HIGH-ACCURACY SKIN LESION SEGMENTATION AND SIZE DETERMINATION

A Thesis

presented to

the Faculty of the Graduate School

University of Missouri-Columbia

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Computer Science

By

JAE WON SHIN

Dr. Yi Shang, Project Supervisor

DECEMBER 2011

The undersigned, appointed by the dean of the Graduate School, have examined the Project entitled

HIGH-ACCURACY SKIN LESION SEGMENTATION AND SIZE DETERMINATION

Presented by Jae Won Shin

A candidate for the degree of Master Science in Computer Science

And hereby certify that, in their opinion, it is worthy of acceptance.

Dr. Yi Shang

Dr. Ye Duan

Dr. Jianlin Cheng

ACKNOWLEDGEMENTS

I would like to appreciate my advisor, Dr. Yi Shang for his continuous support during my master degree thesis and for inspiring me during my master's study. Also, I would like to appreciate to Dr. Ye Duan and Dr. Jianlin Cheng who are members of the defense committee for insightful comments and for their encouragement.

Last but not least, thanks to my family, my parents Taesub Shin and Jonggyun Park and my sister Kyungran Shin for all of their support for me, and finally, thanks to my friends.

ABSTRACT

Melanoma is one of the most common skin cancers. 5% of all cancer cases that occur in a year are cases of skin cancer and many people die each year by melanoma because of late recognition. Moreover, the number of new cases of melanoma in the United States has been increasing for at least 30 years. Fortunately, this melanoma is highly curable if it is detected early. In order to detect skin lesions, the automated skin lesion segmentation and diagnosis system on the Android system is an outstanding program to use.

In this thesis, there are 4 functionalities, which include: camera data collecting, image processing, feature calculation, and classification. With these processes, an image is captured and is converted to a grey image. After a grey image is created, a skin lesion contour is found by OpenCV functions, cvFindContour() and cvWatershed(). When the contour is found, the features color, shape, and size of the lesion are extracted in order to classify whether the lesion is benign or malignant by using the KNN classifier.

The goals and achievements of this thesis are to implement a function that captures images with the Android using camera properties, improve image segmentation and size estimation based on the previous prototype system on the Android platform with OpenCV. An image can be captured by a capturing function in this thesis and can be saved in a jpg file and a data xml file, which are used for image processing and camera features. In image processing, the watershed function is used instead of cvThreshold function, which is on the previous prototype system, so that the number for the threshold

value is no more needed and the system can find the most method contour instead of finding several contours as it does in the previous system. In size estimation, two methods, which are reference method and camera distance method, are used. Reference method is when the system can estimate the area of a lesion by comparing reference pixels and lesion pixels. Camera distance method is when the system can estimate the area of a lesion according to camera distance properties, which are near, optimal, and far camera distance values. With these two methods, the system can estimate real area of a lesion without rulers instead of counting the number of pixels.

Contour detection is improved to 98%. Reference size estimation and camera focus distance size estimations are 1.04% and 8.23%

Table of Contents

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

1.1 Melanoma Skin Cancer

Melanoma cancer, a type of skin cancer, which is the most common cancer of all cancers, grows in melanocytes that are melanin-producing cells. This cancer not only easily grows in the outer layer where melanocytes are located on the skin, but also can easily spread to other parts of the body, bones and internal organs such the lungs. Therefore, Melanoma is lethal for people who do not detect it in the early stages. According to the American Cancer Society (ACS), even though melanoma cancer is only 5% of all skin cancer cases, it is the cause of most skin cancer deaths. However, Melanoma can be easily cured if it is detected early. [1] Melanoma in individuals 10~39 years old is highly curable, with 5-year survival rates exceeding 90% according to 2009 cancer facts from ACS [2].

Fortunately, there is a recommended decision rule for detecting Melanoma, which is called the "ABCD rule'. It is useful for people to determine whether they have melanoma or not by using the rules. As shown in Figure 1, if a mole changes in a short period of time or keeps changing, if one half of a lesion is different than other half, if the border of a lesion is not a round shape or looks like a blur, if the color of a lesion is dark in color, or if the diameter of a lesion is greater than 6mm, then the mole needs to be shown to a doctor.

Asymmetry Border Color Diameter

Figure 1 - Examples of ABCD rule for detecting Melanoma [4]

1. Asymmetry – One half is different than the other half.
2. Border – The lesion of border is irregular, uneven, and unclear.
3. Color – The color is unequal. There are shades of brown, tan, and black.
4. Diameter – The size of lesion is greater than 6mm.

Although melanoma is fatal in the late stage of cancer, it is curable in the early stages. The ABCD rule is an excellent method to test whether a lesion is benign and malignant.

Benign Malignant

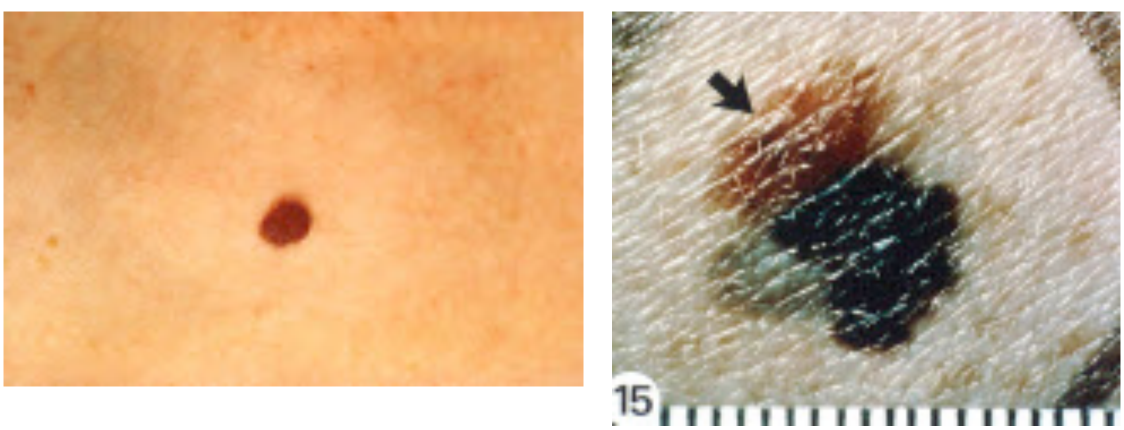

Figure 2 - Pictures of benign and malignant

1.2 Related Works and Current Technology

The Automated Melanoma Recognition system, which was developed by H.Ganster et al, can recognize melanoma by image analysis using over 122 features and over 5000 images in several steps: Imaging Process, segmentation, feature calculation, and feature selection and classification.

Recently, in the mobile app markets, several apps have been published. One of them is Skin Scan on the iPhone shown in Figure 3 [5]. This app can find melanoma easily and anyone can use it, but it takes quite a long time and easily gives analysis failures in unclear images. In this thesis, the processing time is less than 10 seconds, which is hardly a wait at all.

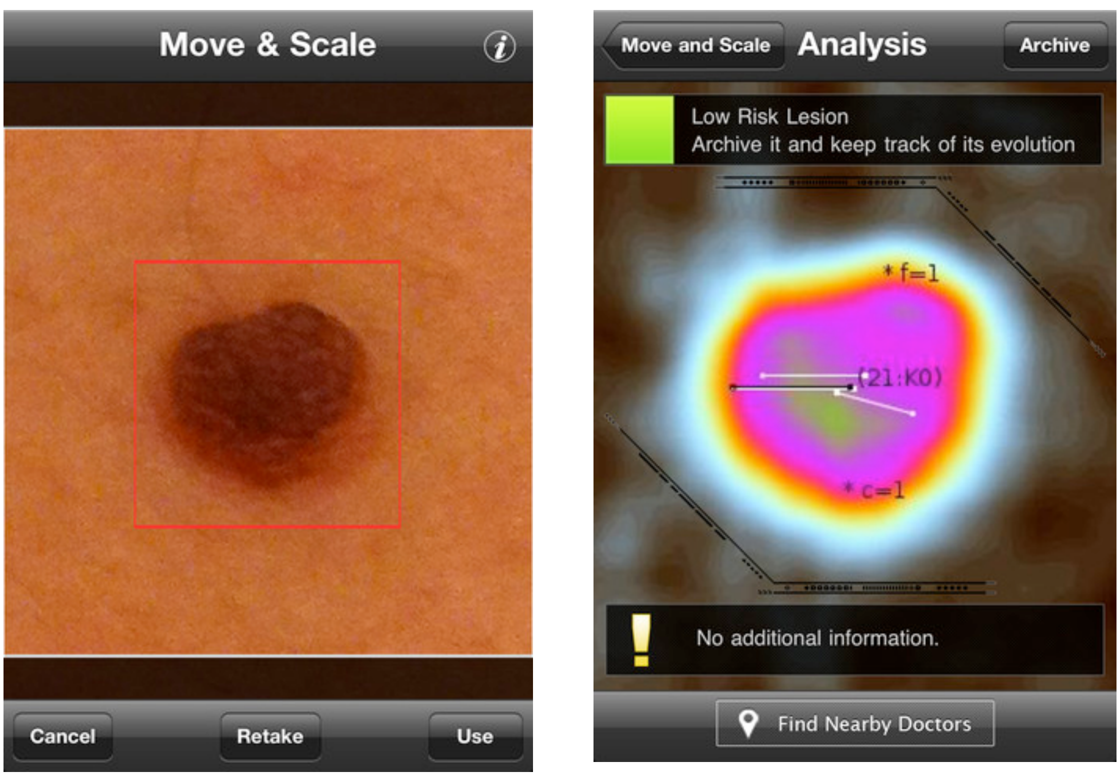

Figure 3 – Image of "Skin Scan" Apps

Another app is called Skin of Mine that uses the ABCD rules for analysis, as shown in figure 4 [6]. However, the app does not determine whether a mole is melanoma or not. It only gives the user a warning if a dangerous lesion is found based on the ABCD rules. Differently than Skin of Mine, this system chooses what kind of mole the picture portrays based on 2 classes: benign and malignant. The analysis in the thesis is more precise and detailed than this app.

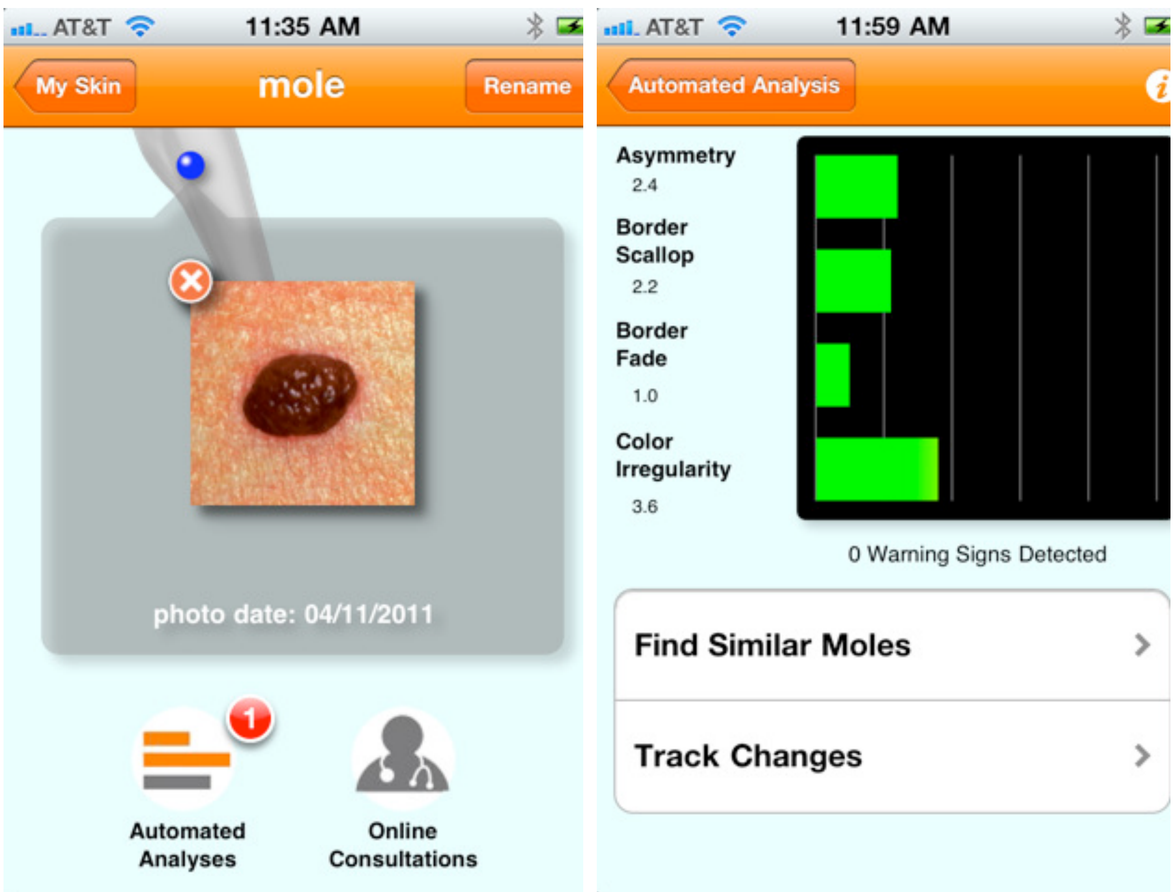

Figure 4 – Image of "Skin of mine" Apps

1.3 Motivation

Mobile devices have been changing and have been becoming faster, so many tasks that computers do are moved into mobile devices such as the Android, IOS, and Windows Mobile. With fast and handy smartphone devices, people can do a lot of their tasks such as checking their banking accounts, searching information via web browsers, and shopping. In a recent survey, 42% of all people who have cell phones and 83% of American adults have smartphones [7]. This means that 35% of the world's population use smartphones. Smartphones have become typical and common devices in society,as people can easily access data and run apps on them.

An important fact for detecting melanoma is early detection because melanoma in an early stage can be cured but in a late stage is hard to cure. People typically see a doctor if they has some symptoms of melanoma. However, it could be too late to even see a doctor if the melanoma has already entered the late stages. Self-testing can be very helpful in this situation.

Smartphones are a common device and melanoma detection is needed immediately, so this why mobile devices can be helpful for people to do self-testing for melanoma. Also, the device can collect pictures with the camera function and can make decisions with algorithms.

1.4 Starting the Automated Skin Lesion Classification

This thesis is started from a project, called "The Design and Implementation of a Mobile, Automated Skin Lesion Classification System" [8]. This system can determine if a lesion is malignant or benign by using OpenCV on the Android system. To do that, users load an image from a SD card, and use image processing such as cvFindContour and cvThreshold, and KNN (K-Nearest Neighbor Algorithm) classification. Also, to extract features like color and shapes of a lesion, RGB comparison and convex hull finding are used.

However, this system needs some improvements such as lesion contour detection and size estimation. When the system detects a lesion contour, it finds several look-alike contours by a threshold number. This can work for pictures with good detection conditions, but not in all cases. When the system estimates a lesion size, it extracts pixel data from a picture but cannot calculate a real lesion size. This is a critical feature in order to classify a malignant or benign lesion.

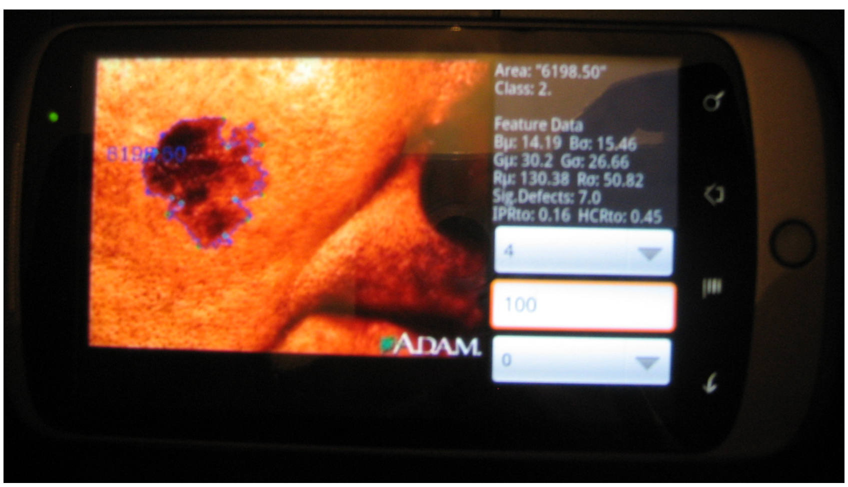

Figure 5 - The Design and Implementation of a Mobile, Automated Skin Lesion Classification System

1.5 The Goal

The main goal of this thesis is to improve the performance of finding and analyzing melanoma. To this end, the scope of this thesis involved saving an image in a SD card with camera properties, finding a lesion contour using the watershed function with guide lines by user drawing, and estimating a lesion size with two methods: the distance method with camera properties and the reference method using a quarter (25-cent coin) as reference.

2. DESIGN AND IMPLEMENTATION

2.1 System Architecture

In this thesis, image capture/selection, image processing, and feature extraction are involved in the automated melanoma detection system, as shown in Figure 6, which shows the flow chart of the entire system architecture.

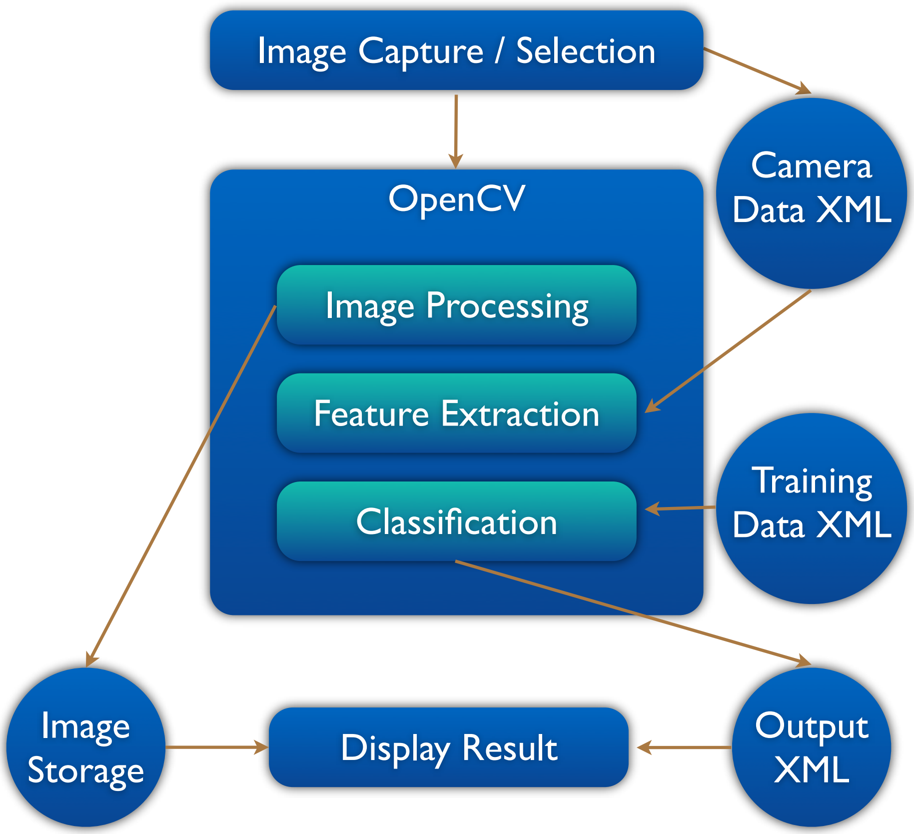

Figure 6 - Architecture Diagram of Auto Melanoma detection system

The entire system runs on the Android smartphone system 2.3 with OpenCV 2.3.1. In the previous system, the version of Android SDK is 2.2. Since the newest Android SDK is 2.3 and has some new functions such as camera properties, the Android system in this project has been updated.

One of the major functions of this project is image processing, so OpenCV, which is an open source image processing kit that runs on the Android system is used. Since OpenCV is written in C/C++, the Android system needs a way to import C/C++ codes. NDK is needed to do this.

The following sections further describe the design and implementation of system functionalities.

2.2 Image Capture/Selection

2.2.1 Image Capture

To diagnose melanoma, image capture is the first step because an image provides excellent data for smartphone devices and their sensors. When lesion images are captured, an Android device can get camera near, optical, and far focus values. This function, getting the camera values, is a new function of the the Android SDK 2.3. Those values and images are saved into JPEG and XML files which are related to the same file names that consist of Patient ID, date, and time. When an image loads, the system also loads a XML data file that has same name for melanoma classification.

Figure 7 - Screenshot of Image capture via an android phone

Figure 7 shows a screenshot of an image capture. There are patient ID, lesion classification, and a capture button. Patient ID is used for cancer tracking. When several lesions are captured or a lesion is captured several times, each lesion image has its own ID, so the system can identify by ID. Below the patient ID, there is a list box, called lesion classification. The purpose of the lesion classification field is collecting doctors' diagnosis for a lesion. The field is for collecting training data for classification. When the capture button is clicked, capture image is created in the SD card on an Android device, and other data is saved into a XML file. The data in the XML file is as follows:

1. Image filename: (Patient ID)_(Date)(Time).jpg
2. Patient ID
3. Camera focus values (cm)
    a. Near, Optimal, Far
4. Camera focus values (m)
    a. Near, Optimal, Far

Pseudo code

1. Call the callback function autofocus when a user starts to capture.
2. In the callback function, get camera parameters
3. Call take-picture function to capture an image and callback function for saving
4. In the image-capture callback function, save an image as a JPEG file and camera parameters as a XML file with patient ID, date, and time into the SD-card
5. Refresh SD-card

2.2.2 Image Selection

The purpose of image selection is to get two contour guidelines from users. The role of the guidelines is for one of them to represent the inside lesion and the other to represent the outside lesion. With these two lines, the system is able to recognize the outside lesion area and inside lesion area by finding the contour process. In order to get guidelines, two display layers are on the screen, as shown in figure 8.

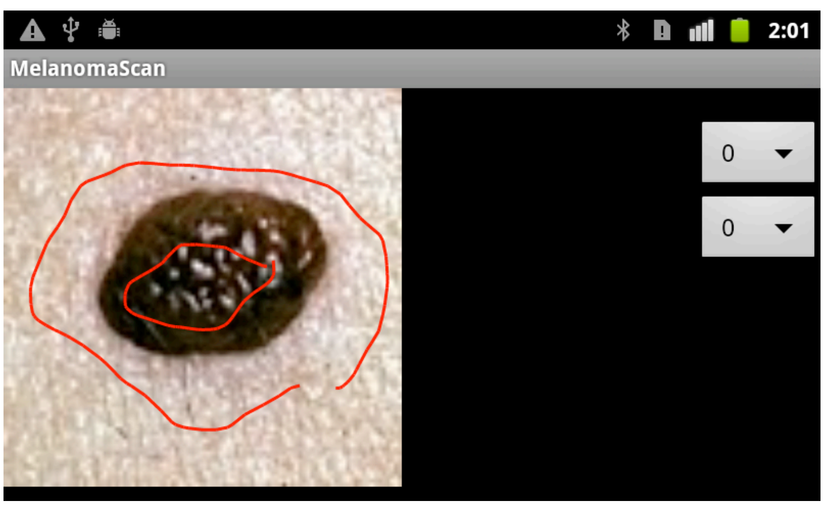

Figure 8 – a screen-capture of drawing guidelines for Image selection

This figure 9 has two display layers: lesion layer and guideline layer. On the lesion layer, a lesion image is displayed, and on the guideline layer, two guidelines, an inside lesion line and an outside lesion line are displayed.

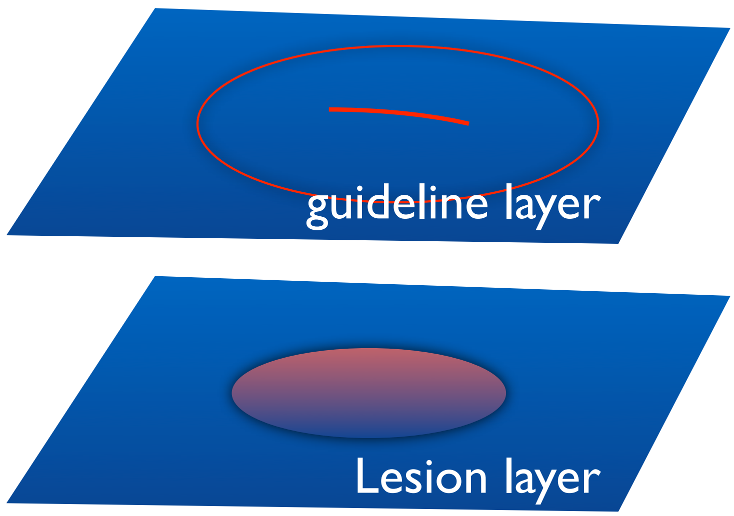

Figure 9 – a diagram of the screen consists of a guideline layer and a lesion layer

Pseudo code

1. Create and place a view layer on the top of layers on the screen with line setup, color and width of lines.
2. When users touch and draw lines, start and end positions of lines are stored in an array and drawn on the screen
3. If drawing reset is needed, empties the array and refreshes the screen.
4. Once user selection is done, the array goes to OpenCV process

2.3 Image Processing

The objective of this function is to detect a lesion contour on pictures. User guidelines from the image selection function are required to use lesion contour detection. Users indicate which area is the lesion area by the lines. Drawing more accurate lines gets more accurate contour results at unclear border images. The system finds and draws the best matching lesion contour with the guidelines.

2.3.1 Lesion Contour Detection

Before lesion contour detection is begun, original source image, converting the image from the original image to signed 32 bits image, and gray-scale source image is needed. Original source image is loaded from the SD-card, 32 bits image is an input image to the watershed function and shows edges of source image and guidelines from the image selection in 2.2.2.

First of all, an input of the watershed function is a 32 bits image of edges and user guidelines. User guidelines are directly drawn from the image selection in 2.2.2, and edges are derived from the original image by using the finding contour method, OpenCV implementation of Suzuki and Abe's (1985) Border Following algorithm [9].

Finding Contours

Various edge detection methods exist and are used for finding contours in various images. In this thesis, Suzuki and Abe's (1985) [9] Border Following algorithm is implemented to find contours before and after the watershed function.

Before using the watershed function, image edges that are starting points of the watershed function are detected approximately because those edges are only starting points of edge detection. After using the watershed function, it returns a black and white image. Getting contours from the simple black and white image can be easier than color images. Therefore, in this project, Suzuki and Abe's (1985) Border Following algorithm [9] is a good enough method to use.

Suzuki and Abe's (1985) Border Following algorithm [9] is a contour finding function and implements in OpenCV. It computes contours from binary images. To determine contours, the algorithm assigns labels to each sequence of connected components of 0 and 1 pixel along with raster scans of pixels. If hole pixels (interior boundary) and contour pixels (exterior boundary) are found during raster scans, sequential numbers are assigned to the boundaries, and they are the labels of the borders.

The other input of the watershed function is a single channel gray-scale image that is converted from the source image. This gray-scale image will be one of the watershed function inputs. A following equation changes RGB color pixels in the original source image to Luminance value (Y):

$$Y = 0.299R + 0.587G + 0.114B \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(1)$$

After edge detection from an image and guideline drawing are finished, and the gray-scale image is ready, the system can retrieve outer and inner lesion areas from those images by using Meyer's (1992) Watershed algorithm [10].

Watershed algorithm

Many segmentation and edge detection algorithms divide boundaries with certain criteria and affect the entire image, but they do not care to separate background images, inside objects, or outside objects. Watershed is one of the techniques that can distinguish a separate background and an object.

The principal of the method is like a flood in mountain and basin areas. If water comes from the top of mountains, the water goes to the basin (lower areas). Therefore, the lower areas are gradually under the water if the water flows continuously. Finally, the two regions, under the water and above the water, are divided. In detail, the watershed algorithm detects basin areas, which have no textures, and mountain areas, which are dominant edges from image.

To divide two regions, there is one more rule regarding the criteria of region partitions. The watershed algorithm can divide two regions, basin and mountain regions, but it does not know how deep the basin area is. Similarly, in images, it does not know which one is an object or which one is background. Therefore, in the watershed algorithm, user markers represent the criterion. Users draw marks on each object and on the background, so the algorithm can decide and recognize the meeting point between the markers.

In this thesis, guidelines in image selection 2.2.2 and gray-scale images are inputs of watershed function in OpenCV in order to find the most matched border of a lesion. After the watershed function is complete, a black and white image is returned. Finding edges of a lesion in a black and white image can be easily extracted, so Suzuki and Abe's (1985) Border Following algorithm is used for the extraction.

Figure 10 – Original image with markers (left); image after watershed (center); Result image with a contour (right)

Pseudo code

1. The array of mouse start and end positions goes to OpenCV process in the image selection 2.2.
2. In OpenCV process, create a canvas for drawing lines.
3. Draw the lines on the canvas.
4. Find edges of the drawing lines with typical edge detection algorithm. In this thesis, cvFindcontour() is implemented.
5. Draw the edges on an empty canvas.
6. Use the watershed function with a grey image and edge canvas
7. Create an empty canvas and fill the canvas, which has 2 regions, which are divided by the watershed with black and white color.
8. The result of the black and white image shows outside and inside of an object if the two lines from users are located in outside and inside of the object.
9. Find edge of the result canvas with typical edge detection algorithm. In this thesis, cvFindcontour() is implemented.

2.3.2 Lesion Contour Selection

In the previous project, the contour outline is picked by users' selections among several similar contour lines. However, in this project, only one contour outline is drawn in a random color on the screen with the source image and user guidelines.

The accuracy of contour detection is improved. In the previous project, the number of successful detection is 7, and the number of unsuccessful detection is 43 among 50 benign and malignant lesion pictures. This successful rate is 14%. The reason of 14% is that the default threshold is used in the previous system. On the other hand, in this project, the contour detection rate is higher than the previous result using same picture from the previous system. The number of successful detection is 49, and the number of unsuccessful detections is 1 among 50 benign and malignant lesion pictures. This success rate is 98%.

2.4 Feature Extraction

The overall objective of feature extraction is to derive various features from a lesion image and camera on a mobile device according to "ABCD" rule suggestions. Asymmetry, border, and color are implemented in the previous project, so in this project, dimension is the objective of feature extraction. Therefore, this section includes real size estimation, reference method and camera distance method.

2.4.1 Size features

Size is a significant feature to determine melanoma according to the ABCD rule. However, measuring size of a lesion from an image is difficult because an image has limited features such as pixels, values of pixel changes, and color. Therefore, to get a real size, users provide evidence of size with the image or the system gets evidence of size from equipment of mobile devices such as camera, GPS, and gyroscope. In this thesis, there are two good methods, using a quarter as a reference and camera focus values, to classify lesions into benign and malignant lesions.

Reference Method

The reference method in this project is pixel comparison between a lesion and a quarter as a reference. Conceptually, pixels of an image are bigger when the distance between a camera and a target object is closer. Reversely, pixels of an image are smaller when distance between a camera and a target object is farther. There is no linear relation between distance between a camera and a target object and pixel size, but pixels in an image are related each other. If there are two objects, one is a reference, and the other is an object that is the target object in an image. Size of the target object can be estimated based on the real size of the reference object using the relationship between pixels of the reference and real size of the reference. In this thesis, the following formula is generated with a quarter as a reference:

*Real size lesion = Pixels of lesion * Real size quarter / Pixels of quarter. ……(2)*

Pseudo code of the reference method

1. Once contour finding for a reference is completed, users select the "chooseRef'" button.
2. The data of the reference such as the number of pixels and diameter is stored in an xml fie, outputRef.xml.
3. Open the file when a mole is selected by the find contour routine.

4. With reference data and a mole data, calculate by equation (2).
5. Display results

Camera Focus Distance Method

The other method that is used in this project is the Camera Focus Distance Method. This method is more convenient than the reference method. When the system refers a quarter to estimate real size, a lesion has to be taken with a coin as a reference. On the other hand, the camera focus distance method only needs a mobile device. When an image is captured, an android mobile phone can retrieve it from its camera, which is a new function from android version 2.3.

Camera focus values are three parts: near, optimal, and far values. These three values are camera properties from Depth of Field (DOF) to focus on a target object. Near focus distance value is the shortest distance value and the minimum distance of remaining a sharpened target, optimal focus distance value is a distance between near and far values and the best distance of a sharpened target, and far focus distance value is the maximum distance of remaining a sharpened target.

These three values are changed according to the distance between a camera and a target. When the target is farther and farther from the camera, the three values are increased, otherwise, when the target is closer and closer to the camera, the values are decreased. The amount of changes of the three values is regularly increased and decreased.

However, the camera focus distance values are not accurate on mobile devices because the range of the values is too wide to focus well at a certain distance. Therefore, the values need to be corrected. Figure 11 shows relations between real distance and average optimal camera focus distance values. In the graph, a trend line is drawn for approximating the values.

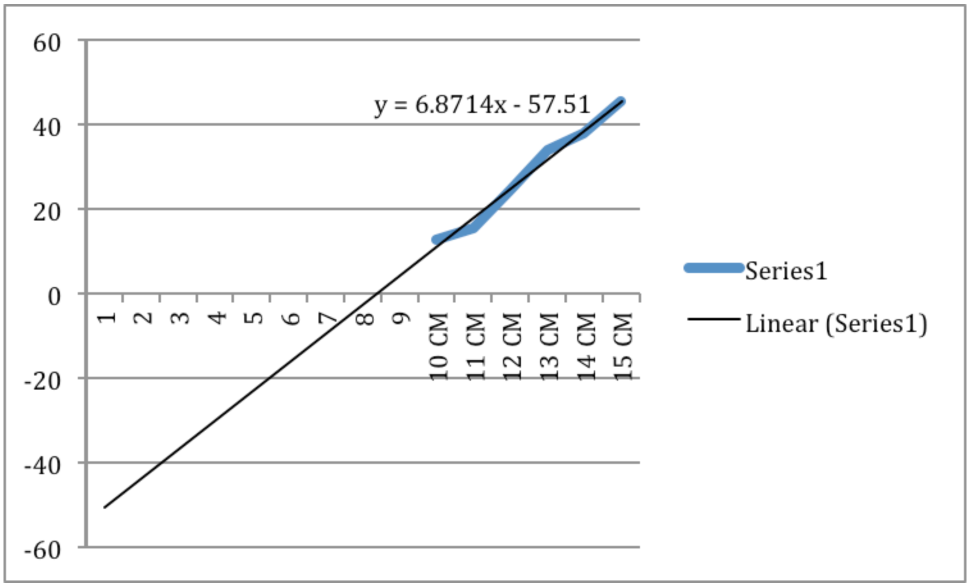

Figure 11 – A graph of real distance according to optimal camera focus distance values

From the trend line on the figure 11, a formula is generated blow:

$$RD = (OV + 57.51) / 6.8714 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (3)$$

RD is a calculated real distance from a camera to a target; OV is an optimal value of camera focus distance values. Because the camera values are not exactly equal to real distance, there is correction in equation (3)

After correcting the real distance, the system needs to estimate the pixel size according to the real size. For the correction, a well-known size, a quarter size, is used for estimating the real size of a pixel. Figure 12 shows the number of pixels of quarter size area in 10 to 15cm.

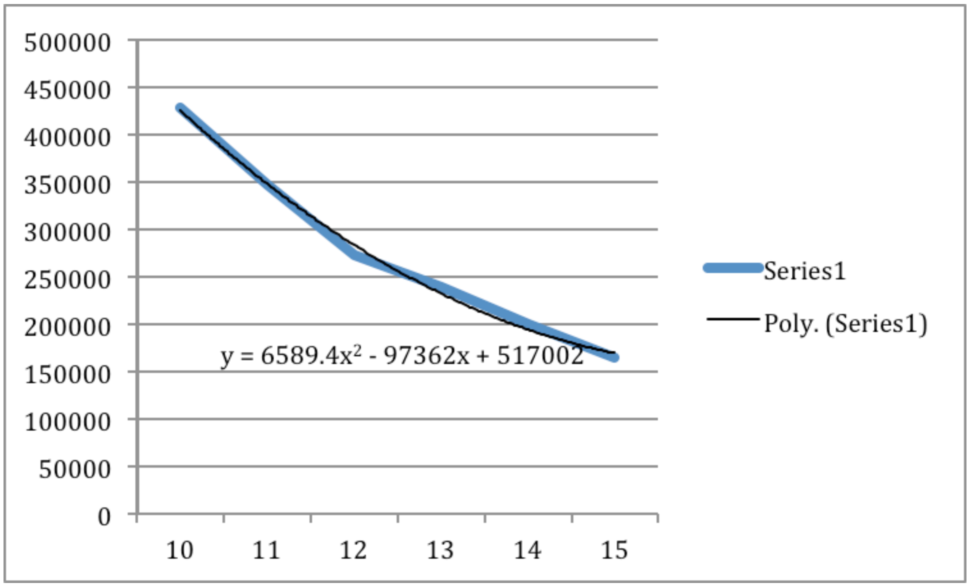

Figure 12 – A graph of a quarter area according to distance

From the trend line in figure 12, follow equation is made:

$$Y = 6589.4x^2 - 97362x + 517002 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(4)$$

X is a real distance and Y is the number of pixels of an area of a quarter coin, according to X. This equation can be used within 15cm distance between a camera and a target. Camera focus distance values are quite accurate in a fancy camera like DSLR, but they are not accurate enough in a mobile device, so at a long distance between a camera and a target, the accuracy of the measuring distance between two objects is not good enough to estimate the distance and size.

Relations between distance from a camera to a target and pixels of an area of a quarter are known by the formula (4). Once the relations are able to calculate and real distance of the target is obtained from the formula (3), the following equation is used for getting real size area of a target from ratio calculation of pixels of a target area and pixels of a quarter area:

$$RA = PA\ (QA\ /(6589.4RD^2 - 97362RD+517002))\ \ldots\ldots\ldots\ldots\ldots\ (5)$$

RA is the real size area of a target, PA is the number of pixels of a target area, QA is a real size area of a quarter, and RD is a calculated real distance value from the equation (3). The reason for using an optimal value instead of using other values is that optimal values indicate the right target position. In fact, the values are not accurate enough to indicate positions, so they need to be adjusted by the equation (3).

Pseudo code of the camera focus distance method

1. Once an image is captured with camera properties, the data goes to the OpenCV process using an xml file.
2. When finding a contour of a mole is completed, the number of pixels of the mole area goes to OpenCV.
3. With the camera properties and the number of pixels of the mole, calculate equations 3, 4, and 5.
    a. The optimal camera focus value is corrected by the equation 3.
    b. Real size area is from the equation 5 with the result from the equation 3.
4. Display the real size area value.

2.5 Other Implementation Details

Hardware

This project is developed and examined on a Nexus one smartphone with Google Android 2.3 operating system. The detail features that are used in this project are 1Ghz ARM processor, 512MB RAM and a 5 megapixels camera. Also, a 4GB micro-SD card is used for storage purposes.

Software

Based on the Android system, Android OpenCV is able to run. The Android OpenCV is developed to implement OpenCV functions that are written in C/C++ to the Android system. There are many improvements of using OpenCV in the Android system, so now OpenCV 2.3.1 can be implemented in the system mentioned by Kirill Kornyakov [11]. The difference between OpenCV usage in this project and in the previous project is that Simplified Wrapper and Interface generator (SWIG) is not used anymore. Instead of using that, Java Native Interface (JNI) is written manually. After the OpenCV version is changed to 2.3.1, it is easier to set environments and use Opencv in android than in previous versions. Also, more than 700 functionalities or methods of OpenCV are added, and some bugs are fixed.

For storage purposes, XML formats are used for passing camera properties from the camera activity to the OpenCV. When an image is captured, an XML file for camera properties is generated in the SD-card. Appendix A provides more detailed information about an XML file of camera properties.

3. EXPERIMENTAL RESULTS

3.1 Image Segmentation

In this thesis, the first step of the system is finding a contour of a lesion. Extracting contours are influenced by conditions such as different lighting, resolutions, focusing, camera, and distance from a lesion to a camera. To get features from those conditions, in the previous project, Otsu threshold method [12] was implemented and in this thesis, watershed method is implemented. In this section, the details of comparing two methods' performances will be mentioned. To compare the two methods, pictures ([13] to [28]) that are from the previous project are used for an experiment that compares the contours by the watershed method and the Otsu threshold method [12].

	Successful	Unsuccessful
Previous project (Otsu's algorithm [12])	14% (7/50)	86% (43/50)
Current project (Watershed algorithm [10])	98% (49/50)	2% (1/50)

Table 1 – Successful and unsuccessful rates of contour detection among 50 pictures

Table 1 shows the successful and unsuccessful rates of contour detection among 50 pictures. The pictures used in this experiment are from the previous project. The threshold of Otsu's algorithm [12] is set up automatically, and when the method is used, the results of contour finding are 7 successful results and 43 unsuccessful results out of 50 pictures. When the other algorithm is used, the results of contour finding are 49 successful results and 1 unsuccessful result out of 50 pictures. These 14% successful rates and 86% unsuccessful rates are very different from the results of contour finding by watershed algorithm, which were 98% successful rate and 2% unsuccessful rate.

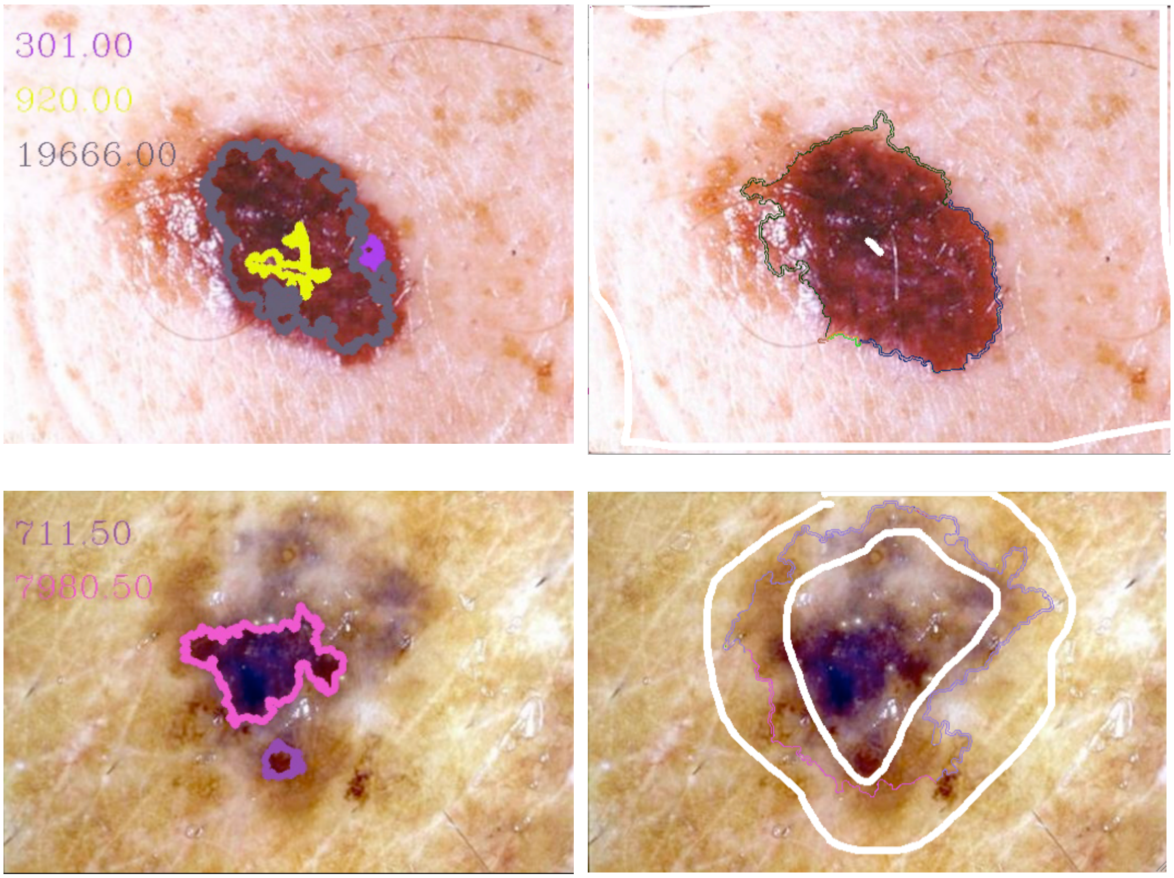

Figure 13 - Pictures of left column are malignant cases detecting by the Otsu threshold method [12]. Pictures of right column are malignant cases detecting by the watershed method [10]

Figure 14 - Pictures of left column are benign cases detecting by the Otsu threshold method [12]. Pictures of right column are benign cases detecting by the watershed method [10]

Figures 13 and 14 show detecting contours in benign and malignant pictures in two ways: Otsu threshold method [12] and watershed method [10]. Results of the right column have more perfect contour findings than results of the left column.

According to this experiment, the accuracy of detecting contours by the watershed method is much higher than the Otsu threshold method [12]. Moreover, the watershed method can find the best matching contour by drawing two lines rather than finding several contours like the left column pictures.

3.2 Size Estimation

To get the real size of the lesions, both the reference method and the camera focus distance method are used. There are two experiments that are conducted to verify the accuracy of these two methods.

3.2.1 Reference Method

In the reference method, a reference, the actual size of a quarter, is used. In this experiment, the error rate of the size estimation using the diameter and the number of pixels are measured at a distance within a range of 3~10 inches between a camera and a target, and the real area is calculated by ratio calculation in equation (2).

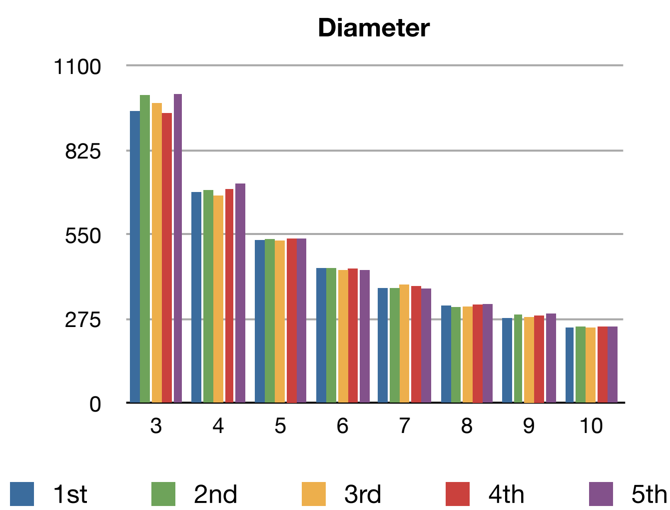

Figure 15 – the number of pixels of lesion diameters between 3 to 10 inches after 5 experiments

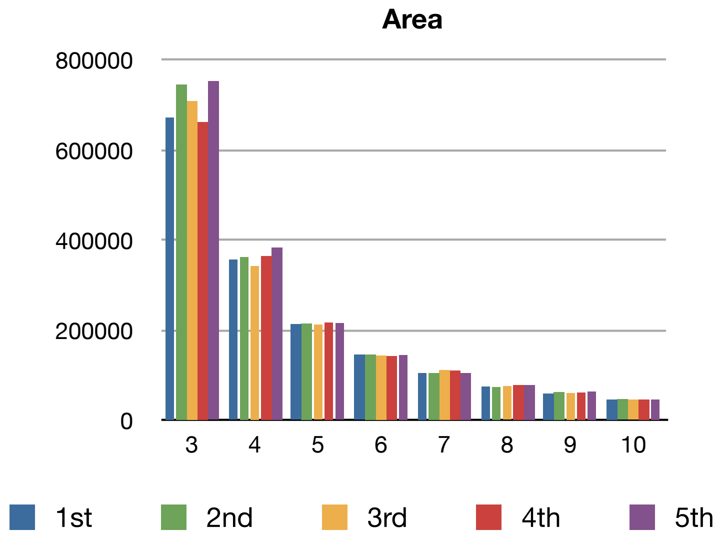

Figure 16 – the number of pixels of lesion area between 3 to 10 inches in 5 experiments

In both figures 15 and 16, the pixels of lesion diameters and area show the relationship between distance from a camera and a target and pixels of a lesion. These graphs decrease exponentially and at each distance, results from all 5 experiments' measurements yielded similar values. Therefore, at each distance, a pixel represents a certain area, so the ratio calculation between a reference and a lesion is used for estimating the real size of the lesion.

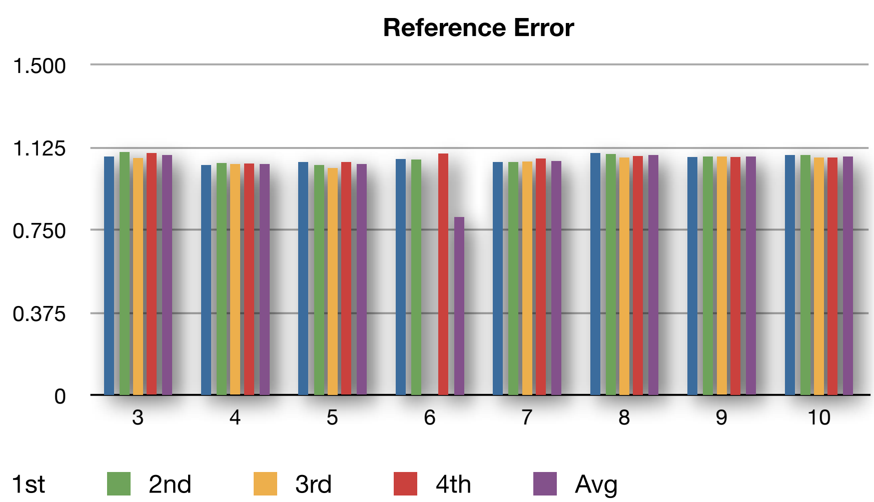

Figure 17 – error rates of reference method after 4 experiments

The above graph shows the error of the reference method after 4 experiments. All of the numbers are below 1.125. The average error rate is 1.04% in this experiment.

3.2.1 Camera Focus Distance Method

This method is more convenient than the reference method, but it has a higher average error rate of 8.23% and a maximum error rate of 24.03% in this experiment. This error rate can be affected by camera focus values and the degree of the tilt of the camera. Also, it can be reduced by using more accurate data than this experiment data. In this experiment, the accuracy of the size estimation is shown by the pixel results from the calculation using equation (5).

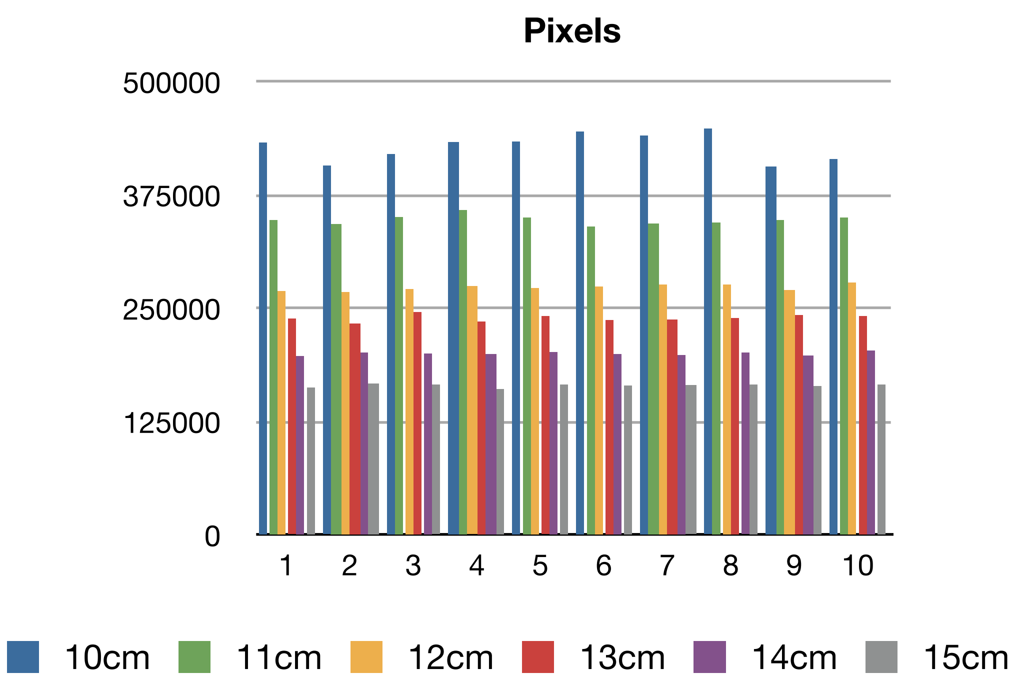

Figure 18 – the number of pixels on a quarter according to distances ranging from 10-15 cm in 10 experiments

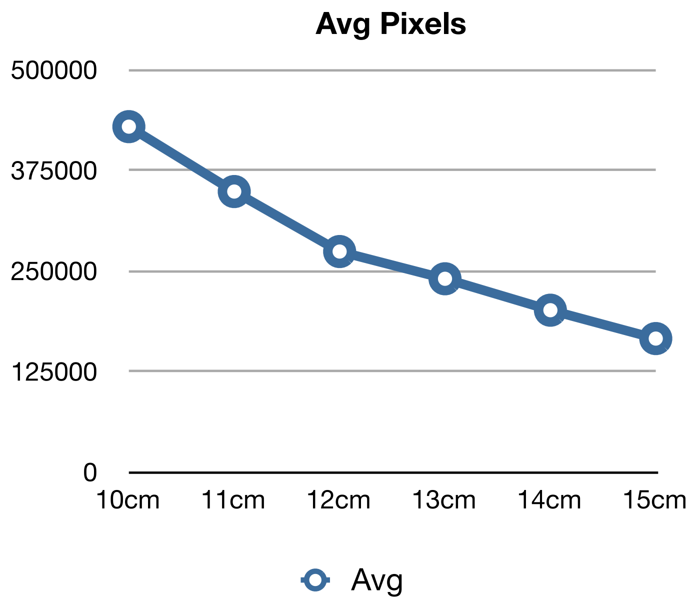

Figure 19 – Average of object area pixels according to distances ranging from 10-15cm

First of all, the number of pixels on a quarter is collected as data. The details of the data are shown in figure 18 and 19. It shows the number of pixels on a quarter on measurements taken every 1 cm between 10 to 15cm. At a further distance away, there were fewer pixels and at a shorter distance away there were more pixels.

Camera focus values, which are real distance values, are also affected by distance shown in figure 20. This change is not linear and is irregular, so equation (3) can be used to change from the camera values to real distance.

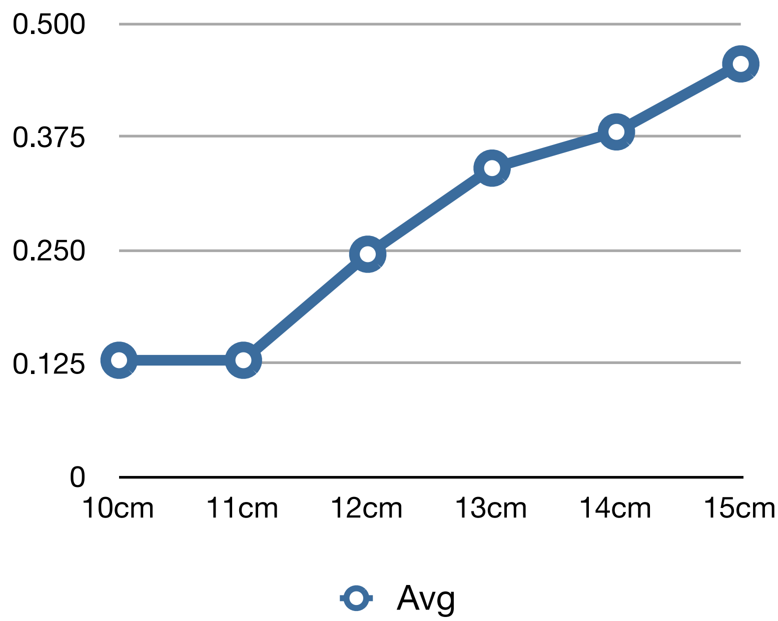

Figure 20 – Average of optimal camera focus distance values (10-15cm)

With camera focus distance values and pixel changing according to different distances, an estimated real size area of a lesion can be calculated by using equation (5). The result is shown in figure 21.

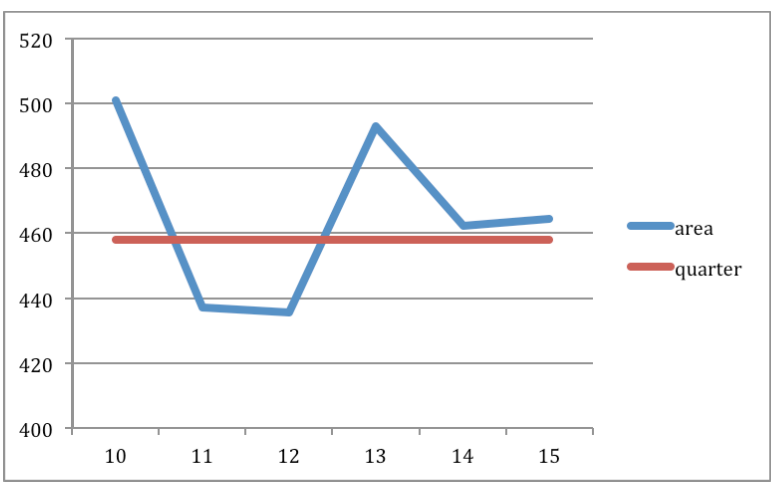

Figure 21 – graph of average size estimation area results with real size quarter area

The estimated values in figure 21 are around its real size value of a quarter's area. Following figure 22 shows differences between size estimation area and real area.

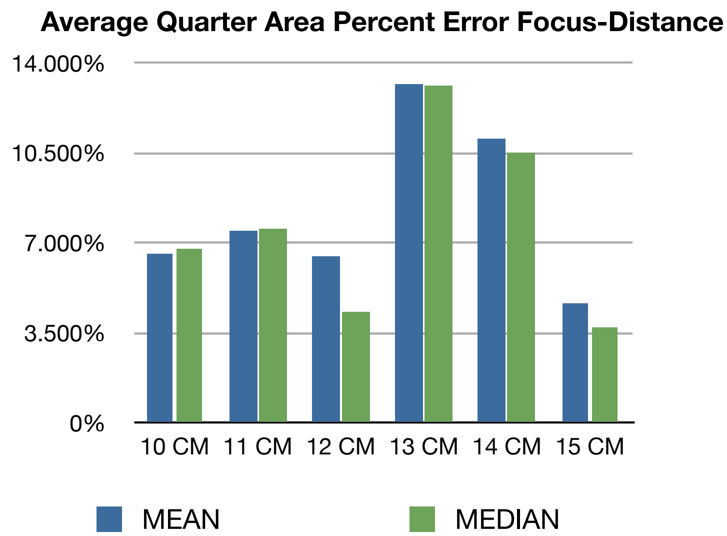

Figure 22 – Average Quarter Area Percent Error Focus-Distance

4. CONCLUSION AND FUTURE WORK

This thesis demonstrates that size can be estimated with a reference and camera focus distance values and that image segmentation can be improved on a smart mobile device. When comparing the results of the previous project with the current thesis, the success rate of finding the contour is increased by 84% from 14% to 98% (table 1). Also only one contour is found with the current thesis instead of several contours with the previous project. Moreover, users can draw the two guidelines to find contours instead of putting in a threshold number. This result is able to detect more lesions, which were not detected before. When estimating the size of a lesion, the reference method and camera focus distance method can be used. In fact, the reference method is more accurate than the camera focus distance method, but with the camera focus distance method no reference is needed. Even though it has a higher error rate than the reference method, it can be improved in the future.

In the future, collecting more and improved lesion data will be useful for classification of melanoma because more training data will result in more accurate classifications. Also, this thesis needs to be tested in cameras in various smartphone devices since each camera may differ in performance and have different property values. In the segmentation, in this thesis, the user needs to draw two guidelines. However, in the future, there will be a simpler method where if users touch a lesion area, the system will automatically draw a line around the legion.

5. APPENDIX A – XML FILES

This section describes the camera property XML. The XML file stores camera focus values and other properties and passes the values to the OpenCV process. This file is saved in a directory called dataCollection on the device’s SD-card. When an image is captured, the file is generated with the patient’s ID, date, and time.

Camera property XML


```xml
<?xml version="1.0"?>
<opencv_storage>
<Time>"Tue, Nov 01, 2011 20:10"</Time>
<imageFile>"0_2011110120100800529.jpg"</imageFile>
<ColorEffect>none</ColorEffect>
<FlashMode>off</FlashMode>
<FocusMode>auto</FocusMode>
<WhiteBalance>auto</WhiteBalance>
<patient></patient>
<lesionClassification>Unknown</lesionClassification>
<meterDistanceNear>0.122</meterDistanceNear>
<meterDistanceOptimal>0.13</meterDistanceOptimal>
<meterDistanceFar>0.139</meterDistanceFar>
<centimeterDistanceNear>12.2</centimeterDistanceNear>
<centimeterDistanceOptimal>13.0</centimeterDistanceOptimal>
```

<centimeterDistanceFar>13.9</centimeterDistanceFar>

</opencv_storage>

When a contour is found with the reference method or the camera focus distance method, the system needs to know information of the reference such as the number of pixels, diameter of a lesion, or a path of a camera property xml file. For storing these values, a XML file, named outputRef.xml, is generated.

outputRef.xml

<?xml version="1.0"?>

<opencv_storage>

<refArea>73217.</refArea>

<Defects>0.</Defects>

<refDiameter>309.</refDiameter>

<xmlPath>/mnt/sdcard/dataCollection/0_2011102815085400828.xml</xmlPath>

</opencv_storage>

When displaying the result, values that are related to the area estimation are added into output.xml, which has features to classify the lesions.

Output.xml

<?xml version="1.0"?>

<opencv_storage>

<Bmean>3.5105935776338889e+01</Bmean>

<Bstddev>1.0510356821379961e+01</Bstddev>

<Gmean>2.8748915036320170e+01</Gmean>

<Gstddev>9.0488703422714938e+00</Gstddev>

<Rmean>3.2660160752550915e+01</Rmean>

<Rstddev>9.9176613039048309e+00</Rstddev>

<BmeanLeft>3.5372828679129164e+01</BmeanLeft>

<BstddevLeft>1.0746280235248353e+01</BstddevLeft>

<GmeanLeft>2.8851742441726284e+01</GmeanLeft>

<GstddevLeft>9.0917737051891798e+00</GstddevLeft>

<RmeanLeft>3.3264251096238169e+01</RmeanLeft>

<RstddevLeft>9.9570942672536074e+00</RstddevLeft>

<BmeanRight>3.4835057802119145e+01</BmeanRight>

<BstddevRight>1.0256263105527232e+01</BstddevRight>

<GmeanRight>2.8643991847022761e+01</GmeanRight>

<GstddevRight>9.0010045247621555e+00</GstddevRight>

<RmeanRight>3.2048233262280029e+01</RmeanRight>

<RstddevRight>9.8365146618030561e+00</RstddevRight>

<BmeanDiff>5.3777087701001847e-01</BmeanDiff>

<BstddevDiff>4.9001712972112088e-01</BstddevDiff>

<GmeanDiff>2.0775059470352275e-01</GmeanDiff>

<GstddevDiff>9.0769180427024310e-02</GstddevDiff>

<RmeanDiff>1.2160178339581407e+00</RmeanDiff>

<RstddevDiff>1.2057960545055124e-01</RstddevDiff>

<IsoperimetricRatio>7.7501075753374782e-01</IsoperimetricRatio>

<HullContourRatio>8.8777796545660614e-01</HullContourRatio>

<SignificantDefects>0.</SignificantDefects>

<DefectTotal>56.</DefectTotal>

<DefectDepthSum>1.3204260504245758e+02</DefectDepthSum>

<HullLength>1.2821984863281250e+03</HullLength>

<ContourLength>1.4442783401012421e+03</ContourLength>

<Area>128647.</Area>

<AreaString>"128647.00"</AreaString>

<Class>1.</Class>

<File>/mnt/sdcard/dataCollection/0_20111127124221000099.jpg</File>

<Accuracy>3.</Accuracy>

<SigDefDepthSumRto>0.</SigDefDepthSumRto>

<SigDefWidthSum>0.</SigDefWidthSum>

<Diameter>411.</Diameter>

<xmlPath>/mnt/sdcard/dataCollection/0_20111127124221000099.xml</xmlPath>

<realCentimeterDistancesNear>1.8475471698113207e+01</realCentimeterDistancesNear>

<realCentimeterDistancesOptimal>2.0011933521553100e+01</realCentimeterDistancesOptimal
>

<realCentimeterDistancesFar>2.3258990336103860e+01</realCentimeterDistancesFar>

<xmlPath>/mnt/sdcard/dataCollection/0_20111127124221000099.xml</xmlPath>

<avgDistance>2.0011933521553100e+01</avgDistance>

<pixPerMil>5.2589509083296480e+02</pixPerMil>

<estArea>2.4462483533785439e+02</estArea>

</opencv_storage>

APPENDIX B – USER INTERFACE

This section shows the user interface of the application. The user interface has two main menus, an Android menu and a menu on the right side of the screen. In the Android menu, there are 6 menu items: take pic, choose pic, select, choose ref, reset drawing, and classify.

When take pic is clicked a camera app is opened. In the camera app, a patient's ID and type of classification of the lesion will shown be when data needs to be collected. There is also a capture button for saving a capture image.

When choose pic is clicked, a gallery is opened, so the program can load an image. After the program loads an image, the user can draw two guidelines inside an object and outside an object.

After the drawing is finished and select is clicked, the program shows a contour line on the screen. Now the users need to use the right side menu. The first line input is about selecting a contour, and the second input is a option to choose which size estimation method is used; 1 is the reference method, and 2 is the camera focus distance method.

If users choose the reference method and draw guidelines for a reference, the reference information is stored into a XML file after the user the clicks on choose ref.

If users choose the camera focus distance method and draw guidelines for a lesion or if users choose the reference method and are ready to save the reference information, the program is ready to classify. After users click the classify button, results are displayed on the screen with an image.

With the reset-drawing button, users can reset guidelines when they make mistakes drawing their guidelines.

APPENDIX C – DEVELOPMENT NOTES

The following prerequisites are required to set up the development environment for the project.

1	Android NDK 6	The Android Native Development Kit
2	Android-OpenCV	An Android port of OpenCV [11].

To compile an android project and OpenCV codes inside of the project, NDK and Android-OpenCV are needed. At the root of the project directory, a file, ndk-build, is executed and it will generate .a so file. After a so file is generated, in the eclipse, the project needs to be refreshed. Finally, running the project in Eclipse would deploy the project to the attached device.

REFERENCES

[1] American Cancer Society. (2011, 6 21). How many people get melanoma? Retrieved 11 4, 2011, from http://www.cancer.org/Cancer/SkinCancer-Melanoma/OverviewGuide/melanoma-skin-cancer-overview-key-statistics

[2] American Cancer Society. (2009). 2009 Cancer Facts and Figures. American Cancer Society.

[3] H. Ganster, A. Pinz et al. "Automated Melanoma Recognition", IEEE Transactions on Medical Imaging March 2001

[4] American Melanoma Foundation. (n.d.). ABCD's of Melanoma. Retrieved from AMF: http://www.melanomafoundation.org/prevention/abcd.htm

[5] Scan, S. (2011, 10 4). Skin Scan - The Most Accurate Skin Cancer Detector on iPhone. Retrieved 11 17, 2011, from Apple Store: http://itunes.apple.com/us/app/skin-scan-the-most-accurate/id434196122?mt=8

[6] Medical Image Mining Labortories. (2011, 8 9). Skin of Mine. Retrieved 11 17, 2011, from Apple Store: http://itunes.apple.com/us/app/skin-of-mine/id432628083?mt=8

[7] Smith, A. (2011, 7 11). a project of the PewReaserchCenter. Retrieved 11 21, 2011, from Smartphone Adoption and Usage: http://www.pewinternet.org/Reports/2011/Smartphones.aspx

[8] Ramlakhan, K. THE DESIGN AND IMPLEMENTATION OF A MOBILE, AUTOMATED SKIN LESION CLASSIFICATION SYSTEM. University of Missouri.

[9] Suzuki, A. (1985). Topological Structural Analysis of Digital Binary Images by Border Following. IEEE. 30, pp. 32-46. IEEE Computer Vision, Graphics & Image Proc.

[10] Meyer, F. Color image segmentation. International Conference on Image Processing and its Aplications, (pp. 303-306).

[11] Kornyakov, K. (2011, 9 21). OpenCVWiki. Retrieved 11 29, 2011, from Android: http://opencv.willowgarage.com/wiki/Android

[12] Nobuyuki Otsu (1979). "A threshold selection method from gray-level histograms". IEEE Trans. Systems., Man. and Cybernetics.

[13] Lesion Images . Retrieved April 2011, http://www.healthhype.com/wp-content/uploads/melanoma_skin_cancer_back.jpg

[14] Lesion Images . Retrieved April 2011, http://www.webmd.com/melanoma-skin-cancer/slideshow-precancerous-skin-lesions-and-skin-cancer

[15] Lesion Images . Retrieved April 2011, http://www.ohiohealth.com/mayo/images/image_popup/r7_skinc.jpg

[16] Lesion Images . Retrieved April 2011, http://www.dermatology.org/molemelanoma/watch2.html

[17] Lesion Images . Retrieved April 2011, http://www.robertsreview.com/cancer_pictures_melanoma.html

[18] Lesion Images . Retrieved April 2011, http://skincancer.about.com/od/skincancerbasics/ss/melanomaimages.htm

[19] Lesion Images . Retrieved April 2011, http://skincancer.about.com/od/symptoms/ss/mole_2.htm

[20] Lesion Images . Retrieved April 2011,
http://skincancer.about.com/od/symptoms/ss/mole_6.htm

[21] Lesion Images . Retrieved April 2011,
http://picturesskincancer.com/index.php?page=malignant-melanoma

[22] Lesion Images . Retrieved April 2011,
http://picturesskincancer.com/index.php?page=malignant-melanoma

[23] Lesion Images . Retrieved April 2011,
http://dermatology.about.com/od/melanoma/ss/ABCD_SBS_3.htm

[24] Lesion Images . Retrieved April 2011,
http://www.skinsight.com/adult/melanoma.htm

[25] Lesion Images . Retrieved April 2011,
http://visualsonline.cancer.gov/details.cfm?imageid=2360

[26] Lesion Images . Retrieved April 2011, http://www.medinik.com/skin/types-of-moles

[27] Lesion Images . Retrieved April 2011,
http://www.cancer.gov/cancertopics/wyntk/moles-and-dysplastic-nevi/page8

[28] Lesion Images . Retrieved April 2011,
http://dermatology.about.com/od/melanoma/ss/ABCD_SBS.htm

[29] Lesion Images . Retrieved April 2011, http://melanomapictures.org/melanoma-pictures/melanoma-library