb0298ad.tif	0.2936	0.4465	0.3351	0.1000	0.0976	0.0024
96-22-19.tif	0.2840	0.4354	0.3381	0.0708	0.0278	0.0430
6045d.tif	0.3411	0.5016	0.3427	0.0674	0.0590	0.0084
Gs0499ad.tif	0.3406	0.5010	0.3443	0.1020	0.0776	0.0244
6056d.tif	0.4226	0.5960	0.3543	0.0851	0.0673	0.0178
4025d.tif	0.3556	0.5184	0.3569	0.1144	0.0938	0.0206
6055d.tif	0.3549	0.5175	0.3619	0.1240	0.1163	0.0077
6675d.tif	0.3239	0.4817	0.3627	0.0555	0.0299	0.0256
Nd0599ad.tif	0.4109	0.5824	0.3671	0.0635	0.0602	0.0033
6048d.tif	0.3836	0.5508	0.3682	0.0942	0.0790	0.0152
Ra0997ad.tif	0.4287	0.6031	0.3895	0.2229	0.2228	0.0001
6602d.tif	0.4352	0.6106	0.4008	0.2222	0.2222	0.0000
6065d.tif	0.3860	0.5536	0.4037	0.0669	0.0544	0.0125
6017d.tif	0.4078	0.5788	0.4088	0.1046	0.1033	0.0013

km0301ad.tif	0.4077	0.5787	0.4103	0.0978	0.0037	0.0941
4056d.tif	0.4064	0.5772	0.4222	0.1003	0.1003	0.0000
6105d.tif	0.4291	0.6035	0.4839	0.0839	0.0751	0.0088
6052d.tif	0.3754	0.5413	0.5152	0.0985	0.0844	0.0141
sd0597ad.tif	0.4737	0.6551	0.5362	0.0580	0.0151	0.0429
			Average	0.1112	0.0920	0.0191

BIBLIOGRAPHY

- [1] A. Jemal, R. Siegel, E. Ward, et al. "Cancer Statistics, 2008". *CA: A Cancer Journal for Clinicians* 2008; Vol. 58, No. 2, pp. 71-96, 2008.
- [2] Z. Zhang, W.V. Stoecker, and R.H. Moss, "Border Detection on Digitized Skin Tumor Images," *IEEE Trans. Med. Imag.*, vol. 19, no. 11, pp. 1128-1143, 2000.
- [3] J. S. Tau, G. H. Lee, C. W. Tao, C. C. Chen, and C. W. Yang. "Segmentation of Psoriasis Vulgaris Images Using Multiresolution-Based Orthogonal Subspace Techniques" *IEEE Trans on Systems, Man, and Cybernetics—Part B: Cybernetics*, vol. 36, no. 2, pp. 1011-1023, April 2006
- [4] Y. W. Lim and S. U. Lee, "On the Color Image Segmentation Algorithm Based on the Thresholding and the Fuzzy C-means Techniques," *Pattern Recognit.* vol. 23, no. 9, pp. 935–952, 1990.
- [5] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active Contour Models," in *Proc. 1st ICCV*, 1987, pp. 259–267.
- [6] P. L. Palmer, H. Dabis, and J. Kittler, "A Performance Measure for Boundary Detection Algorithms," *Comput. Vis. Image Understand.*, vol. 63, pp. 476–494, 1996.
- [7] R. Adams and L. Bischof, "Seeded Region Growing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 6, pp. 641–647, Jun. 1994.
- [8] R. M. Haralick and L. G. Shapiro, "Survey: Image Segmentation Techniques," *Comput. Vis. Graph. Image Proces.*, vol. 29, pp. 100–132, 1985.
- [9] Y. L. Chang and X. Li, "Adaptive Image Region-growing," *IEEE Trans. Image Process.*, vol. 3, no. 6, pp. 868–872, Nov. 1994.
- [10] S. A. Hijjatoleslami and I. Kitter, "Region Growing: A New Approach," *IEEE Trans. Image Process.*, vol. 7, no. 7, pp. 1079–1084, Jul. 1998.
- [11] T. Pavlidis and Y. T. Liow, "Integrating Region Growing and Edge Detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 3, pp. 225–233, Mar. 1990.
- [12] J. Haddon and J. Boyce, "Image Segmentation by Unifying Region and Boundary Information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 10, pp. 929–948, Oct. 1990.
- [13] C. Chu and J. K. Aggarwal, "The Integration of Image Segmentation Maps Using Region and Edge Information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 12, pp. 1241–1252, Dec. 1993.

- [14] H. Ganster, A. Pinz, R. Rohrer, E. Wildling, M. Binder, and H. Kittler, "Automated Melanoma Recognition," *IEEE Trans. Med. Imag.*, vol. 20, no. 3, pp. 233-239, 2001.
- [15] X. Chen, "Skin Lesion Segmentation by an Adaptive Watershed Flooding Approach", Ph. D. Dissertation in Electrical Engineering, University of Missouri-Rolla, 2007.
- [16] A.L. Bleau and L. Joshua, "Watershed-based Segmentation and Region Merging," *Computer Vision and Image Understanding*, vol. 77, no. 3, pp. 317-370, 2000.
- [17] C. Sun and P. Vallotton, "Fast Linear Feature Detection Using Multiple Directional Non-Maximum Suppression," The 18th International Conference on Pattern Recognition (ICPR'06), vol. 13, 2003, pp. 49-53
- [18] P. Schmid, "Lesion Detection in Dermatoscopic Images Using Anisotropic Diffusion and Morphological Flooding" Proceedings of the 6th International Conference on Image Processing (ICIP '99), vol. 3, 1999, pp. 449-453 Boston/Dordrecht/London : SPIE, 1999.
- [19] E. Hecht. *Optics*. Addison-Wesley, 4th edition, 2002.
- [20] N. Asada, A. Amano, and M. Baba. "Photometric Calibration of Zoom Lens Systems," *Proc. IEEE Int. Conf. on Pattern Recognition*, pp. 186-190, Aug. 1996.
- [21] W. Yu, Y. Chung, and J. Soh, "Vignetting Distortion Correction Method for High Quality Digital Imaging", 17th International Conference on Pattern Recognition (ICPR'04) pp. 666-669 Cambridge, UK
- [22] F. Shafait, D. Keysers, and T. M. Breuel, "Response to Projection Methods Require Black Border Removal," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 13, No. 4, pp. 1211-1215. April 2009
- [23] L. Najman and M. Schmitt. "Geodesic Saliency of Watershed Contours and Hierarchical Segmentation", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 18, No. 12, pp. 1163-1173. 1996
- [24] B. Daku. *Learn Matlab Fast: The M-Tutor Interactive Tutorial*. John Wiley & Sons, Incorporated. August 2005
- [25] Dr. G. Bradski; A. Kaehler. *Learning OpenCV*. 1st Edition, O'Reilly Media. Inc. October 3, 2008.
- [26] http://en.wikipedia.org/wiki/Neural_network Last referred 02/14/2009

- [27] Z. Yu, Y. Zhao, X. Wang. “Research Advances and Prospects of Mathematical Morphology in Image Processing”. 2008 IEEE Conference on Cybernetics and Intelligent Systems, vol. 10, 2008, pp.1242 – 1247.
- [28] C. de Boor, *A Practical Guide to Splines*, Springer-Verlag, 2nd edition., New York, NY, 2001.

VITA

Hanzheng Wang was born on March 1, 1986 in Shangqiu, China. He completed his primary education in Shangqiu. His secondary education was at Zhengzhou, China. He graduated with a Bachelor's degree in Electronics of Science and Technology from the College of Information Science and Technology, Beijing Institute of Technology, China in July 2007.

He enrolled for the Master of Science in Computer Engineering at the University of Missouri-Rolla in June 2007. He worked as a Graduate Research Assistant with the DERMVIS group of skin cancer research from June 2007 to May 2009. He completed his Master of Science degree in Electrical Engineering in May 2009.