



STA-3900

MASTER'S THESIS IN STATISTICS

Unsupervised Segmentation of Skin Lesions

Kajsa Møllersen

November, 2008

FACULTY OF SCIENCE
Department of Mathematics and Statistics
University of Tromsø

STA-3900

MASTER'S THESIS IN STATISTICS

Unsupervised Segmentation of Skin Lesions

Kajsa Møllersen

November, 2008

Acknowledgements

First and foremost I would like to thank my supervisor Professor Fred Godtlielsen for always asking the right questions and pushing me forward step by step.

Thanks to the Norwegian Center for Telemedicine and the researchers at Tromsø Telemedicine Laboratory. Special thanks to Thomas G. Schopf (MD), Tor Arne Øigård, Kevin Thon and Kristian Hindberg for sharing their knowledge and giving me valuable feedback.

Thanks to my fellow students at the University of Tromsø, Department of Mathematics and Statistics. Special thanks to Jørgen Skancke for his catching enthusiasm.

Thanks to the Barents Institute in Kirkenes for providing me with a place to work during my stays in my hometown.

Last, but not least, I would like to thank Hilja Lisa Huru for inspiration, as a friend and mathematician, during all my years as a student at the University of Tromsø.

Abstract

During the last decades, the incidence rate of cutaneous malignant melanoma, a type of skin cancer developing from melanocytic skin lesions, has risen to alarmingly high levels. As there is no effective treatment for advanced melanoma, recognizing the lesion at an early stage is crucial for successful treatment. A trained expert dermatologist has an accuracy of around 75 % when diagnosing melanoma, for a general physician the number is much lower. Dermoscopy (dermatoscopy, epiluminescence microscopy (ELM)) has a positive effect the accuracy rate, but only when used by trained personnel. The dermoscope is a device consisting of a magnifying glass and polarized light, making the upper layer of the skin translucent.

The need for computer-aided diagnosis of skin lesions is obvious and urgent. With both digital compact cameras and pocket dermoscopes that meet the technical demands for precise capture of colors and patterns in the upper skin layers, the challenge is to develop fast, precise and robust algorithms for the diagnosis of skin lesions. Any unsupervised diagnosis of skin lesions would necessarily start with unsupervised segmentation of lesion and skin.

This master's thesis proposes an algorithm for unsupervised skin-lesion segmentation and the necessary pre-processing. Starting with a digital dermoscopic image of a lesion surrounded by healthy skin, the pre-processing steps are noise filtering, illumination correction and removal of artifacts. A median filter is used for noise removal, because of its edge-preserving capabilities and computer efficiency. When the dermoscope is put in contact with the patient's skin, the angle between the skin and the magnifying glass impacts on the distribution of the light emitted from the diodes attached to the dermoscope. Scalar multiplication with an illumination correction matrix, individually adapted to each image, facilitates the analysis of the image, especially for skin lesions of light color. Artifacts such as scales printed on the glass of the dermoscope, hairs and felt-pen marks on the patient's skin are all obstacles for correct segmentation. This thesis proposes a new, robust and computer effective algorithm for hair removal, based on morphological operations of binary images.

The segmentation algorithm is based on global thresholding and histogram analysis. Unlike most segmentation algorithms based on histogram analysis, the algorithm proposed in this thesis makes no assumptions on the

characterization of the lesion mode. From the truecolor RGB image, the first principal component is used as grayscale image. The algorithm searches for the peak of the skin mode, and the skin mode's left bound. The pixel values belonging to the bins to the left of the bound, are regarded as samples from an underlying distribution and the expected value of this distribution is estimated. The value of the pixels in the bin located at equal distance from the expected value of the lesion mode, and the skin-mode peak is used as threshold value. After global thresholding, post-processing is applied to identify the lesion object. The only parameters in this algorithm are the number of bins in the histogram and the shape of the local minimum regarded as skin-mode bound.

The dermoscopic images have been divided into two independent sets; training set and test set. The training set consists of 68 images, and the test set consists of 156 images. 80 of the images from the test set have been evaluated by expert dermatologists by visual inspection.

Contents

1	Introduction	1
1.1	Skin Cancer	1
1.2	Dermoscopy	3
1.3	The Camera and the Image Format	3
1.4	The Dermoscope	4
1.5	The Skin Lesions	4
2	Mathematics, Statistics and Image Processing	7
2.1	Polynomial Interpolation	7
2.2	Matrix Algebra	8
2.3	Statistical Analysis	10
2.3.1	Univariate Statistics	10
2.3.2	Multivariate Statistics	12
2.4	Filtering	14
2.4.1	Noise Removing Filters	14
2.5	Presentation of Images	16
2.5.1	Presentation of Grayscale Images	16
2.6	Gray Conversion	17
2.6.1	RGB-to-luminance	18
2.6.2	Principal Component Transform	18
2.6.3	Application	21
2.7	Binary Images	23
2.7.1	Otsu's Method	24
2.8	Morphological Operations	28
2.8.1	Dilation and Erosion	29
2.8.2	Opening and Closing	29
2.9	Mapping	31

2.9.1	Rotation	31
2.10	Connectivity	32
3	Pre-processing	35
3.1	Masking	35
3.2	Filtering	37
3.3	Color Representation	37
3.3.1	Gray-conversion	38
3.4	Illumination	38
3.4.1	Centering the Illuminated Disk	39
3.4.2	Center of Illumination	40
3.4.3	Illumination Correction Matrix	41
4	Removal of Artifacts	49
4.1	Scales	51
4.2	Hairs	55
4.2.1	Improvement of the Hair-Removing Algorithm	64
5	Segmentation of Skin Lesions	67
5.1	Color Space and Gray Conversion	68
5.2	Threshold Segmentation	68
5.2.1	Histogram Analysis	69
5.3	Segmentation Algorithm	73
5.3.1	Skin Mode	73
5.3.2	Lesion Mode	77
5.3.3	Unimodal histograms	78
5.3.4	Global Thresholding	78
5.4	Separation of Lesion and Skin	81
5.5	Drawing of Borders	85
6	Evaluation of Segmentation Algorithms	91
6.1	Comparison With Other Algorithms	92
7	Discussion	95
7.1	Hair Removal	95
7.1.1	Comparison With Other Algorithms	96
7.2	Segmentation Algorithm	96
7.2.1	Comparison with other algorithms	97

CONTENTS

ix

7.2.2 Challenges	98
7.3 Expert Evaluation of Segmentation Algorithm	98
7.4 Further Work	99
Appendix A	101
Appendix B	127
Bibliography	131

List of Figures

1.1	Age-adjusted incidence rates per 100 000 person-years.	2
2.1	The same 2000×2000 pixel grayscale image in (a) gray presentation (b) color presentation.	17
2.2	Three-layer color image.	21
2.3	Principal components. (a) First. (b) Second. (c) Third.	22
2.4	Proportion of total variance explained by the three principal components, respectively.	22
2.5	Grayscale image based on (a) principal component transform (b) RGB-to-luminance transform.	23
2.6	(a) Original binary image. (b) After dilation, new pixels of value 1 colored gray. (c) After dilation, all pixels of value 1 colored white. (d) After erosion.	30
2.7	Illustrating connectivity.	33
3.1	Illuminated center and near-black surround (2400×3276 pixels).	36
3.2	The image with a superimposed black mask.	36
3.3	(a) Off-centered illuminated disk in a 2400×2400 pixel square. (b) The new 2000×2000 matrix with centered illuminated disk.	39
3.4	Binary image with the 0.75-quantile as threshold (a) surrounding the lesion (b) not surrounding the lesion (2000×2000 pixels).	40
3.5	Small skin lesion surrounded by skin with homogeneous color (2400×2400 pixels).	42
3.6	Grayscale image based on (a) RGB-to-luminance and (b) PCT (2000×2000 pixels).	42

3.7	Mean illumination as a function of the distance from the center of the image.	43
3.8	Smoothed graph (blue) and interpolated graph(red).	44
3.9	The general correction matrix (2000×2000 pixels).	44
3.10	The correction matrix for the image in Figure 3.11 (1700×1700 pixels).	45
3.11	The image (a) before and (b) after scalar multiplication with correction matrix.	46
3.12	The image (a) before and (b) after scalar multiplication with a possibly incorrect correction matrix.	46
3.13	Segmentation (a) without and (b) with illumination-correction.	47
4.1	(a) Air bubble with distinct borders. (b) The borders detected by the hair-removing algorithm. (c) The borders replaced by skin-colored pixel values.	50
4.2	(a) The original image. (b) The binary image. (c) Detected scale objects, after dilation. (d) Final image with scales removed.	54
4.3	The red layer of the image.	57
4.4	Binary images with threshold value equal to (a) 0.85 times Otsu's threshold, (b) 0.95 times Otsu's threshold, and (c) 1.05 times Otsu's threshold.	58
4.5	(a) The image after closing. (b) The difference between Fig. 4.4(a) and Fig. 4.5(a). (c) After opening.	59
4.6	Pixels possibly belonging to hairs in (a) vertical direction, and (b) horizontal direction.	59
4.7	Pixels possibly belonging to thick hairs in (a) vertical direction, and (b) horizontal direction.	60
4.8	Pixels considered to belong to (a) hairs in the vertical direction, (b) hairs in the horizontal direction, (c) thick hairs in the vertical direction, and (d) thick hairs in the horizontal direction (no hairs).	61
4.9	Dilated hair objects in (a) vertical direction, and (b) horizontal direction.	62
4.10	All the hair pixels.	63
4.11	Final image with hairs and scales removed.	64

5.1	(a) Lesion and skin clearly separable by the human eye. (b) The mode representing the lesion is easily detected.	70
5.2	(a) Lesion and skin clearly separable by the human eye. (b) The mode representing the lesion is not easily detected.	71
5.3	The left half of Figure 5.2(b).	71
5.4	Images challenging for histogram analysis. (a)-(b) The global maximum and the bound of the skin mode is no more than 19 bins apart. (c)-(d) The skin mode has a double peak, where the peak to the right is the global maximum. (e)-(f) Multiple lesion modes.	75
5.5	Images challenging for histogram analysis. (a)-(b) The lesion mode peak is nearly as high as the skin mode peak. (c)-(d) Unimodal histogram. (e)-(f) No obvious lesion mode.	76
5.6	Lesion with honeycomb-like network.	79
5.7	The first principal component of the image in Figure 5.6 (a) before and (b) after opening. Segmentation (c) without and (d) with opening.	80
5.8	The binary image of Figure 5.4(a) (a) before and (b) after the objects in connection with the mask have been removed.	82
5.9	The binary image of Figure 5.6 (a) before and (b) after filling the holes.	82
5.10	(a) The image on which the segmentation is based. (b) The binary image before and (c) after the opening.	83
5.11	(a) The image on which the segmentation is based. (b) After the post-processing. (c) The two largest objects.	84
5.12	(a) The binary image after post-processing. (b) After filtration. (c) The absolute difference. (d) The final borders.	86
5.13	Original image with borders superimposed.	87
5.14	Images with borders superimposed, based on segmentation of (a) Figure 5.1(a), (b) Figure 5.2(a), (c) Figure 5.4(a), (d) Figure 5.4(c), (e) Figure 5.4(e) and (f) Figure 5.5(a).	88
5.15	Images with borders superimposed, based on segmentation of (a) Figure 5.5(c), (b) Figure 5.5(e), (c) Figure 5.6, (d) Figure 5.10(a) and (e) Figure 5.11(a).	89

Chapter 1

Introduction

This master's thesis tries to solve a problem many have tried to solve before, some with greater success than others. The problem to be solved is segmentation or border detection. This is in fact two different tasks, closely tied together, and often not distinguished from one another in the literature. Both segmentation and border detection have the same goal, namely to separate one object or region from the background. This thesis handles specifically the segmentation of skin lesions from the surrounding skin.

1.1 Skin Cancer

Skin cancer is one of the most common and one of the deadliest type of cancer among the fair-skinned population in Europe, North America and Australia. In Norway, skin cancer was among the top three new cases of cancer registered in the period 2002-2006. In 2006, 2517 persons got the diagnosis, 293 people died from the disease in 2004 (CRN, 2006).

The incidence rate of skin cancer has been growing ever since the Cancer Registry of Norway started its data collection in 1952. Among the different types of skin cancer, melanoma of the skin (from now on referred to only as melanoma) is the deadliest of them all. In 2006, less than 50% (1178) of the total registered new cases of skin cancer was melanoma, but more than 90% (266) of the total deaths caused by skin cancer in 2004 was due to melanoma. As seen in Figure 1.1, the number of new cases of melanoma is stabilizing on a high level. The incidence rate shows number of new cases per 100,000 inhabitant, standardized according to the world population (for

details, see CRN (2006)).

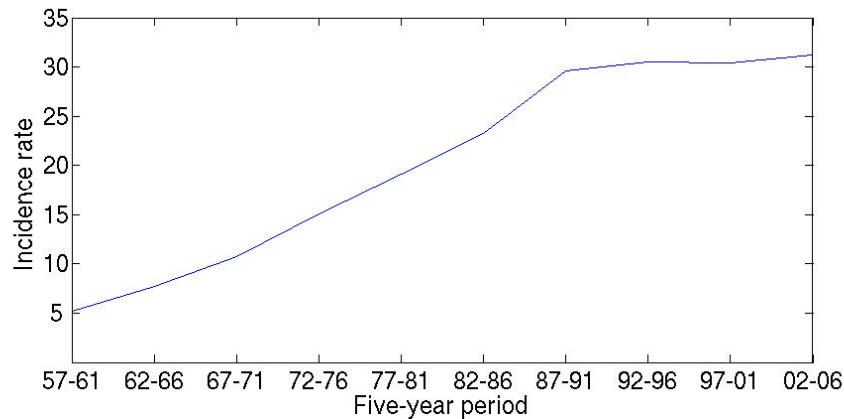


Figure 1.1: Age-adjusted incidence rates per 100 000 person-years.

Melanoma is the fifth and sixth most common type of cancer among females and males, respectively, for the years 2002-2006. In the age group 30-54 years, melanoma is the second most common type of cancer for both genders in the same time period.

Melanoma starts in a melanocytic lesion in the upper layers of the skin (dermis and epidermis). Melanocytic lesions form the largest group of pigmented skin lesions. If the melanoma is left alone, and the body itself is unable to fight it, it moves on to the lower layers of the skin, to vessels, and finally to other organs. There is today no effective treatment against advanced melanoma, and the only way to treat it is to physically remove it in its early stage. Melanoma *in situ* are the melanomas still situated only in the upper skin layers. Removing the melanoma at this stage is an easy task, it is cost effective and the patient hardly gets any bother from it.

To discover melanoma in its early stage is the difficult part. Even discriminating between melanocytic and non-melanocytic lesions is not straightforward (Stolz et al., 2002, p.42). Dermatologists have an accuracy of around 75% when diagnosing melanoma. For general physicians the number is much lower (Stolz et al., 2002, p.1). The clinical ABCD rule for diagnosing skin lesions is widely used. The clinical ABCD rule states that the skin lesion is likely to be a melanoma if the following criteria are fulfilled: Asymmetry, Border irregularity, Color variation and Diameter of more than 6 mm.

Unfortunately, the criteria are not unambiguously defined, and the result is highly subjective (Stolz et al., 2002, p.1).

In the 1990's a new device, the dermoscope, was introduced. The use of the dermoscope makes diagnosing the skin lesion more successful for trained dermatologists. Unfortunately, giving a dermoscope to untrained personnel can lead to even lower percentage of correct diagnosis. A computer-aided diagnostic system is therefore most needed, and would be of special importance among general physicians that do not have the training and experience of a dermatologist.

1.2 Dermoscopy

Dermoscopy, dermatoscopy or epiluminescence microscopy (ELM), is a non-invasive technique for diagnosing skin lesions. A dermoscope, or dermatoscope, is a combination of a magnifying glass (lens) and polarized light. The lens is encircled by light sources and is put in direct contact with the patients skin. Fluid (usually water or oil) is used to refract the light. Without the polarized light and the fluid, light is reflected at the surface of the skin, and only the patterns visible there can be examined by the physician. When the light is refracted by the polarized light and the fluid, it is the reflections from the subsurface that reaches the physicians eye. This way, the physician is able to extract much more information from a simple non-invasive examination.

1.3 The Camera and the Image Format

There are thousands of different ways to record the waves of light reflecting from an object, with different wavelengths and different angles. This thesis utilizes one of them.

The skin lesions are photographed with a Ricoh GR digital camera. The images are recorded with the size 2460×3276 pixels, which gives a spatial resolution of 0.011×0.011 mm per pixel. The depth is 8 bit per pixel. The images are stored in the digital RAW format DNG, and then converted to the uncompressed format ppm (Portable Pixel Map). Converting to the often used format jpg gave poorer results. Not surprisingly, since jpg is a compressed format and therefore some of the information is lost. When

converting from DNG to ppm the gamma-correction was not used. When an image has been gamma-corrected, it is easier for the human eye to see the details, but for the performance of the segmentation algorithm, uncorrected pixel values proved superior. The DNG-to-ppm conversion is a nonlinear transformation, and the pixel values were stored as 8 bit integers with range from 0 to 255 or 16 bit integers with range from 0 to 65,535.

The color space used is the RGB (red, green, blue) color space. The three perpendicular axes represents the intensities of the three primary colors. The white point used for conversion is D65.

1.4 The Dermoscope

A DermLite Pro II HR dermoscope is attached in front of the lens of the camera. The dermoscope has a magnifying lens of $\times 10$, and 32 light-emitting diodes (LED) of polarized light with the output of approximately 18 000 lux (1800 foot-candles) (DermLite). The dermoscope, with the camera behind, is placed on the patients skin, and water is used as fluid. One or more photos are taken. Because of the polarized light and the fluid, the light penetrates the uppermost skin layer.

1.5 The Skin Lesions

A set of 234 digital dermoscopic images has been collected for this study. The images have been separated into two independent groups; training sample and test sample. The training sample consists of 68 lesions from two different sources. 20 of the lesions come from the University Hospital of North Norway (UNN), the rest comes from Germany. All the lesions from UNN come from patients that have been referred to a specialist by a general physicians. The lesions from Germany come from patients without referral. The lesions, both from UNN and Germany, have been photographed by a trained expert dermatologist. After photographing, the lesions have been removed and sent to pathology for testing. Among the 234 lesions, only 5 are confirmed melanomas. 10 images have been excluded according to the following criteria: The dermoscope is not in full contact with the skin (2 images), the fluid used for refraction of the light covers less than half of the image (2 images), the lesion exceeds the image border (more than 16 mm in

diameter) (5 images), or the image is out of focus (1 image).

The remaining 224 lesions are both melanocytic and non-melanocytic. Both the training sample and the test sample contains lesions of large variation in size, color and contrast to the surrounding skin. The smallest lesion in the training sample covers an area of 23,000 pixels and has a major axis length of about 2 mm (Figure 5.14(d)), while the largest lesion in the training set covers an area of 660,000 pixels and has a major axis length of about 14 mm (Figure 5.14(f)).

Chapter 2

Mathematics, Statistics and Image Processing

Definitions of mathematical and statistical expressions used later in the thesis are presented in this chapter. Much of the mathematics used in both multivariate statistics and image analysis is based on matrix algebra. The definitions are just as general as they need to be for the thesis, and do not necessarily cover all mathematical eventualities.

2.1 Polynomial Interpolation

Polynomial interpolation is a method for constructing a continuous function, $f(x)$, on the bases of a few points, $(x_0, y_0), (x_1, y_1), \dots, (x_k, y_k)$, where the points will be parts of the resulting function, $f(x_0) = y_0, f(x_1) = y_1, \dots, f(x_k) = y_k$ (Kincaid and Cheney, 2002, p.308-330).

Theorem 1 (Polynomial Interpolation). *If x_0, x_1, \dots, x_k are distinct real numbers, then for arbitrary values y_0, y_1, \dots, y_k , there is a unique polynomial p_k of degree at most k such that*

$$p_k(x_i) = y_i, \quad \text{for } i = 0, 1, \dots, k.$$

For proof of Theorem 1, see Kincaid and Cheney (2002, p.309).

Definition 1 (Divided Differences). *Given a set of $k + 1$ data points,*

$(x_0, y_0), (x_1, y_1), \dots, (x_k, y_k)$, the divided differences are

$$f[x_0] \equiv y_0,$$

$$f[x_0, x_1, \dots, x_j] = \frac{f[x_1, x_2, \dots, x_j] - f[x_0, x_1, \dots, x_{j-1}]}{x_j - x_0}, \quad j = 1, \dots, k.$$

The notation $f[x_0, x_1, \dots, x_j]$ (not to be confused with $f(x_0, x_1, \dots, x_j)$) is used for historical reasons, for details, see Kincaid and Cheney (2002, p.327-330).

Definition 2 (Interpolation Polynomial in Newton's Form). *Given a set of $k + 1$ data points, $(x_0, y_0), (x_1, y_1), \dots, (x_k, y_k)$, the interpolation polynomial in Newton's form, $p(x)$, is defined as (Kincaid and Cheney, 2002, p.328)*

$$p(x) = \sum_{j=0}^k c_j q_j(x), \quad \text{where}$$

$$q_j(x) = \prod_{i=0}^{j-1} (x - x_i) \quad \text{and} \quad c_j = f[x_0, x_1, \dots, x_j].$$

Note that $\prod_{i=0}^m (x - x_j) = 1$, whenever $m < 0$.

2.2 Matrix Algebra

A digital image can be represented as a matrix or an array, and matrix algebra is therefore highly appreciated in image analysis.

Definition 3 (Transpose). *The transpose of a $n \times p$ matrix \mathbf{A} with entries $[a_{ij}]$ is the $p \times n$ matrix \mathbf{A}' with entries $[a'_{ij}] = [a_{ji}]$ (Nicholson, 1995, p.41).*

Definition 4 (Trace). *The trace of a $p \times p$ square matrix \mathbf{A} with entries $[a_{ij}]$ is the sum of the diagonal elements (Nicholson, 1995, p.257),*

$$\text{tr}(\mathbf{A}) = \sum_{i=1}^n a_{ii}.$$

Definition 5 (Eigenvectors and Eigenvalues). *Given a $p \times p$ square matrix \mathbf{A} , a $p \times 1$ nonzero vector $\mathbf{v} = [v_1, v_2, \dots, v_p]$ is an eigenvector if it satisfies the linear equation*

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v}, \quad (2.2.1)$$

where the scalar λ is the eigenvalue corresponding to the eigenvector \mathbf{v} (Nicholson, 1995, p.251).

The eigenvector \mathbf{v} is not unique. Any scalar multiple of \mathbf{v} , $a\mathbf{v}$, satisfies Equation 2.2.1. For some purposes the eigenvectors are therefore *normalized*;

$$\mathbf{e} = \frac{\mathbf{v}}{\|\mathbf{v}\|}, \quad \|\mathbf{v}\| = \sqrt{v_1^2 + v_2^2 + \cdots + v_p^2}.$$

The result of a normalization is that $\mathbf{e}'\mathbf{e} = \sum_{i=1}^p e_i e_i = 1$.

Definition 6 (Symmetric Matrix). A square matrix \mathbf{A} with entries $[a_{ij}]$ is symmetric if $\mathbf{A} = \mathbf{A}'$, or equivalently (Nicholson, 1995, p.42),

$$a_{ij} = a_{ji}, \quad \forall i, j.$$

Definition 7 (Positive Definite). A $p \times p$ symmetric matrix Σ is positive definite if (Johnson and Wichern, 2002, p.63)

$$\mathbf{a}'\Sigma\mathbf{a} > 0, \quad \forall p \times 1 \text{ vectors } \mathbf{a} \neq \mathbf{0}.$$

Definition 8 (Orthogonality and Orthonormality). Two $p \times 1$ vectors, $\mathbf{u} = [u_1, u_2, \dots, u_n]$ and $\mathbf{v} = [v_1, v_2, \dots, v_n]$ are orthogonal if (Nicholson, 1995, p.271)

$$\mathbf{u}'\mathbf{v} = \sum_{i=1}^n u_i v_i = 0.$$

The vectors \mathbf{u} and \mathbf{v} are orthonormal if they are orthogonal and

$$\|\mathbf{u}\| = 1 \quad \text{and} \quad \|\mathbf{v}\| = 1.$$

Theorem 2 (Eigenvectors of symmetric matrices). If \mathbf{A} is a $p \times p$ symmetric matrix, then \mathbf{A} has an orthonormal set of p eigenvectors.

For proof of Theorem 2, see Nicholson (1995, p.283).

Theorem 3 (Eigenvalues of real symmetric matrices). If \mathbf{A} is a real, symmetric matrix, then the eigenvalues are real.

For proof of Theorem 3, see Nicholson (1995, p.307)

Theorem 4 (Positive Definite). *A square matrix is positive definite if it is symmetric and all its eigenvalues are real.*

For proof of Theorem 4, see Nicholson (1995, p.290).

2.3 Statistical Analysis

In statistical analysis there is a differentiation between univariate statistics and multivariate statistics.

2.3.1 Univariate Statistics

In this thesis the univariate random variable, X , is defined to be a real scalar number.

Definition 9 (Mean). *The expected value or mean, $\mu_X = E(X)$, of a continuous probability function, $f(x)$, is*

$$\mu_X = E(X) = \int_{-\infty}^{\infty} xf(x) dx.$$

For a discrete probability function $p(x)$ the mean is

$$\mu_X = E(X) = \sum_i x_i p(x_i).$$

If $E|X| = \infty$, the mean is said to not exist (Casella and Berger, 2002, p.55). The sample mean, \bar{X} , of a random sample, X_1, X_2, \dots, X_n , is defined as (Casella and Berger, 2002, p.212)

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i. \quad (2.3.1)$$

Definition 10 (Variance). *The variance, $\sigma^2 = Var(X)$, of a random variable X is given by (Casella and Berger, 2002, p.59)*

$$\sigma^2 = Var(X) = E[(X - \mu_X)^2].$$

The standard deviation, σ , is given by the positive square root of σ^2 . The sample variance, S^2 , of a random sample X_1, X_2, \dots, X_n , is defined as

(Casella and Berger, 2002, p.212)

$$S^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2.$$

The sample standard deviation, S , is given by the positive square root of the sample variance.

Definition 11 (Median). *The median of a distribution is a value m such that (Casella and Berger, 2002, p.78)*

$$\begin{aligned} P(X \leq m) = P(X \geq m) &= \frac{1}{2}, & \text{for discrete distributions, and} \\ \int_{-\infty}^m f(x) dx = \int_m^{\infty} f(x) dx &= \frac{1}{2}, & \text{for continuous distributions.} \end{aligned}$$

The sample median is the middle of an ordered sample. To find the sample median, order the n observations in ascending order. If n is an odd number, the sample median is the $\frac{n+1}{2}$ th number. If n is an even number, the median is the average of the $\frac{n}{2}$ th and the $(\frac{n}{2} + 1)$ th number (Casella and Berger, 2002, p.226).

Definition 12 (Covariance). *The covariance, $Cov(X, Y)$, between two random variables X and Y is defined as (Casella and Berger, 2002, p.169)*

$$Cov(X, Y) = E[(X - \mu_X)(Y - \mu_Y)].$$

Lemma 1 (Covariance).

$$Cov(X, Y) = Cov(Y, X).$$

Proof of Lemma 1.

$$\begin{aligned} Cov(X, Y) &= E[(X - \mu_X)(Y - \mu_Y)] \\ &= E[XY - X\mu_Y - \mu_X Y + \mu_X \mu_Y] \\ &= E[YX - Y\mu_X - \mu_Y X + \mu_Y \mu_X] \\ &= Cov(Y, X). \end{aligned}$$

□

Definition 13 (Quantile). *A quantile, $q(p)$, of a sample is a value for which a specified fraction, p , of the data values is less than or equal to $q(p)$ (Walpole et al., 2002, p.204).*

Definition 14 (Interquartile Range (IQR)). *The interquartile range, IQR, is defined as (Martinez and Martinez, 2002, p.71)*

$$IQR = q(0.75) - q(0.25),$$

where $q(p)$ is the quantile defined in Definition 13.

Definition 15 (Conditional Probability). *If A and B are events in the sample space S , and $P(B) > 0$, then the conditional probability of A given B is (Casella and Berger, 2002, p.20)*

$$P(A|B) = \frac{P(A \cap B)}{P(B)}.$$

Definition 16 (Moments). *The k th moment of the random variable X , μ'_k , is (Casella and Berger, 2002, p.59)*

$$\mu'_k = EX^k.$$

The k th central moment of the random variable X , μ_k , is

$$\mu_k = E(X - \mu_X)^k.$$

Definition 17 (Sample Moments). *If X_1, X_2, \dots, X_n is a random sample, the k th sample moment, m_k is (Casella and Berger, 2002, p.312)*

$$m_k = \frac{1}{n} \sum_{i=1}^n X_i^k.$$

2.3.2 Multivariate Statistics

In this thesis, the multivariate random variable \mathbf{X} is defined to be an $n \times 1$ vector of real numbers.

Definition 18 (Mean Vector). *If the p random variables X_1, X_2, \dots, X_p of the $p \times 1$ random vector \mathbf{X} have means $\mu_{X_1}, \mu_{X_2}, \dots, \mu_{X_p}$, respectively, then the mean vector, $\mu_{\mathbf{X}}$, of \mathbf{X} is the $p \times 1$ vector (Johnson and Wichern, 2002,*

p.70)

$$\mu_{\mathbf{X}} = E\mathbf{X} = \begin{pmatrix} EX_1 \\ EX_2 \\ \vdots \\ EX_p \end{pmatrix} = \begin{pmatrix} \mu_{X_1} \\ \mu_{X_2} \\ \vdots \\ \mu_{X_p} \end{pmatrix}.$$

The sample mean vector, $\bar{\mathbf{X}}$, of a random sample $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_n$, where each $\mathbf{X}_i, i = 1, 2, \dots, n$, is a $p \times 1$ vector, $\mathbf{X}_i = [X_{i1} X_{i2} \dots X_{ip}]$ is the $p \times 1$ vector

$$\bar{\mathbf{X}} = \begin{pmatrix} \bar{X}_1 \\ \bar{X}_2 \\ \vdots \\ \bar{X}_p \end{pmatrix},$$

where \bar{X}_i is the univariate sample mean defined in Equation 2.3.1 (Johnson and Wichern, 2002, p.122).

Definition 19 (Covariance Matrix). The covariance matrix, Σ , of a $p \times 1$ random vector \mathbf{X} is the $p \times p$ matrix

$$\Sigma = Cov(\mathbf{X}) = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \dots & \sigma_{1p} \\ \sigma_{21} & \sigma_{22} & \dots & \sigma_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{p1} & \sigma_{p2} & \dots & \sigma_{pp} \end{pmatrix},$$

where the entry $[\sigma_{ij}]$ is the covariance $Cov(X_i, X_j)$ between the random variables X_i and X_j as defined in Definition 12 (Johnson and Wichern, 2002, p.70).

Note that Σ is a real, symmetric matrix. All its entries are real, and $\sigma_{ij} = \sigma_{ji}$ since $Cov(X, Y) = Cov(Y, X)$ (see Lemma 1).

The sample covariance matrix, \mathbf{S} , of a random sample $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_n$, is the $p \times p$ matrix with entries $[s_{ij}]$, where

$$s_{ij} = \frac{1}{n} \sum_{k=1}^n (X_{ki} - \bar{X}_i)(X_{kj} - \bar{X}_j).$$

A note on notation

In univariate statistics, the most common way to denote the variance is by σ^2 . In multivariate statistics, the entries of the covariance matrix, Σ , are commonly denoted σ_{ij} .

Theorem 5 (Variance of Linear Combinations). *If \mathbf{a} is a $p \times 1$ vector of real numbers, and \mathbf{X} is a $p \times 1$ vector of random variables with covariance matrix Σ , then the linear combination $\mathbf{a}'\mathbf{X} = a_1X_1 + a_2X_2 + \dots + a_pX_p$ has variance*

$$\text{Var}(\mathbf{a}'\mathbf{X}) = \mathbf{a}'\Sigma\mathbf{a}.$$

For proof of Theorem 5, see Johnson and Wichern (2002, p.76).

2.4 Filtering

Often the first step in image analysis is to filter the image. There exists an unlimited number of filters, and the choice of filter must reflect the next step in the analysis of the image. The similarity between almost all image filters is that they try to remove noise while preserving the original image. Noise in an image does not have one precise definition. When recording the reality, the image is never an exact copy, and the difference between the image and the reality can be thought of as noise. Often this will be pixels having a different value than the neighboring pixels, where the value cannot be explained by the object in the image itself. Another way to define noise is to say that noise is all the unwanted or unnecessary information in the image. In that case, a hair in an image of a skin lesion can be regarded as noise. This thesis will stick to the first definition of noise, which is the most intuitive and general definition.

2.4.1 Noise Removing Filters

Schmid (1999c) gives in his thesis an extensive introduction to noise-removing image filters. The Gaussian filter used by Magliogannis et al. (2006) and Xu et al. (1999) fulfills the noise removal, but fails on the edge preserving, which is important when it comes to segmentation (Schmid, 1999c, p.9-10). The Wiener filter has the same disadvantage of over-smoothing the image.

Several different Wiener filters were tried, but the edge-preserving capabilities were too poor. A morphological filter has the great disadvantage of not respecting the natural shape of the objects in the image. The result of morphological filtering is highly dependent on the shape of the structuring element compared to the shape of the objects in the image, and is therefore not suited if prior knowledge is missing (Schmid, 1999c, p.13). A simple and popular filter is the *median* filter, used in its original form in Zagrouba and Barhoumi (2004), Zagrouba and Barhoumi (2005), Magliogannis (2003) and Zhang et al. (2000), as *pseudo-median* filter in Hance et al. (1996), or as *multi-stage* median filter in Lee (2001).

Median filter

A median filter is, as the name suggests, based on finding the median of a set of numbers. As most noise-removing filters, median filters are *low-pass* filters. A low-pass filter is a filter that passes low-frequency signals, but reduces the amplitude of signals with high frequency. The median filter replaces the current pixel by the median of the neighboring pixels, including the current pixel. The neighborhood can be of any size and shape, but for noise removing it will normally be a square or a disk. As always, a correct choice of parameters is important, in this case the size and the shape of the neighborhood. If the neighborhood is too big, even a median filter will smoothen the edges; a neighborhood too small will not remove enough noise. Objects that are smaller than the neighborhood can be totally removed when using a median filter (Schmid, 1999c, p.10). The median filter is computer intensive, and has therefore been replaced by the pseudomedian filter in Hance et al. (1996). The problem with the pseudomedian filter is that it can introduce line artifacts and generally gives a poorer result than the median filter (Schmid, 1999c, p.10). The multi-stage median filter has even better edge-preserving capabilities than the median filter (Lee, 2001, p.44), but is more computer intensive. Schmid (1999c, p.15–37) introduces a filter based on nonlinear isotropic diffusion, that performs better on the edge-preserving than the median filter, but is again more computer intensive. His conclusion is that the median filter has very good denoising capabilities and is sufficient in most cases (Schmid, 1999c, p.87), and uses it himself in Schmid (1999a).

Celebi et al. (2007) use a median filter to smooth the images before border detection, but with a big enough neighborhood to smooth out artifacts

such as hair and air bubbles. This method for removing hair fails when the hairs are too thick or they appear in clusters, so they use the DullRazor-algorithm (see Section 4.2 for details) when the filtering does not give the wanted result.

2.5 Presentation of Images

All images presented in this thesis are digital images. In a digital image, each pixel is associated with a scalar number or vector. These values can be represented to the human eye in different ways. Normally one will choose the presentation that lies closest to the common understanding of what an object looks like in the real world. Therefore, all three-layer color images are represented normalized; $Image = Image / maximum(Image)$.

The pre-processing (see Chapter 3) starts with a $2400 \times 3276 \times 3$ pixel image and returns a $1600 \times 1600 \times 3$ pixel image with a black mask superimposed. The mask has a disk-shaped opening with a radius of 800 pixels. Although the image is not cropped to its final size before the removal of artifacts is finished, the images presented in the thesis all have the size of 1600×1600 pixels with a 5 pixel wide frame, unless otherwise mentioned. The only pixels relevant for final segmentation are these pixels. The pixels used for intermediate calculations are therefore not shown.

2.5.1 Presentation of Grayscale Images

The two images shown in Figure 2.1, are both presentations of the same intensity- or *grayscale*-image (one-layer image), but they are displayed in two different ways. The reason for displaying a grayscale image in color is that when an image is displayed in shades of gray, the intensity differences might be difficult to perceive for the human eye. By displaying intensity images in colors from dark blue (lowest value) to dark red (highest value), the differences in intensity are clearer. When necessary, a colorbar (as seen in Figure 2.1(b)), showing the pixel values of the various colors, is displayed next to the image. Note that this colorbar is unique for each image. The disadvantage of displaying intensity images in color is that the observer can get a false impression of discontinuous jumps. In Figure 2.1(b) it might seem like there are four levels (blue, green, yellow and red) that are quite separated. In fact, the transition between red and yellow is just as smooth

as the one between darker red and lighter red. Displaying intensity images in color is merely done to facilitate the apprehension; the pixel values remain unchanged.

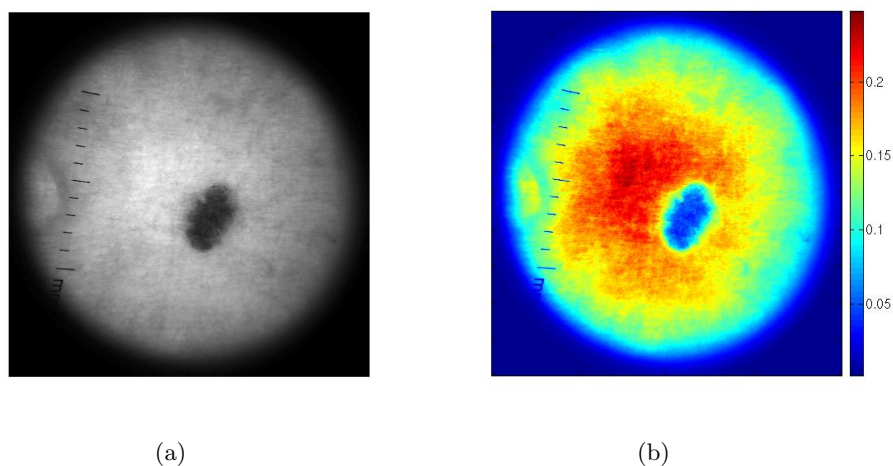


Figure 2.1: The same 2000×2000 pixel grayscale image in (a) gray presentation (b) color presentation.

2.6 Gray Conversion

A truecolor, RGB, image is an $n \times m \times 3$ pixel array. The layers of the image represent the color red, green and blue, respectively. For several purposes, it might be necessary to reduce the three-layer array to a one-layer matrix. When this is done, we get a grayscale image. There are various methods that can be applied for an RGB-to-grayscale conversion. Depending on the purpose of the conversion, one method might be better suited than another.

Two methods for gray conversion are introduced here. The first and simplest one is a linear transformation with fixed weights. The pixel values in the resulting grayscale matrix is a weighted sum of the three corresponding pixel values in the truecolor array. The second method is also a linear transformation, but the weights in the sum is individually computed for each image.

2.6.1 RGB-to-luminance

To transform an RGB image to a grayscale image representing the *luminance* of the original image, a linear transformation is used. The transformation used is derived from the NTSC-standard. The NTSC (National Television System Committee) is a standard used for analog television signals. The RGB color image is transformed to the NTSC image YIQ by the following operation (Burdick, 1997, p.33)

$$\begin{pmatrix} Y \\ I \\ Q \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}. \quad (2.6.1)$$

The Y component represents the luminance, the I component represents the hue (hue is approximately proportional to the average wavelength of a color), and the Q component represents the saturation (saturation measures the amount of white in a color). The Y component is the grayscale signal used to display images in a monochrome (black and white) TV. The RGB-to-luminance transformation uses the weights from the Y component to convert an RGB image to a grayscale image. The following linear combination is used (MathWorks)

$$grayscale = 0.2989 \cdot R + 0.5870 \cdot G + 0.1140 \cdot B,$$

where R , G and B represents the three layers in the color image.

2.6.2 Principal Component Transform

Principal component transformation (PCT) (also referred to as Karhunen-Loève transformation or Hotelling transformation) is a transformation used to analyze, and possibly reduce the dimension of, multi-dimensional data sets (Johnson and Wichern, 2002, p.426). In the case of image analysis the data set has three dimensions (the three layers of the color image array) that should be reduced to a one-dimensional data set (grayscale image). The pixel values are regarded as samples from an unknown probability distribution. The PCT-method seeks to maximize the *proportion of total population variance explained* (see Definition 21) by the new lower-dimensional data set.

The PCT-method is an orthogonal linear transformation that transforms the data set into a new coordinate system. This is done in such a manner that the greatest variance lies in the direction of the first axis, the second greatest variance lies in the direction of the second axis, and so on. The new axes are mutually orthogonal, meaning that the new data sets (vectors) are mutually independent.

If a data set consists of n measurements on p variables, and the number of variables should be reduced to $k < p$, principal component transformation might be used. Principal components are linear combinations of the vectors $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p$, all with the same length n . From p vectors, p principal components are calculated. The new lower-dimensional data set will consist of only one or some of the principal components.

Definition 20 (Principal Component). *If \mathbf{X} is a $p \times n$ matrix and \mathbf{a}_i is a $p \times 1$ vector, where $\mathbf{a}_i' \mathbf{a}_i = 1$, $i = 1, 2, \dots, p$, then the i th principal component \mathbf{y}_i is the linear combination $\mathbf{a}_i' \mathbf{X}$ that maximizes the variance*

$$\text{Var}(\mathbf{a}_i' \mathbf{X}), \quad \text{and meets the requirement} \quad (2.6.2)$$

$$\text{Cov}(\mathbf{a}_i' \mathbf{X}, \mathbf{a}_k' \mathbf{X}) = 0, \quad \text{for } k < i \quad (2.6.3)$$

(Johnson and Wichern, 2002, p.427).

Theorem 6 (Principal Component). *Let the p vectors $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p$, all with the same length n , represent a data set of n measurements on p variables. Let the p vectors $\mathbf{x}'_1, \mathbf{x}'_2, \dots, \mathbf{x}'_p$ be the rows in a $p \times n$ matrix \mathbf{X} . Let Σ be the covariance matrix corresponding to the $n \times p$ matrix \mathbf{X}' . Let the eigenvalue-eigenvector pairs of Σ , $(\lambda_1, \mathbf{e}_1), (\lambda_2, \mathbf{e}_2), \dots, (\lambda_p, \mathbf{e}_p)$, be ordered such that $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p \geq 0$, and normalized such that $\mathbf{e}_i' \mathbf{e}_i = 1$, $i = 1, 2, \dots, p$. Then the i th principal component \mathbf{y}_i is given by*

$$\mathbf{y}_i = \mathbf{e}_i' \mathbf{X}, \quad i = 1, 2, \dots, p. \quad (2.6.4)$$

Lemma 2 (Maximization). *If \mathbf{a} is a $p \times 1$ vector and Σ is a $p \times p$ positive definite matrix with eigenvalue-eigenvector pairs $(\lambda_1, \mathbf{e}_1), (\lambda_2, \mathbf{e}_2), \dots, (\lambda_p, \mathbf{e}_p)$*

where $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p \geq 0$, then

$$\max_{\mathbf{a} \neq \mathbf{0}} \frac{\mathbf{a}'\boldsymbol{\Sigma}\mathbf{a}}{\mathbf{a}'\mathbf{a}} = \lambda_1, \quad \text{attained when } \mathbf{a} = \mathbf{e}_1, \text{ and} \quad (2.6.5)$$

$$\max_{\mathbf{a} \perp \mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_k} \frac{\mathbf{a}'\boldsymbol{\Sigma}\mathbf{a}}{\mathbf{a}'\mathbf{a}} = \lambda_{k+1}, \quad \text{attained when } \mathbf{a} = \mathbf{e}_{k+1}, k = 1, 2, \dots, p-1. \quad (2.6.6)$$

For proof of Lemma 2, see Johnson and Wichern (2002, p.81).

Proof of Theorem 6. Since the covariance matrix $\boldsymbol{\Sigma}$ is real and symmetric it has real eigenvalues according to Theorem 3. It must then be positive definite according to Theorem 4. Theorem 2 shows that $\boldsymbol{\Sigma}$ has an orthonormal set of p eigenvalues. So, $\mathbf{e}_i'\mathbf{e}_i = 1$, for $i = 1, 2, \dots, p$ and $\mathbf{e}_i \perp \mathbf{e}_j$, for $i \neq j$. By using Equation 2.6.5 it follows that

$$\lambda_1 = \frac{\mathbf{e}_1'\boldsymbol{\Sigma}\mathbf{e}_1}{\mathbf{e}_1'\mathbf{e}_1} = \mathbf{e}_1'\boldsymbol{\Sigma}\mathbf{e}_1 = \text{Var}(\mathbf{e}_1\mathbf{X}).$$

From Theorem 5 we get that $\text{Var}(\mathbf{e}_1\mathbf{X}) = \mathbf{e}_1'\boldsymbol{\Sigma}\mathbf{e}_1$. Similarly, by using Equation 2.6.6 it follows that

$$\lambda_{k+1} = \frac{\mathbf{e}_{k+1}'\boldsymbol{\Sigma}\mathbf{e}_{k+1}}{\mathbf{e}_{k+1}'\mathbf{e}_{k+1}} = \mathbf{e}_{k+1}'\boldsymbol{\Sigma}\mathbf{e}_{k+1} = \text{Var}(\mathbf{e}_{k+1}\mathbf{X}), \quad k = 1, 2, \dots, p-1.$$

The eigenvalues maximizes the variance for each $i = 1, 2, \dots, p$, so the maximization of Equation 2.6.2 holds. It remains to prove that Equation 2.6.3 also holds. Since $\boldsymbol{\Sigma}$ is a symmetric matrix, we have p orthonormal eigenvectors. Therefore $\mathbf{e}_i'\mathbf{e}_k = 0$, for $i \neq k$ and

$$\begin{aligned} \text{Cov}(\mathbf{y}_i, \mathbf{y}_k) &= \text{Cov}(\mathbf{e}_i'\mathbf{X}, \mathbf{e}_k'\mathbf{X}) = \mathbf{e}_i'\text{Cov}(\mathbf{X}, \mathbf{X})\mathbf{e}_k = \mathbf{e}_i'\text{Cov}(\mathbf{X})\mathbf{e}_k \\ &= \mathbf{e}_i'\boldsymbol{\Sigma}\mathbf{e}_k = \mathbf{e}_i'\lambda_k\mathbf{e}_k = \lambda_k\mathbf{e}_i'\mathbf{e}_k = 0 \end{aligned}$$

□

Definition 21 (Total Population Variance). *The total population variance is defined as the sum of the diagonal elements in the covariance matrix $\boldsymbol{\Sigma}$ (Johnson and Wichern, 2002, p.429)*

$$\text{Total population variance} = \sigma_{11} + \sigma_{22} + \dots + \sigma_{pp} = \text{tr}(\boldsymbol{\Sigma}).$$

Theorem 7.

$$\text{tr}(\mathbf{\Sigma}) = \lambda_1 + \lambda_2 + \cdots + \lambda_p.$$

For proof of Theorem 7, see Nicholson (1995, p.309).

Lemma 3 (Total Variance Explained). *From Theorem 7 it immediately follows that the proportion of total variance explained by the i th principal component is*

$$\frac{\lambda_i}{\lambda_1 + \lambda_2 + \cdots + \lambda_p}, \quad i = 1, 2, \dots, p.$$

2.6.3 Application

Figure 2.2 shows an RGB color image of a skin lesion. When the PCT-

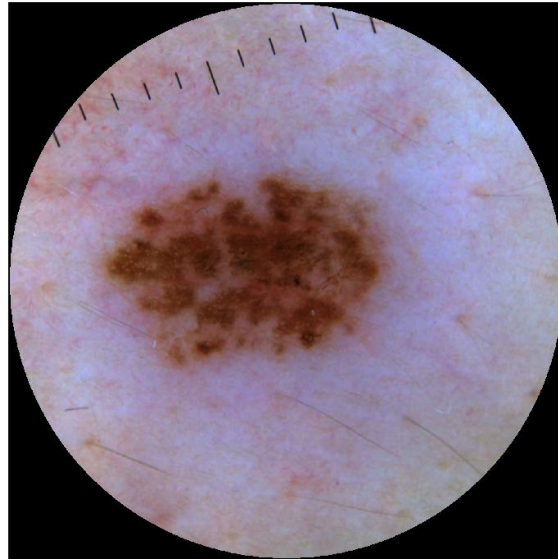


Figure 2.2: Three-layer color image.

method is applied on this image, the three separate grayscale images in Figure 2.3 are produced.

The graph in Figure 2.4 shows the proportion of variance explained by the first, second and third principal component. The proportion of variance

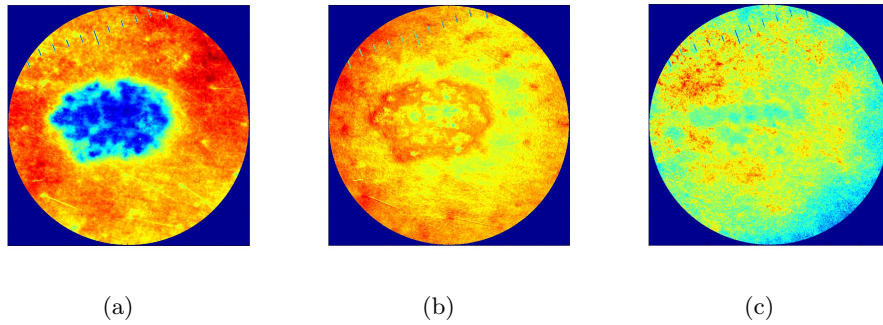


Figure 2.3: Principal components. (a) First. (b) Second. (c) Third.

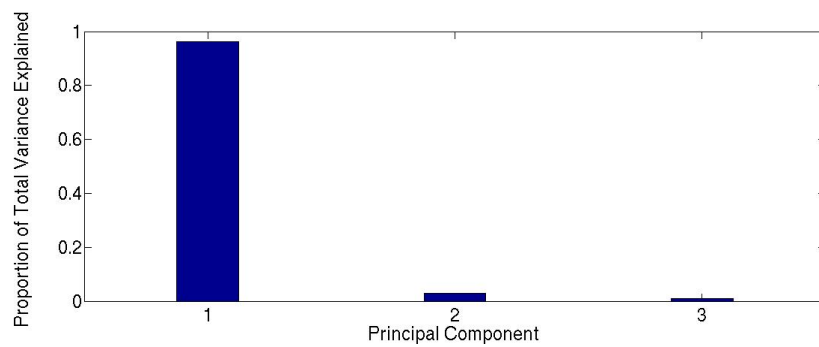


Figure 2.4: Proportion of total variance explained by the three principal components, respectively.

explained by the first principal component is more than 95%, so there are good reasons to use this as a grayscale image for further analysis. The grayscale image produced by using the RGB-to-luminance transformation (see Figure 2.5(b)), is noticeably different from the first principal component image. PCT is a common data reduction method for image analysis in the

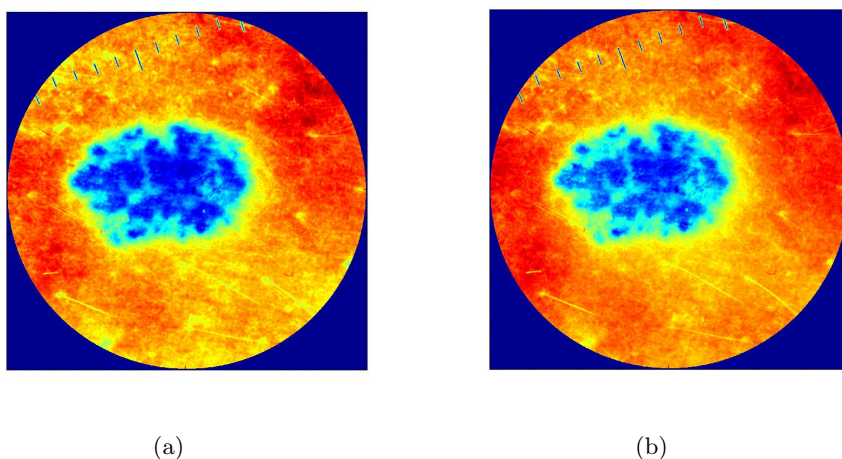


Figure 2.5: Grayscale image based on (a) principal component transform (b) RGB-to-luminance transform.

purpose of image segmentation (Fleming et al., 1998; Hance et al., 1996; Schmid-Saugeon et al., 2003; Zagrouba and Barhoumi, 2004, 2005). It is also worth noting that the last principal component contains most of the image noise, and by removing it has a positive effect on the segmentation (Schmid, 1999c, p.55).

2.7 Binary Images

While both the truecolor images and the grayscale images have pixel values from a certain range (depending on the data class), a binary image consists of pixels with value 0 or 1. When a binary image is made from a grayscale image, a threshold value is set. All pixels with value below the threshold will be set to 0, while the rest will be set to 1. The threshold value can be decided from a number of criteria, according to what the binary image is meant to represent. The simplest is to choose a fixed threshold. If this cannot be done, an automatic selection of threshold value must be used.

2.7.1 Otsu's Method

Nobuyuki Otsu developed this method in 1979 (Otsu, 1979):

The pixels of a grayscale image is represented by L gray levels. The number of pixels at level i is denoted by n_i . The gray-level histogram is normalized and regarded as a probability distribution where

$$p_i = n_i/N, \quad p_i \geq 0, \quad \sum_{i=1}^L p_i = 1,$$

where N is the total number of pixels. The pixels are put in one of two classes, C_0 or C_1 . The task is to find the optimal threshold value k^* . All pixels with value below this threshold will be put in class C_0 , the rest in C_1 . The probability of class occurrence is

$$\begin{aligned} \omega_0 = P(C_0) &= \sum_{i=1}^k p_i = \omega(k), \\ \omega_1 = P(C_1) &= \sum_{i=k+1}^L p_i = 1 - \omega(k). \end{aligned}$$

The probabilities sum up to 1 (as they should):

$$\omega_0 + \omega_1 = \sum_{i=1}^k p_i + \sum_{i=k+1}^L p_i = \sum_{i=1}^L p_i = 1.$$

The total mean and class means are, respectively

$$\begin{aligned} \mu_T &= \sum_{i=1}^L ip_i, \\ \mu_0 &= \sum_{i=1}^k P(i|C_0) = \sum_{i=1}^k ip_i/\omega_0 = \mu(k)/\omega(k), \\ \mu_1 &= \sum_{i=k+1}^L P(i|C_1) = \sum_{i=k+1}^L ip_i/\omega_1 = \frac{\mu_T - \mu(k)}{1 - \omega(k)}, \end{aligned}$$

where

$$\omega(k) = \sum_{i=1}^k p_i \quad \text{and} \quad \mu(k) = \sum_{i=1}^k ip_i.$$

The relation between total mean and class means is

$$\omega_0\mu_0 + \omega_1\mu_1 = \mu_T,$$

since

$$\begin{aligned} \omega_0\mu_0 + \omega_1\mu_1 &= \omega_0 \sum_{i=1}^k ip_i/\omega_0 + \omega_1 \sum_{i=k+1}^L ip_i/\omega_1 \\ &= \sum_{i=1}^k ip_i + \sum_{i=k+1}^L ip_i = \sum_{i=1}^L ip_i = \mu_T. \end{aligned}$$

The total variance and the class variances are, respectively

$$\begin{aligned} \sigma_T^2 &= \sum_{i=1}^L (i - \mu_T)^2 p_i, \\ \sigma_0^2 &= \sum_{i=1}^k (i - \mu_0)^2 P(i|C_0) = \sum_{i=1}^k (i - \mu_0)^2 p_i \omega_0, \\ \sigma_1^2 &= \sum_{i=k+1}^L (i - \mu_1)^2 P(i|C_1) = \sum_{i=k+1}^L (i - \mu_1)^2 p_i \omega_1. \end{aligned}$$

The within-class variance and between-class variance (also referred to as intra- and inter-class variance) are, respectively

$$\begin{aligned} \sigma_W^2 &= \omega_0\sigma_0^2 + \omega_1\sigma_1^2, \\ \sigma_B^2 &= \omega_0(\mu_0 - \mu_T)^2 + \omega_1(\mu_1 - \mu_T)^2 = \omega_0\omega_1(\mu_1 - \mu_0)^2. \end{aligned} \quad (2.7.1)$$

The right equal-sign in Equation 2.7.1 follows from

$$\begin{aligned}
\sigma_B^2 &= \omega_0(\mu_0^2 - 2\mu_0\mu_T + \mu_T^2) + \omega_1(\mu_1^2 - 2\mu_1\mu_T + \mu_T^2) \\
&= \omega_0\mu_0^2 - 2\omega_0\mu_0\mu_T + \omega_0\mu_T^2 + \omega_1\mu_1^2 - 2\omega_1\mu_1\mu_T + \omega_1\mu_T^2 \\
&= (\omega_0 + \omega_1)(\omega_0\mu_0 + \omega_1\mu_1)^2 + \omega_0\mu_0^2 + \omega_1\mu_1^2 \\
&\quad - 2\omega_0\mu_0(\omega_0\mu_0 + \omega_1\mu_1) - 2\omega_1\mu_1(\omega_0\mu_0 + \omega_1\mu_1) \\
&= \omega_0^2\mu_0^2 + 2\omega_0\omega_1\mu_0\mu_1 + \omega_1^2\mu_1^2 + \omega_0\mu_0^2 + \omega_1\mu_1^2 \\
&\quad - 2\omega_0^2\mu_0^2 - 2\omega_0\omega_1\mu_0\mu_1 - 2\omega_0\omega_1\mu_0\mu_1 - 2\omega_1^2\mu_1^2 \\
&= \omega_0\mu_0^2(\omega_0 + \omega_1) + \omega_1\mu_1^2(\omega_0 + \omega_1) - \omega_0^2\mu_0^2(\omega_0 + \omega_1) - \omega_1^2\mu_1^2(\omega_0 + \omega_1) \\
&\quad - 2\omega_0\omega_1\mu_0\mu_1 \\
&= \omega_0\omega_1(\mu_1^2 - 2\mu_0\mu_1 + \mu_0^2) + \omega_0^2\mu_0^2 + \omega_1^2\mu_1^2 - \omega_0^3\mu_0^2 - \omega_0^2\omega_1\mu_0^2 - \omega_0\omega_1^2\mu_1^2 \\
&\quad - \omega_1^3\mu_1^2 \\
&= \omega_0\omega_1(\mu_1 - \mu_0)^2 - \omega_0^2\mu_0^2(\omega_0 + \omega_1 - 1) - \omega_1^2\mu_1^2(\omega_0 + \omega_1 - 1) \\
&= \omega_0\omega_1(\mu_1 - \mu_0)^2.
\end{aligned}$$

Theorem 8 (Relation between variances). *The relation between within- and between-class variance, and total variance is*

$$\sigma_W^2 + \sigma_B^2 = \sigma_T^2. \quad (2.7.2)$$

Proof of Theorem 8.

$$\begin{aligned}
\sigma_T^2 &= \sum_{i=1}^L (i - \mu_T)^2 p_i = \sum_{i=1}^L (i^2 - 2\mu_T i + \mu_T^2) p_i \\
&= \sum_{i=1}^L (i^2 + \mu_T^2) p_i - 2\mu_T \sum_{i=1}^L i p_i = \sum_{i=1}^L (i^2 + \mu_T^2) p_i - 2\mu_T^2, \\
\sigma_W^2 &= \omega_0 \sigma_0^2 + \omega_1 \sigma_1^2 = \sum_{i=1}^k (i - \mu_0)^2 p_i + \sum_{i=k+1}^L (i - \mu_1)^2 p_i, \\
\sigma_B^2 &= \omega_0 (\mu_0 - \mu_T)^2 + \omega_1 (\mu_1 - \mu_T)^2 \\
&= \sum_{i=1}^k p_i (\mu_0 - \mu_T)^2 + \sum_{i=k+1}^L p_i (\mu_1 - \mu_T)^2, \\
\sigma_W^2 + \sigma_B^2 &= \sum_{i=1}^k (i^2 - 2\mu_0 i + \mu_0^2 + \mu_0^2 - 2\mu_0 \mu_T + \mu_T^2) p_i \\
&\quad + \sum_{i=k+1}^L (i^2 - 2\mu_1 i + \mu_1^2 + \mu_1^2 - 2\mu_1 \mu_T + \mu_T^2) p_i \\
&= \sum_{i=1}^L (i^2 + \mu_T^2) p_i + 2 \sum_{i=1}^k p_i \mu_0^2 - 2 \sum_{i=1}^k i p_i \mu_0 - 2 \sum_{i=1}^k p_i \mu_0 \mu_T \\
&\quad + 2 \sum_{i=k+1}^L p_i \mu_1^2 - 2 \sum_{i=k+1}^L i p_i \mu_1 - 2 \sum_{i=k+1}^L p_i \mu_1 \mu_T \\
&= \sum_{i=1}^L (i^2 + \mu_T^2) p_i + 2\omega(k) \mu_0^2 - 2\mu_0 \mu(k) - 2\omega(k) \mu_0 \mu_T \\
&\quad + 2(1 - \omega(k)) \mu_1^2 - 2\mu_1 (\mu_T - \mu(k)) - 2(1 - \omega(k)) \mu_1 \mu_T \\
&= \sum_{i=1}^L (i^2 + \mu_T^2) p_i + 2 \frac{\omega(k) \mu^2(k)}{\omega^2(k)} - 2 \frac{\mu(k) \mu(k)}{\omega(k)} - 2 \frac{\omega(k) \mu(k)}{\omega(k)} \mu_T \\
&\quad + 2 \frac{(1 - \omega(k)) (\mu_T - \mu(k))^2}{(1 - \omega(k))^2} - 2 \frac{(\mu_T - \mu(k)) (\mu_T - \mu(k))}{1 - \omega(k)} \\
&\quad - 2 \frac{(1 - \omega(k)) (\mu_T - \mu(k))}{1 - \omega(k)} \mu_T \\
&= \sum_{i=1}^L (i^2 + \mu_T^2) p_i - 2\mu(k) \mu_T - 2\mu_T^2 + 2\mu(k) \mu_T \\
&= \sum_{i=1}^L (i^2 + \mu_T^2) p_i - 2\mu_T^2 = \sigma_T^2.
\end{aligned}$$

□

For measuring how good the estimator fits the parameter, Otsu introduces three criteria from discriminant analysis;

$$\lambda = \sigma_B^2/\sigma_W^2, \quad \kappa = \sigma_T^2/\sigma_W^2, \quad \eta = \sigma_B^2/\sigma_T^2.$$

Because of Theorem 8, maximizing one of the discriminant criteria is equivalent to maximizing another. The criterion η is chosen for convenience. Since σ_T^2 is independent of the threshold level k , it is enough to maximize σ_B^2 to find the optimal threshold value k^* .

Let $\sigma_B^2(k)$ be the between-variance for threshold value k . From (2.7.1) we get

$$\begin{aligned} \sigma_B^2(k) &= \omega_0\omega_1(\mu_1 - \mu_0)^2 = \omega(k)[1 - \omega(k)] \left[\frac{\mu_T - \mu(k)}{1 - \omega(k)} - \frac{\mu(k)}{\omega(k)} \right]^2 \\ &= \omega(k)[1 - \omega(k)] \left[\frac{\omega(k)[\mu_T - \mu(k)] - \mu(k)[1 - \omega(k)]}{\omega(k)[1 - \omega(k)]} \right]^2 \\ &= \frac{[\omega(k)\mu_T - \mu(k)]^2}{\omega(k)[1 - \omega(k)]}. \end{aligned}$$

From 2.7.1 we see that σ_B^2 takes the minimum value of zero for $\omega(k) = 1$ or $\omega(k) = 0$. These cases occur when all pixels are either of class C_0 or C_1 . For $0 < \omega(k) < 1$ the value is positive and bounded, and the maximum is guaranteed to exist.

The value $\eta(k^*)$ has one very interesting interpretation. It can be used as a measure for the separability of classes or the bimodality of the histogram. It is uniquely determined within the range $0 \leq \eta(k^*) \leq 1$.

Calculating $\eta(k^*)$ for the grayscale images in the training sample showed no concurrence between $\eta(k^*)$ and the visually inspected bimodality of the histogram.

2.8 Morphological Operations

Morphological operations are widely used in image analysis (Ganster et al., 2001; Hance et al., 1996; Rajab et al., 2004; Schmid, 1999b; Lee et al., 1997; Gómez et al., 2008). As mentioned in Section 2.4.1, a morphological filter can be used for noise-removing. It is not recommended without any previous

knowledge of the image, since morphological filters are very shape sensitive. If prior knowledge of the objects of interest is available, morphological operations can be used to remove, smooth out or intensify certain objects present in the image.

Morphological operations are transformations that replace a pixel value with a new one on the basis of the values of the neighboring pixels. Depending on the size and shape of the neighborhood, the transformation is sensitive to certain objects in the image. The neighborhood is called a *structuring element*. The structuring element might be of any size and shape. The current pixel will always be the center pixel of the structuring element. This means that if the structuring element is a 7×1 vector, the current pixel will be the fourth pixel in the vector. If the structuring element is a 3×3 square, the current pixel will be the middle pixel.

2.8.1 Dilation and Erosion

There are two fundamental morphological operations; *dilation* and *erosion*. In a binary image, dilation adds pixels to the boundaries of an object, while erosion removes pixels from the boundaries. In a grayscale image, dilation sets the new pixel value to the maximum of the pixels in the neighborhood, while erosion sets the new pixel value to the minimum of the pixels in the neighborhood (Burdick, 1997, p.155-158). Say you have a binary image like the one in Figure 2.6(a). The image is dilated with a 3×3 pixel square, the result will be the image in Figure 2.6(c). Every pixel of value 1 from the old image is now surrounded by 8 pixels of value 1. Figure 2.6(d) shows the dilated image eroded with the same 3×3 structuring element. Every pixel that is not surrounded by 8 white pixels is removed.

2.8.2 Opening and Closing

Two important complex morphological operations are *opening* and *closing*, which both are combinations of dilation and erosion. Morphological opening is performed when, for example, removal of small unwanted objects or pieces of objects is wanted. A structuring element is created. The structuring element must be large enough to contain the unwanted objects. First, the image is eroded, and the unwanted objects are removed. But, the boundaries of larger or different shaped wanted objects are also removed. To restore

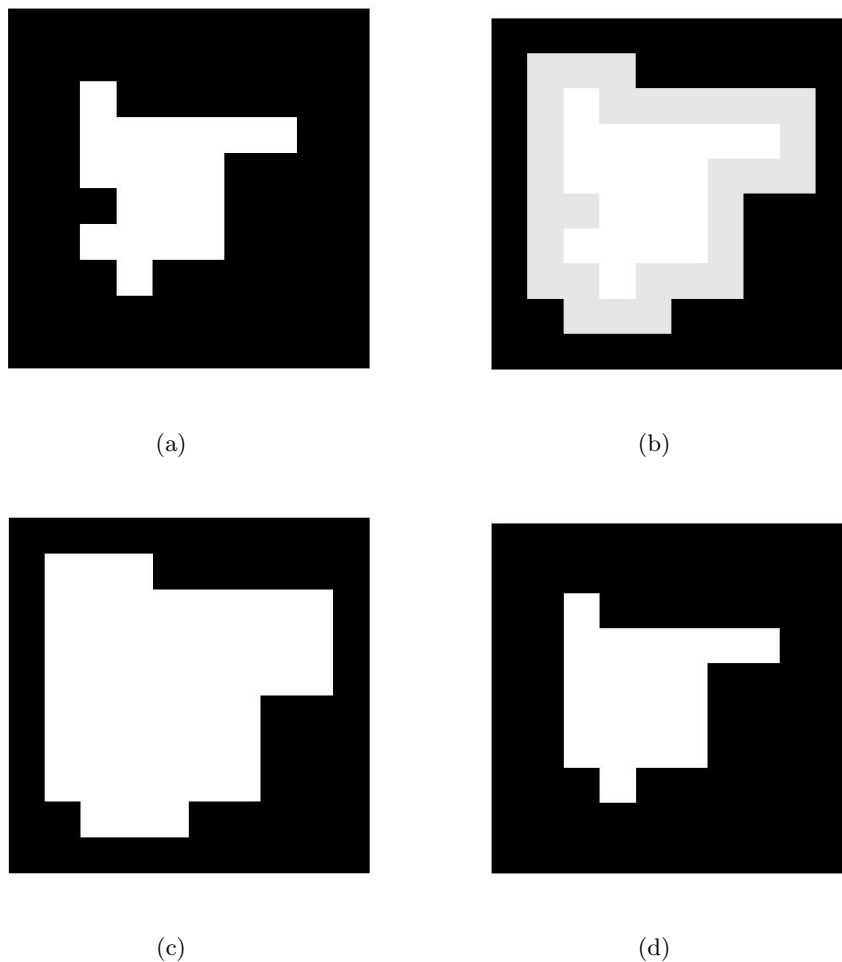


Figure 2.6: (a) Original binary image. (b) After dilation, new pixels of value 1 colored gray. (c) After dilation, all pixels of value 1 colored white. (d) After erosion.

the original size and shape of the wanted objects a dilation with the same structuring element is performed. This dilation does not restore the removed objects (Burdick, 1997, p.160-162).

Morphological closing can for example fill small holes within an object or connect two objects that are located close to each other. The structuring element must in this case be larger than the holes or the distance between the objects that are to be connected. The image is first dilated and then eroded with the same structuring element. The image in Figure 2.6(d) is the image in Figure 2.6(a) closed with a 3×3 structuring element.

2.9 Mapping

Mapping of a matrix or array in this context is used for describing the transformation of pixel destinations from the original image to a transformed image. There are two directions of transformation; forward transformation and inverse transformation. In forward transformation, each pixel in the original image is given a new location in the transformed image. This can, depending on the transformation, lead to holes in the transformed image; there might be pixels in the transformed image that are not given any value. This problem can be solved by using inverse transformation. In this case, each pixel in the transformed image is given a value according to the transformation and the pixel values in the original image. Most transformations lead to locations falling between pixels. These locations are called sub-locations. There are several methods for choosing the most appropriate locations based on the sub-locations. Among the most common methods are nearest neighbor and bilinear interpolation.

2.9.1 Rotation

The rotation of an image is a mapping. The pixel values in the rotated image are results of the pixel values in the original image. When rotating about the center of the image, the inverse rotation transformation leads to the following pixel sub-locations (Burdick, 1997, p.137):

$$\begin{aligned} i &= i_r \cos(\alpha) - j_r \sin(\alpha) + c_i(1 - \cos(\alpha)) + c_j \sin(\alpha), \\ j &= i_r \sin(\alpha) + j_r \cos(\alpha) - c_i \sin(\alpha) + c_j(1 - \cos(\alpha)), \end{aligned}$$

where

(i, j) is the location in the original image,
 (i_r, j_r) is the sub-location for the rotated image,
 (c_i, c_j) is the location of the center of the image and
 α is the angle of rotation.

The decimal numbers i and j must be converted into integers that can refer to a location in the original image. One way of doing this is by a method called nearest neighbor. This is a very simple and intuitive method. The decimal numbers i and j are simply rounded up or down to their nearest neighbor, and the value of pixel (i_r, j_r) in the rotated image will be that of the pixel in the original image closest to (i, j) . Nearest neighbor is fast compared to bilinear interpolation and only pixel values already present in the original image are allowed in the rotated image.

When using bilinear interpolation, the value of the pixels in the transformed image will be weighted sums of the values of the four pixels in the original image surrounding the sub-location. The weight of each value depends on the distance from the sub-location to the pixel, and the angle between them (Burdick, 1997, p.135).

2.10 Connectivity

When analyzing a binary image, the various aspects of the different pixel clusters or *objects*, are investigated. An object is normally defined as a cluster of white pixels. To decide whether two pixels belong to the same object or not, the connectivity has to be defined. Normally when speaking of connectivity, it is the two connectivities *4-connection* and *8-connection*, that are referred to. If 4-connection is used, it means that two pixels belong to the same object if they are connected on one of the four sides; above, below, to the right or to the left. The 8-connection allows pixels to be connected in the diagonal directions as well (Burdick, 1997, p.154-155).

In Figure 2.7 there are 6 or 4 objects, counting using 4-connection or 8-connection, respectively. With 4-connection, the four pixels at the right of the image will be regarded as four different objects, while using 8-connection the same four pixels are regarded as two objects. Throughout the thesis,

8-connection will be used for connectivity, if otherwise is not specified.

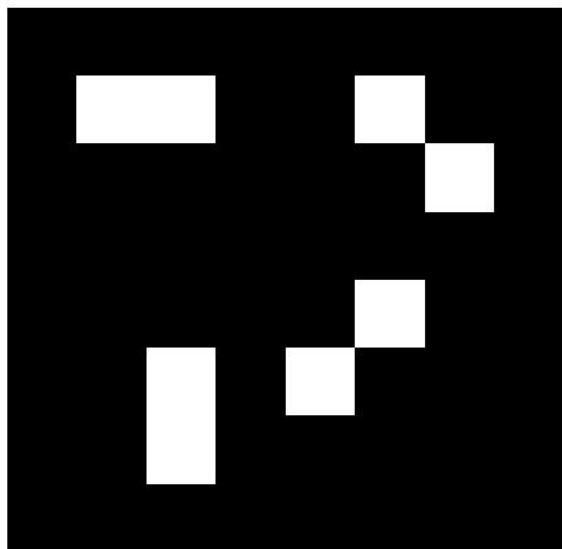


Figure 2.7: Illustrating connectivity.

Chapter 3

Pre-processing

Before the segmentation algorithm can be applied, a rather extensive pre-processing is needed. All parameters are chosen empirically based on the training sample of 68 images. All images used to exemplify the pre-processing, the removal of artifacts and the segmentation algorithm are taken from the training sample.

3.1 Masking

All the images contains a dark surrounding and a disk-shaped illuminated center, as shown in Figure 3.1. It is the illuminated center that is of interest. The surrounding is not entirely black, so to discriminate between the surrounding and the disk, a black mask with a disk-shaped hole is imposed on the image. At the same time, the image is cropped to a square, as in Figure 3.2. The mask covers the entire dark area plus the outer, darkest, illuminated area. This means that not the whole image is taken into consideration, but the masking is necessary to avoid false segmentation. With no black mask, the segmentation algorithm will just distinguish between the dark and the light area, and the lesion will remain undetected. The radius should be as large as possible, to be sure to include the whole lesion, but small enough to leave out the darkest areas. The optimal radius was found to be 800 pixels. It can be seen in Figure 3.7 that after 800 pixels distance from the center of the image, the luminance decays rapidly, and the quality is therefore poor. Unfortunately, the illuminated disk is not necessarily centered, but this is dealt with in Section 3.4.1.



Figure 3.1: Illuminated center and near-black surround (2400×3276 pixels).

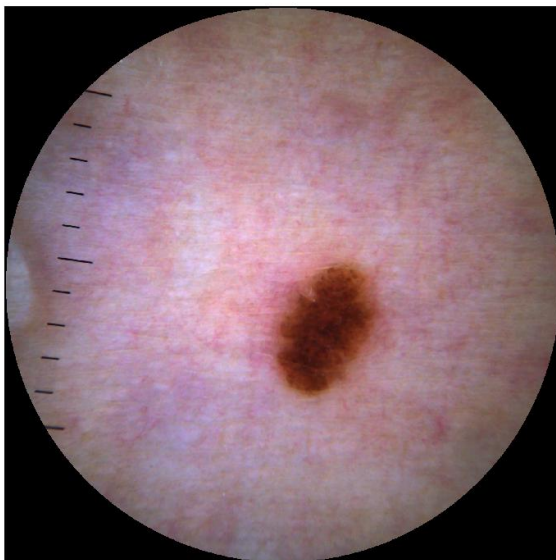


Figure 3.2: The image with a superimposed black mask.

3.2 Filtering

For noise-removing, a median filter was used. The neighborhood was selected to be a 5×5 pixel square. This is a very small neighborhood compared to the size of the images, which are at the end 1600×1600 pixels. In comparison uses Schmid (1999a) a 9×9 pixel neighborhood on his 480×480 pixel images. As pointed out by Hance et al. (1996), the median filter is computer intensive. With modern computers, this is no longer a crucial issue for images of this size. By applying the median filter to the 8 bit pixel arrays, before they are converted into *double*, filtering the three color layers takes 7-8 seconds (mean time for training sample). For 16 bit pixel arrays, the mean running time is 12 seconds. Compared to the overall running time for the whole segmentation program (about 6 minutes), this is fully acceptable.

The purpose of the filtering is to remove noise while preserving image details. The mean absolute differences for the images in the training set before and after filtering lie between 0.27 and 0.71 for the red band, between 0.14 and 0.49 for the green band, and between 0.20 and 0.70 for the blue band. The maximum absolute differences lie between 9 and 127 for the red band, between 6 and 49 for the green band, and between 7 and 95 for the blue band in the training-set images. This is larger differences than reported by Lee (2001, p.47), using a multi-stage median filter.

3.3 Color Representation

The images are so-called truecolor images, meaning that the three layers of the image represents the three basic colors; red, green and blue. The color standard used is sRGB, which is a standard RGB color space. D65 is used as white point. The RGB color space is not uniform, as opposed to for example $L^*u^*v^*$. Selecting the color space is one of the first crucial choices to be made. Every color space has its advantages and disadvantages. Schmid (1999c, p.167) shows in his thesis that the RGB space gives higher proportion of total variance explained by the two first principal components (see Section 2.6.2) than $L^*u^*v^*$. The principal component transformation also creates a color space, if all three components are used. This is an image-dependent color space, as opposed to RGB, $L^*u^*v^*$ and most other color

spaces.

3.3.1 Gray-conversion

For the image analysis, the 3-D color image is converted to a grayscale image using principal component transformation (see Section 2.6.2). This method minimizes the proportion of total variance unexplained when the new 2-D image is a result of a weighted sum of the three color layers, where the weights are computed individually for each image. Proportion of total variance explained lies typically above 0.80 (61, 90%). The smallest proportion of variance explained in the training set was 0.70. It is the images with light lesions that have the lowest proportion of total variance explained by the first principal component. Gómez et al. (2008) point out that the direction of the principal components is influenced under the presence of hair. In this thesis, the PCT-method is used for gray-conversion after the hairs have been removed.

When it comes to analyzing the luminance of the image, a weighted sum of the three layers with fixed weights is used.

When converting a three-layer color image to a one-layer grayscale image by the PCT-method, only the pixels that will be used in further analysis are taken into consideration when deciding the weights. Therefore the pixels belonging to the black surround will not be used in the principal component transform. Including the near-black pixels in the surround will result in a lower proportion of total variance explained by the first principal component.

3.4 Illumination

A great challenge for the segmentation is the non-uniform and non-central illumination. The image is brighter near the center and darker near the edges, but unfortunately the strongest illuminated area of the image is not necessarily at the center of the image, and it does not have the same location from image to image. This is due to the angle between the dermoscope and the skin. When the dermoscope is not held perpendicular to the skin, the strongest illuminated area of the image will not be at the center of the image. Non-perpendicular angle between dermoscope and skin also results in non-circular illumination. Left unattended, the non-uniform illumination results in unsatisfying lesion-segmentation. When the lesion is light brown,

the segmentation algorithm fails to recognize it, since it is no darker than the poorest illuminated areas of skin.

3.4.1 Centering the Illuminated Disk

The illuminated disk, as seen in Figure 3.1, is not necessarily situated at the center of the 2460×3276 pixel image. The disk is not off-centered more than approximately 40 to 100 pixels in each direction, but for a correct analysis of the luminance, centering the disk is necessary. This is done by first cutting a 2400×2400 pixel square from the center of the original image. This new square image contains the entire illuminated disk. The RGB image is converted to a grayscale image using the RGB-to-luminance transform. The 0.5-quantile of the grayscale image is calculated. The grayscale image is converted into a binary image, using the 0.5-quantile as threshold. This quantile is chosen on the basis of the 68 images in the training sample. For all images, this quantile finds the entire disk. The binary image shows the illuminated disk as 1's and the dark surroundings as 0's, as seen in Figure 3.3(a). The largest object in the image is found (the disk), and all

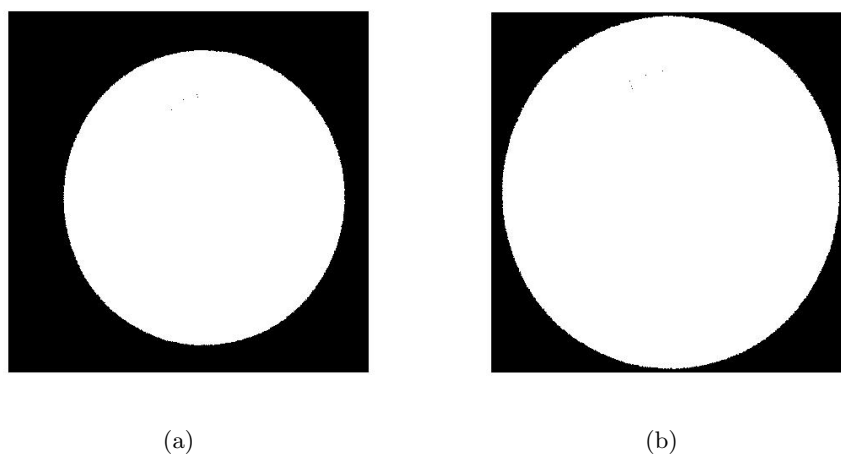


Figure 3.3: (a) Off-centered illuminated disk in a 2400×2400 pixel square. (b) The new 2000×2000 matrix with centered illuminated disk.

other objects are removed (noise). The center of the square made up by the leftmost, rightmost, uppermost and lowermost pixel of value 1 is found. The present 2400×2400 pixel image is cropped. The new image is 2000×2000 pixels of size, and the center coincides with the center found in the binary

image, as seen Figure 3.3(b). This new image contains the entire illuminated disk.

3.4.2 Center of Illumination

The brightest illuminated area of each image must be found. The brightest illuminated area may fall on skin or on the lesion. In the latter case, the brightest illumination will not lead to the brightest spot on the image. The 2000×2000 pixel color image is converted into a grayscale image using the RGB-to-luminance transform, and the 0.75-quantile is calculated. Then a binary image with the 0.75-quantile as a threshold is created. The 0.75-quantile is chosen on the basis of the 68 images from the training sample. A lower quantile will result in more images where the entire lesion is not covered, a higher quantile will result in more images where the outer boundary of the illuminated disk is reached. For the training sample, the 0.75-quantile covered the entire lesion, but did not reach the borders of the illuminated disk in 57 (84 %) of the images. A morphological opening operation with a 5×5 square as structuring element is applied to the binary image to avoid outlying pixels of value 1. The center of the square made up by the leftmost, rightmost, uppermost and lowermost pixel of value 1 is found. This will, in

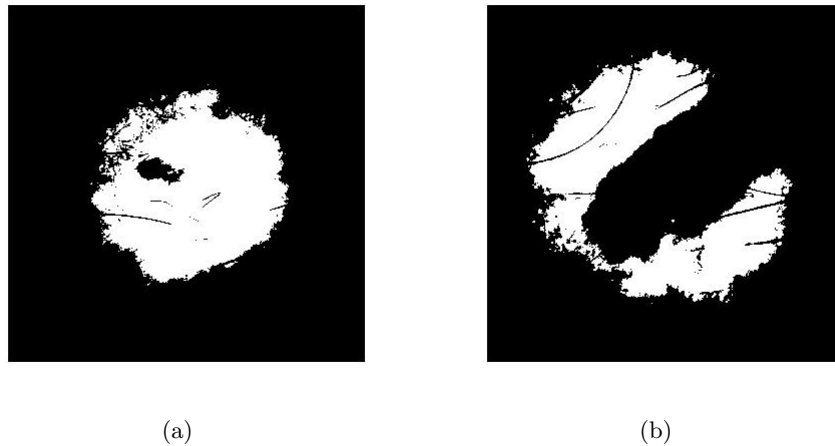


Figure 3.4: Binary image with the 0.75-quantile as threshold (a) surrounding the lesion (b) not surrounding the lesion (2000×2000 pixels).

most cases, be close to the center of illumination. Even in the few cases where this method fails to find the approximate center of illumination, it

does more good than damage and will in any case help significantly with the segmentation. As seen in Figure 3.4(a) the 0.75-quantile covers an area big enough to contain the whole lesion, but small enough not to exceed the borders of the illuminated disk. If the lesion is very big, situated near the boundaries of the disk, or the dermoscope is very tilted, the area defined by the 0.75-quantile may not contain the whole lesion, or it may exceed the borders of the illuminated disk, as seen in Figure 3.4(b).

Although the center of illumination may not be correct in the case of Figure 3.4(b), and the chosen correction matrix may not be the optimal one, illumination correction results in a great improvement for segmentation, as seen in Figure 3.12. When the center of illumination has been found, a 1700×1700 pixel square image is cut from the center of the 2000×2000 pixel image, independently of the center of illumination. The reason that the center of illumination is calculated from the 2000×2000 pixel image, is in case the 0.75-quantile exceeds the 850-pixel radius disk. In that case, the center of illumination would not have been found with only the 1700×1700 square.

3.4.3 Illumination Correction Matrix

To correct the non-uniform illumination, an illumination correcting matrix has been made. This matrix is made on the basis of one of the images with near-central illumination and homogeneous skin color, as seen in Figure 3.5.

First the RGB image is converted into a grayscale image using the RGB-to-luminance transform. Then the center of illumination is found, using the same methods as in Section 3.4.2. A 2000×2000 pixel matrix with center coinciding with the center of illumination is cropped from the original image. This is the matrix seen in Figure 3.6(a). The central disk of radius $r = 1000$ is then used to create a new grayscale image, but now the conversion is done by principal component transformation, as seen in Figure 3.6(b).

The illumination correction matrix is supposed to reverse the non-homogeneous illumination effect. Based on the right part of Figure 3.6(b), where there is only skin, a graph illustrating the mean illumination as a function of distance from the center of the image is made, and can be seen in Figure 3.7. The graph is made by calculating the mean value of all pixels in a certain distance from the center of the image. Near the center the graph is very irregular, due to the small number of pixels in a short distance



Figure 3.5: Small skin lesion surrounded by skin with homogeneous color (2400 × 2400 pixels).

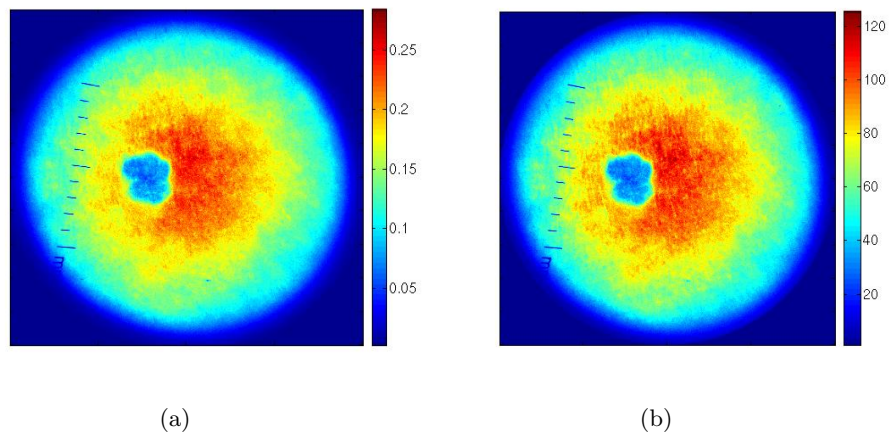


Figure 3.6: Grayscale image based on (a) RGB-to-luminance and (b) PCT (2000 × 2000 pixels).

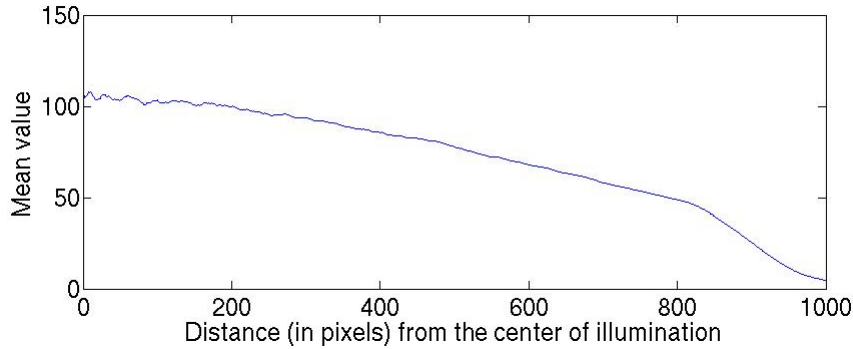


Figure 3.7: Mean illumination as a function of the distance from the center of the image.

from the center. After about 200 pixels from the center of the image the graph decays with an approximately constant value until it reaches about 800 pixels from the center. From here on and out it decays more rapidly. All values from 1 to 200 are set to the mean of the values from 1 to 200. The rest of the graph is smoothed with a 11 pixel long window. The graph is then normalized, such that the values from 1 to 200 are set to 1, and then the graph is reversed, as seen in Figure 3.8. The rapid decay from 800 pixels from the center and out is not wanted in the correction matrix, since it represents the outer boundary of the illuminated disk. This outer boundary will not be present in the 1600×1600 pixel images that is used for image segmentation. A new graph is therefore made based on interpolation of the existing graph. The interpolation polynomial in Newton form is used.

The new graph is based on three separate graphs. From 1 to 200 the value is set to 1. From 201 to 700 the value is set to the newtonian interpolation based on the values of $x_0 = 201, x_1 = 400$ and $x_2 = 700, y_j = f(x_j), j = 0, 1, 2$. From 701 to 1000 the values are set to the newtonian interpolation based on the values of $x_0 = 701, x_1 = 750$ and $x_2 = 800, y_j = f(x_j), j = 0, 1, 2$. The resulting graph can be seen in Figure 3.8.

A 2000×2000 correction matrix is made based on the interpolated graph. The value of each pixel at a certain distance from the center of the matrix, equals the value at the same location on the interpolated graph, as seen in Figure 3.9.

This matrix reverses the unwanted effect of non-central and non-uniform illumination by preserving the values of the pixels in the brightest illu-

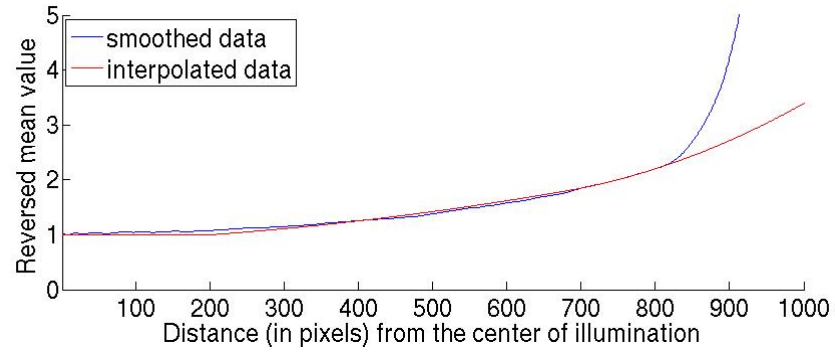


Figure 3.8: Smoothed graph (blue) and interpolated graph (red).

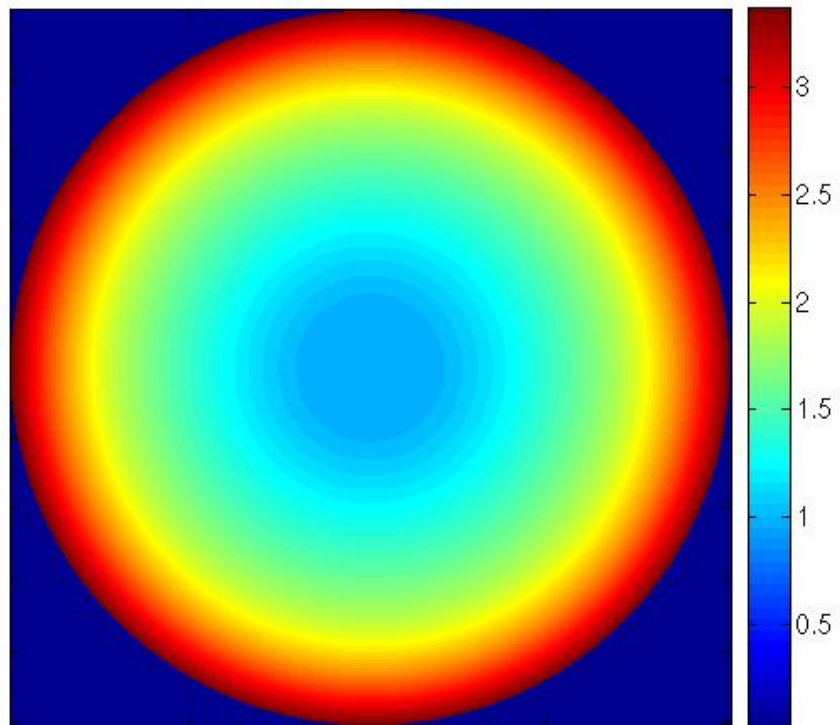


Figure 3.9: The general correction matrix (2000×2000 pixels).

minated area, while increasing the pixel-values in the poorer illuminated areas. When the center of illumination of an image has been found, the image-adjusted correction matrix is made. The general correction matrix is cropped to a 1700×1700 matrix. The matrix is cropped so that the center of illumination of the image coincides with the center of the disk of ones in the correction matrix, as seen in Figure 3.10.

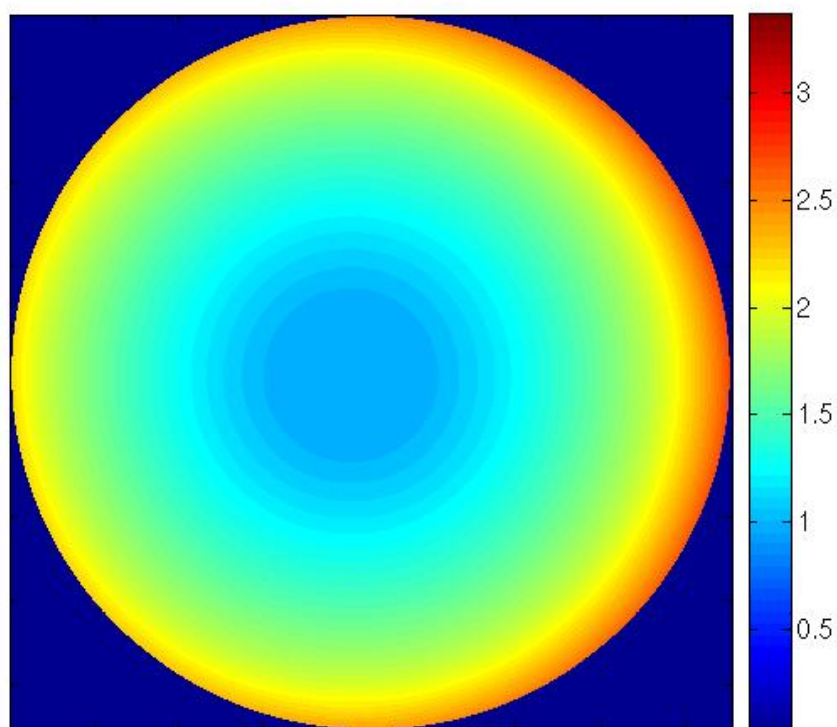


Figure 3.10: The correction matrix for the image in Figure 3.11 (1700×1700 pixels).

Figure 3.11 shows an image before and after scalar multiplication with the correction matrix.

Figure 3.13 shows the segmentation of an image with a light brown lesion. To the left the segmentation is done without any correction for non-uniform illumination. To the right the same segmentation algorithm is applied to the image after illumination-correction.

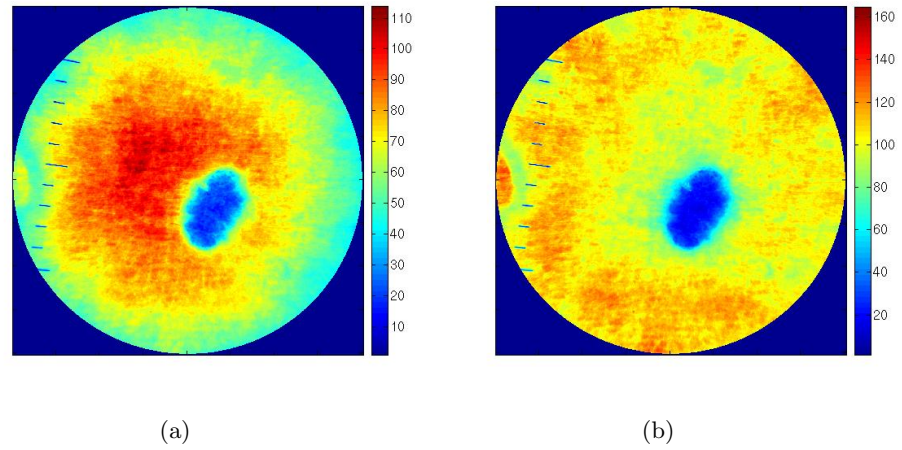


Figure 3.11: The image (a) before and (b) after scalar multiplication with correction matrix.

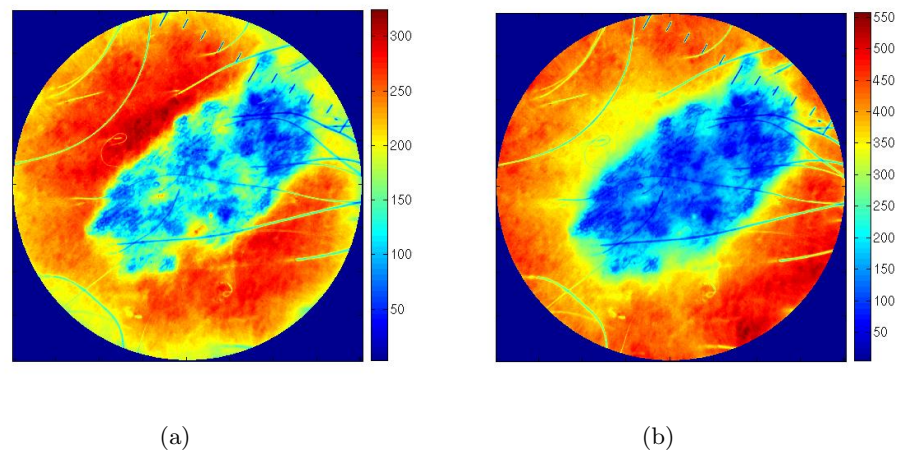


Figure 3.12: The image (a) before and (b) after scalar multiplication with a possibly incorrect correction matrix.

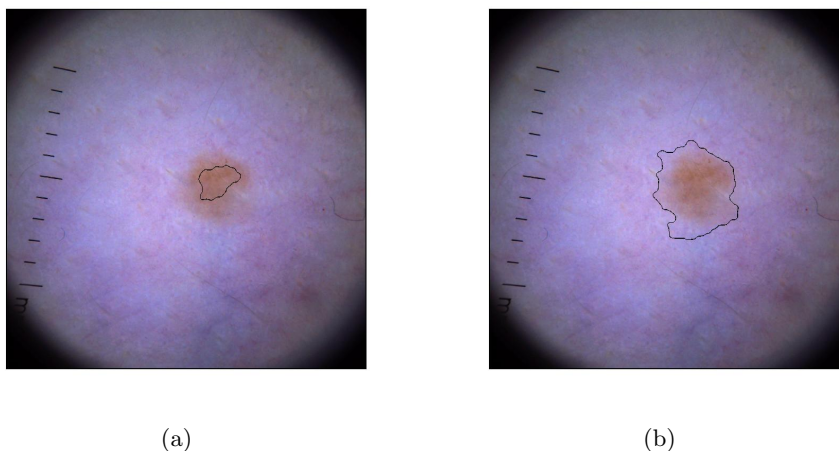


Figure 3.13: Segmentation (a) without and (b) with illumination-correction.

Basing the correction matrix on only half an image may seem a little optimistic, or unwise. While the other parameters in the pre-processing are tuned on the basis of the various results of all the 68 images in the training sample, the correction matrix is based on quite a small sample. Making a different correction matrix, taking more images into consideration was proposed. But, as the current correction matrix shows such satisfying performance, the proposal was left unexplored. The points of interpolation are parameters tuned according to all the 68 images.

The problem with non-circular illumination is left unattended, since the present circular correction matrix is sufficient for the non-uniform illumination not to influence the segmentation.

Chapter 4

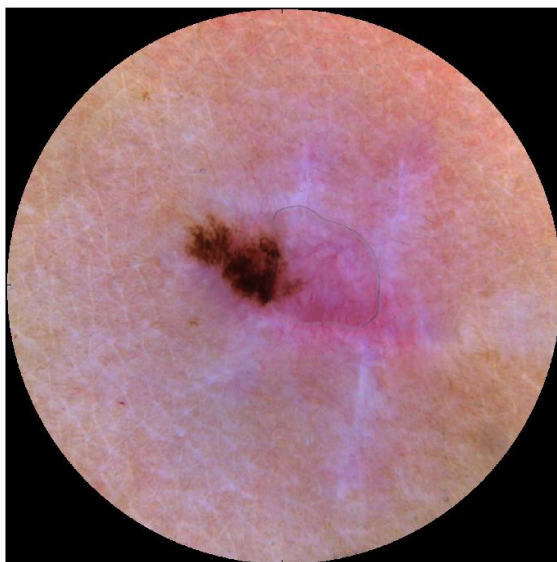
Removal of Artifacts

All algorithms for segmentation of skin lesions need to pay extra attention to the removal of artifacts, either in pre- or post-processing, or during the segmentation itself. Artifacts are in this case, anything present in the image that is not skin or lesion. This might be hairs, air bubbles, scales, or felt-pen marks drawn on the skin by the patient or the physician. Among these, the presence of hairs is the biggest obstacle for correct segmentation for most algorithms. This is because the color of hairs in most cases is very close to the color of the skin lesion. Left unattended, the hairs will be classified as lesion and lead to incorrect segmentation.

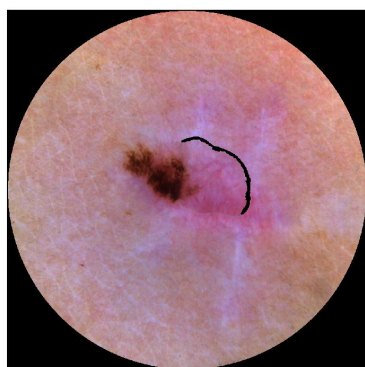
Both the scale-removal and the hair-removal algorithms are based on 1700×1700 pixel images. The final segmentation algorithm uses a 1600×1600 pixel image, but to be able to remove artifacts in contact with the image boundary, that would otherwise be regarded as noise and not removed because of their short length, it is necessary to use a slightly bigger image for artifact removal.

In addition to hairs, the presence of scales and air bubbles may lead to incorrect segmentation. The scales are removed in a separate scale removing algorithm, while the air bubbles usually do not affect the segmentation. The presence of an air bubble changes the color of the skin/lesion underneath. But this color-change is not radical. The air bubbles are usually quite small, and will therefore not affect the shape of the histogram used in the segmentation algorithm. If the air-bubble covers a very large area of the lesion, it might have an effect on the segmentation, but these images are excluded from the study. If the air bubble has a very distinct edge, as seen

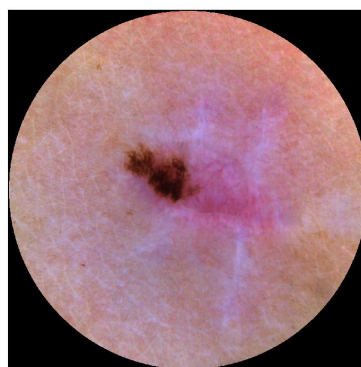
in Figure 4.1, the hair removing algorithm removes the edge because of its dark color.



(a)



(b)



(c)

Figure 4.1: (a) Air bubble with distinct borders. (b) The borders detected by the hair-removing algorithm. (c) The borders replaced by skin-colored pixel values.

4.1 Scales

Scales (also referred to as graduations or rulers) are the black lines printed on the glass plate of the dermoscope to have easy access to a measurement of the size of the lesion. The scales are clearly seen in the upper part of Figure 4.2(a). For automatic segmentation and classification the scales have no purpose since all the images are taken with the same magnification, and have the same spatial resolution.

The scales and the hairs are different in several ways. The scales are often darker than the hairs, they are shorter and they are straight. Therefore the scales and the hairs are removed separately. First the scales are removed (as seen in Figure 4.2) by the following algorithm:

- The three-layer image has to be reduced to a one-layer image. The RGB-to-luminance transformation was chosen. The scales are so dark, that the luminance is what differs the scale pixels the most from the rest of the pixels in the image.
- The detection of scales is done separately for the vertical direction and the horizontal direction. The scales are not situated at the same place in every image, so to make sure they are detected, the image is rotated 0, 30 and 60 degrees. The rotation is done by using inverse transformation and nearest neighbor. Rotating three times proved to be sufficient for the 68 images in the training sample. The resulting images from the three rotations are combined at the end of the procedure. This means that if scale objects were found and replaced in more than one rotation (this can happen when dark hairs are taken to be scales), all objects in all rotations are replaced by skin-valued pixels at the end.
- For each rotation, a binary image is created. The threshold level of the binary image is set to 0.4. The fixed threshold value was obtained empirically on the basis of the images in the training sample. A fixed value can be chosen since the scales are always the same. The pixels belonging to the inner disk with a 450 pixel radius are set to 1. The scales are always situated outside this disk. The resulting image can be seen Figure 4.2(b).

If the scale objects are not detected on basis of the binary image with 0.4 as threshold level, a new binary image with 0.6 as threshold level is created.

- Each black pixel in the binary image is investigated in the vertical and horizontal direction. For the vertical direction, the procedure is as follows: For each black pixel, the 10 pixels to the right and the 10 pixels to the left are checked. If there is one or more white pixels both to the right and the left, this pixel is considered to possibly belong to a scale. Then the 10 pixels above and the 10 pixels below are checked. If all the pixels above or all the pixels below are black, the current pixel is temporarily stored as a scale pixel. This way we find all the pixels in the vertical direction belonging to an object that is thin enough and long enough to be part of a scale. The code used in MatLab is as follows:

```
if (sum(bw(i,t(j)-10:t(j))) > 0 & ...
    ... sum(bw(i,t(j):t(j)+10)) > 0)
    if sum(bw(i-10:i,t(j)))==0 | sum(bw(i:i+10,t(j)))==0
        bb_v(i,t(j)) = 1;
    end
end
```

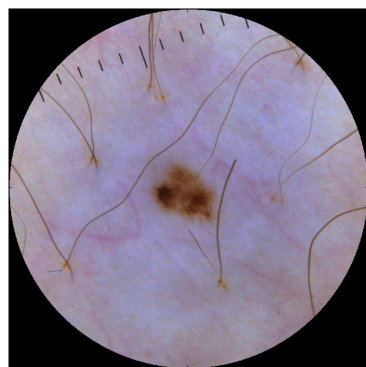
where `bw` is the binary image, `i` is the row number, `t(j)` is the column address of the black pixel and `bb_v` is the matrix where possible scale pixels in the vertical direction are stored. The same procedure is followed to detect scales in the horizontal direction, only the direction is changed:

```
if sum(bw(i-10:i,t(j))) > 0 & sum(bw(i:i+10,t(j))) > 0
    if (sum(bw(i,t(j)-10:t(j)))==0 | ...
        ... sum(bw(i,t(j):t(j)+10))==0)
        bb_h(i,t(j)) = 1;
    end
end.
```

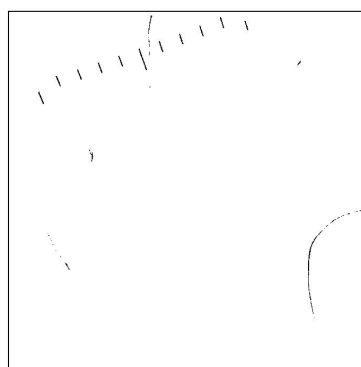
The result is two binary images, one for scales in the vertical direction, and one for scales in the horizontal direction.

- The objects are labeled, and those object not resembling scales are removed. All objects smaller than 100 pixels or bigger than 900 pixels are removed. All objects with *eccentricity* (see **Noise-removal** in Section 4.2) smaller than 0.95 are removed. The remaining objects might be the scales. It happens that dark hairs are taken for scales, and therefore detected by this algorithm. To avoid this, one last criterion remains. If the number of objects in the binary image after the removal is less than 10, which is the minimum number of scales present in each image, all objects are removed. The 10 objects also have to be situated on the same half of the image. The result is seen in Figure 4.2(c).
- A morphological dilation operation with a 7×7 pixel square as structuring element is applied to the scale objects. The resulting pixels are replaced by skin-colored or lesion-colored pixels, according to whether the scale is located on the skin or in the lesion. Each scale pixel found in the vertical direction is replaced by the value from linear interpolation of the 10th non-scale pixel to the left and the 10th non-scale pixel to the right. If the scales are found in the horizontal direction, the pixels used for interpolation are the 10th non-scale pixel above and the 10th non-scale-pixel below the current scale pixel. The image with the scale pixels replaced can be seen in Figure 4.2(d).

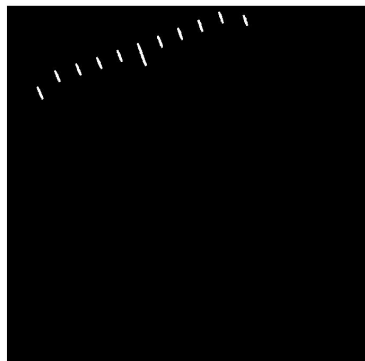
The scale removing algorithm works for all the 68 images with a few exceptions. All scales, and nothing but scales, were removed in 56 of the images. In another 7 images fractions of dark, thin hairs were removed in addition to the scales. In 4 of the images tiny areas inside the lesions were removed. But since they are replaced by the interpolation value of the neighborhood, this has microscopic effect on the segmentation, due to the small size of the replaced areas. For the last image, the algorithm did not remove all the 11 scales. The scales were situated in a very dark area (due to the lightning) and were therefore not detected.



(a)



(b)



(c)



(d)

Figure 4.2: (a) The original image. (b) The binary image. (c) Detected scale objects, after dilation. (d) Final image with scales removed.

4.2 Hairs

The presence of hair on the skin surrounding the lesion is a great obstacle for correct segmentation. The hair is of non-negligible confusion for the segmentation algorithm. Especially the presence of dark, thick hairs will lead to incorrect segmentation, but also lighter hairs if they appear near boundary of the lesion.

It is possible to remove hairs physically by using a razor (Burroni et al., 2005; Rubegni et al., 2002), but it is a point to keep the procedure for diagnosis of skin lesion as simple as possible, both for physician and patient. In digital image analysis, the simplest way to remove hairs is by filtering. Hairs can, because of their long and thin structure, be filtered out if the filter has the needed smoothing capabilities and/or the window size of the filter is big enough. Morphological operations are used to filter out the hairs in Ganster et al. (2001), Rajab et al. (2004) and Gómez et al. (2008). Morphological filtering of the image is a problem, because the shape of the structuring element effects the result (Schmid, 1999c, p.13). It is especially problematic when histogram analysis is to be performed, because it produces a kind of pre-segmented image (Schmid, 1999a). Xu et al. (1999) use a Gaussian filter for hair removal, but as mentioned in Section 2.4.1, it has poor edge-preserving capabilities. A median filter will respect the shape of the lesion more than a Gaussian or morphological filter, and is used for hair-removing purposes in Zagrouba and Barhoumi (2004), Celebi et al. (2007), Magliogannis (2003) and Zhang et al. (2000). The median filter works good for thin hairs, but when the hairs get thicker, the median filter is unable to remove the hairs. Increasing the window size will lead to over-smoothing of the image. Therefore, DullRazor-algorithm is used for thick, dark hairs in both Zagrouba and Barhoumi (2004) and Celebi et al. (2007).

The DullRazor-algorithm was introduced in Lee et al. (1997), and is used in Zagrouba and Barhoumi (2005) and Lee (2001), in addition to Zagrouba and Barhoumi (2004) and Celebi et al. (2007). The DullRazor-algorithm works as follows: Four structural elements are created, each one is an 11 pixel long line, and the orientation is 0, 45, 90 and 135 degrees. The grayscale morphological closing operation with the four structuring elements is applied to the three color bands (red, green, blue). A binary hair mask is computed based on the difference between the original image and the closed image.

If the difference is above a certain pre-defined threshold value, the current pixel is stored as a hair pixel. Each hair pixel is checked to ensure that it belongs to a long and thin structure. Finally, the hair pixels are replaced by a value calculated by interpolation of nearby pixels.

As pointed out in Fleming et al. (1998), the morphological approach is relatively slow. Fleming et al. (1998) reports a method based on Steger's line detection algorithm. To determine if a line segment belongs to a hair or not, various constraints are calculated (number of pixels in the segment, curvature and more). In Schmid (1999a) and Schmid-Saugeon et al. (2003) another method based on morphological grayscale operations is introduced. Here a spherical structuring element is used when a morphological grayscale operation is applied to the three layers of an $L^*u^*v^*$ image. A pre-segmented threshold value is used for making a binary hair mask, as in Lee et al. (1997).

The algorithm for removing hairs in this thesis has several separate steps:

1. Create one-layer image
2. Rotate
3. Create binary image
4. Identify possible hair pixels
5. Remove those pixels not belonging to hairs
6. Replace hair pixels

One-layer image

To remove unwanted objects, the three-layer color image must first be reduced to a one-layer grayscale image. Both the PCT-image and the RGB-to-luminance have been tried, as well as the three color layers separately. Out of these five options, the red color layer gave the best results. The contrast between hair and skin is large for both the red and the blue color layer, while for the green layer it is quite small. Therefore, the red or the blue layer should be used. The red color layer can be seen in Figure 4.3.

Rotation

The detection of hair is done by detecting the hairs separately in horizontal and vertical direction. The image is rotated 0, 22.5, 45 and 67.5 degrees,

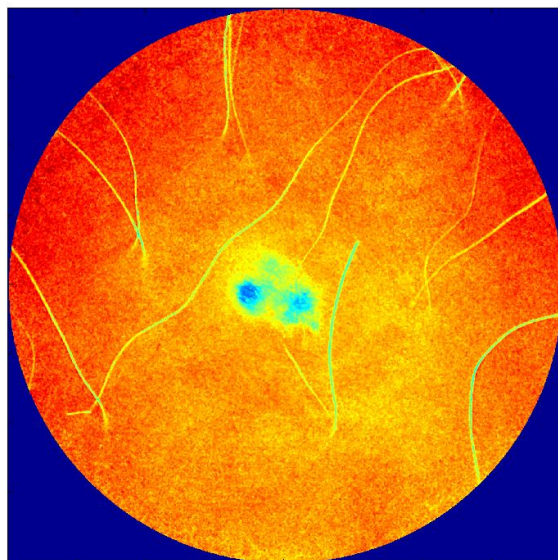


Figure 4.3: The red layer of the image.

by using the inverse transformation and nearest neighbor, as explained in Section 2.9. By examining the 68 images in the training sample, it was found that four rotations was the minimum to detect all the hairs necessary for correct segmentation.

Binary image

Five binary images are made. The threshold value is found by using Otsu's method (see Section 2.7.1). Otsu's method chooses the threshold that maximizes the between-class variance of the two classes (pixels given the value 0, pixels given the value 1). In most cases, Otsu's threshold differentiates between the lesion pixels and the skin pixels. Otsu's method is used for approximate localization of the lesion in Celebi et al. (2007). Since it is the hairs with color close to the color of the lesion that needs to be removed, Otsu's method serves the aim.

Since Otsu's method sometimes over-segments and sometimes under-segments, the five binary images are made by multiplying Otsu's threshold by 0.85, 0.90, 0.95, 1.00 and 1.05, respectively. The binary images consist of

a white background with black objects, as seen in Figure 4.4. The objects are hairs, lesions and parts thereof, and parts of the skin. The five scalars, 0.85 to 1.05, is based on the results from the 68 images of the training sample.

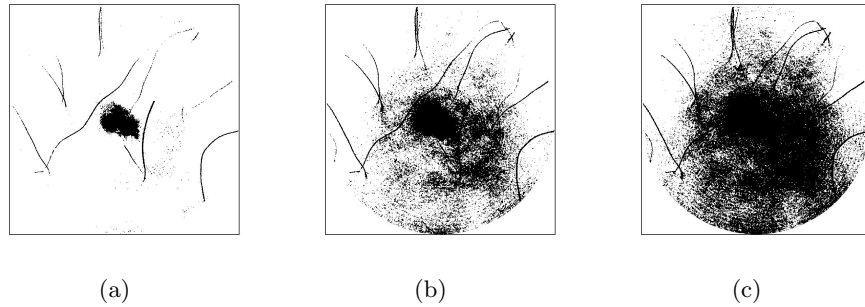


Figure 4.4: Binary images with threshold value equal to (a) 0.85 times Otsu's threshold, (b) 0.95 times Otsu's threshold, and (c) 1.05 times Otsu's threshold.

Possible hair pixels

As seen in Figure 4.4, both hairs and lesion appear as black pixels in the binary images. The black pixels belonging to hairs must therefore be differentiated from those belonging to lesion or skin. Hairs are long and thin structures, and can therefore be identified by morphological operations using structuring elements that takes advantage of the difference in the shape of the hairs and the lesion. The hairs are detected in vertical and horizontal direction.

For the vertical direction a 21 pixel long horizontal line is used as structuring element. The binary image is closed, and the result is shown in Figure 4.5(a). Compared to the original binary image in Figure 4.4(a), all black pixels with less than 10 black pixels to the right and to the left is replaced by white pixels. These pixels are considered to belong to objects that are thin enough to be hairs. The difference between the closed image and the binary image is shown in Figure 4.5(b). This image is opened with an 11 pixel long line in the vertical direction as structuring element. The result is seen in Figure 4.5(c). White pixels that are not part of a continuous vertical line, at least 11 pixel long, are replaced by black pixels. This procedure is repeated for all five binary images, and the five resulting images are

combined, and the result, seen in Figure 4.6(a) are all the pixels considered to possibly belong to a hair in the vertical direction.

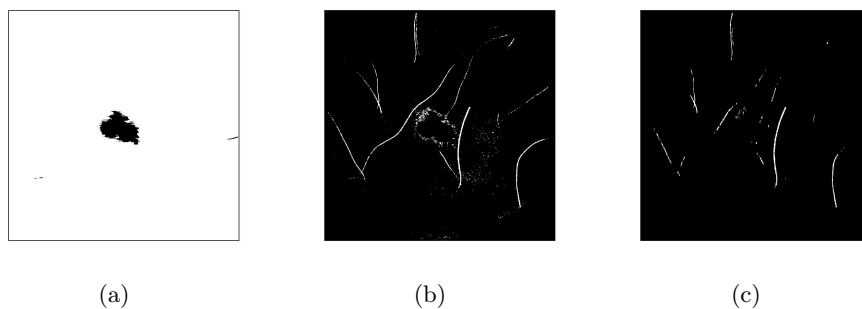


Figure 4.5: (a) The image after closing. (b) The difference between Fig. 4.4(a) and Fig. 4.5(a). (c) After opening.

The same procedure is applied to find hairs in the horizontal direction, the only difference is that the structuring element used for the closing is vertical, and the structuring element used for the opening is horizontal. There are now two binary images, one for the hairs in the vertical direction, and one for the hairs in the horizontal direction.

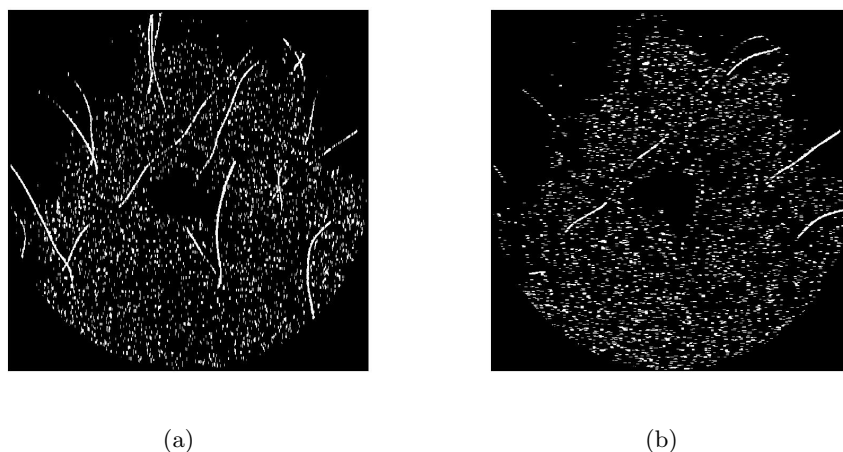


Figure 4.6: Pixels possibly belonging to hairs in (a) vertical direction, and (b) horizontal direction.

When very thick hairs are present in the image, the 21 pixel long structuring element used in the initial closing is too short to detect them. Therefore, two more binary images are made, one for thick hairs in the vertical

direction, and one for thick hairs in the horizontal direction. The procedure is as described above, but the structuring element for the initial closing is 41 pixels long, and for the final opening it is 31 pixels long. The thick hairs has to be part of a longer element than the thin hairs, because the initial opening detects a lot more than hairs when the structuring element gets longer. There are only rare cases when a thick hair is too short or too curly to be detected by a 31 pixel long structure. The possible thick hair pixels are shown in Figure 4.7.

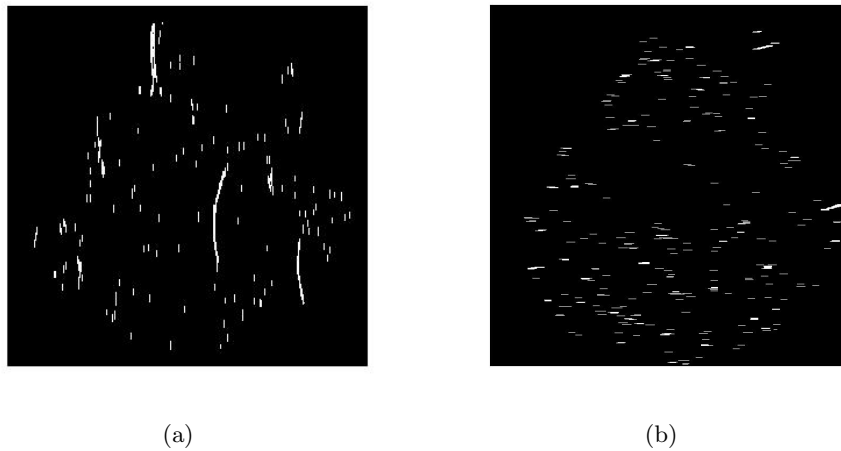


Figure 4.7: Pixels possibly belonging to thick hairs in (a) vertical direction, and (b) horizontal direction.

Noise-removal

As seen in Figure 4.6 and Figure 4.7, there is a lot of noise; small objects that are not hairs. These cluster will be identified and removed by the following procedure. In each matrix all the pixel clusters or objects are labeled. For each object the eccentricity and the length of the major axis is calculated. The eccentricity and the length of the major axis is calculated on the basis of the ellipse with the same second central moments (see Definition 16) as the object. The eccentricity of an ellipse is the ratio between the length between the foci of the ellipse, and its major axis length. An ellipse with eccentricity 0 is a perfect circle, while an ellipse with eccentricity 1 is a straight line.

For the thin hairs, the object is removed if the length of the major axis is shorter than 100 pixels. For the thick hairs, the object is removed if the

length of the major axis is shorter than 250 pixels. It is considered to be too short to be a hair. If the eccentricity of an object is less than 0.975, it is also removed, both for thin and thick hairs. It is considered not to be slim and straight enough to be a hair. The remaining pixels are believed to belong to a hair, as seen in Figure 4.8.

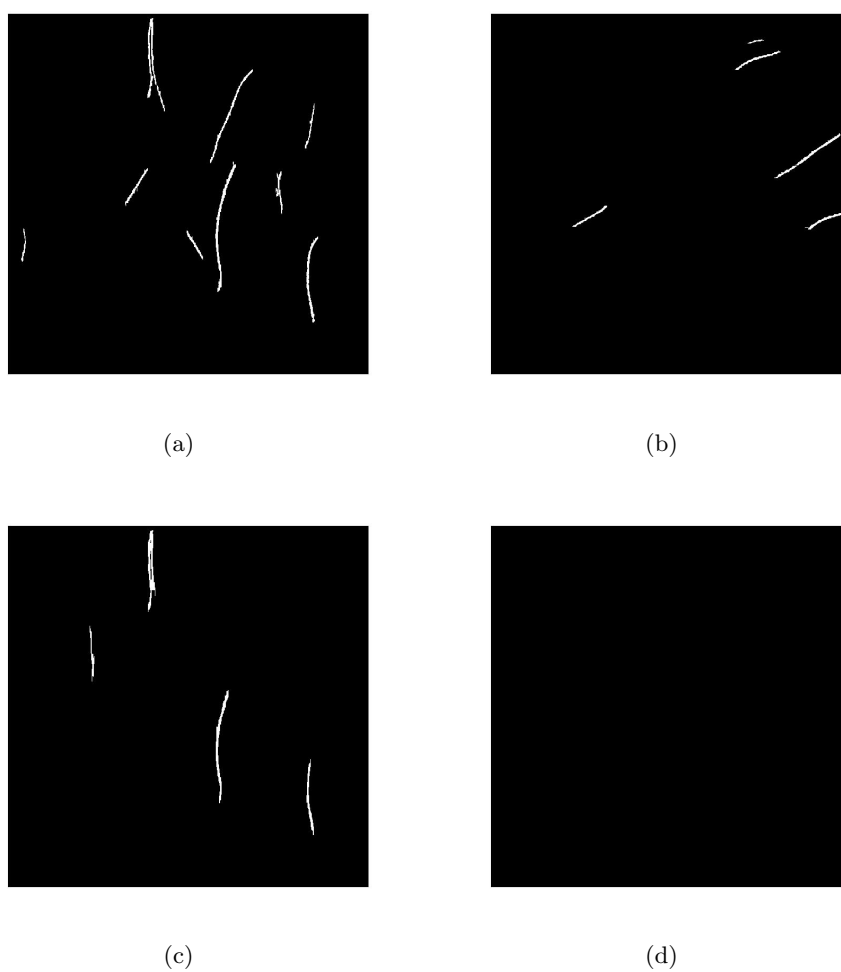


Figure 4.8: Pixels considered to belong to (a) hairs in the vertical direction, (b) hairs in the horizontal direction, (c) thick hairs in the vertical direction, and (d) thick hairs in the horizontal direction (no hairs).

Replacement of hair-pixels

The two vertical hair matrices are combined, and the two horizontal hair matrices are combined. A dilation operation with a 7×7 square as structuring element is now applied on the two hair matrices, and the result is the matrices in Figure 4.9.



Figure 4.9: Dilated hair objects in (a) vertical direction, and (b) horizontal direction.

The dilation process makes the hairs thicker and with smoother borders. This is done because most hairs create a shadow on the skin that is not detected by the hair-detection algorithm. But this shadow must be removed and replaced by some skin-colored or lesion-colored pixels, according to whether the detected hair is located on the skin or on the lesion.

These are the pixels that should be replaced. For calculating the new pixel values, linear interpolation is used. For each hair pixel in the vertical direction, the 10th non-hair pixel to the right and the left is found, and the linear interpolation is based upon their values and position. In case the dilation did not cover all the pixels in the shadow of the hair, the 10th non-hair pixel is chosen. For the pixels belonging to horizontal hair-objects, the 10th non-hair pixel found above and below are used for interpolation. The result is seen in Figure 4.11.

The above described procedure is applied on the images for all four rotations. At the end, all images are rotated back to the initial position and combined. Some of the hairs detected at one rotation or threshold,

are detected at other rotations or thresholds as well. But since the hair removal is unsupervised, it is not known beforehand which rotations and which thresholds are necessary to detect all the hairs.

In Figure 4.10 all the pixels belonging to hairs, after the dilation, are shown.

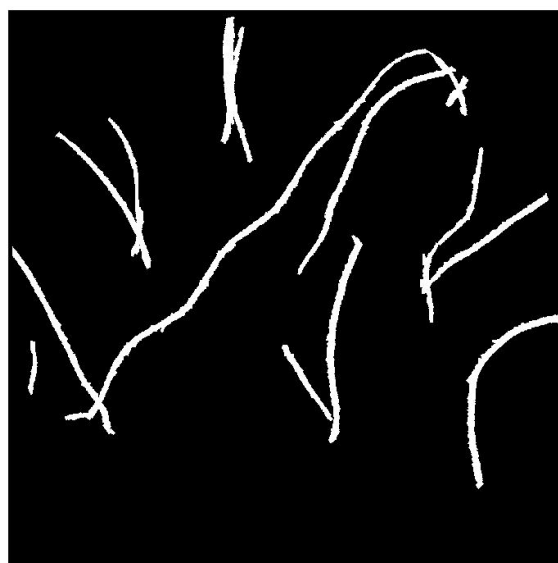


Figure 4.10: All the hair pixels.

Taking a close look on the image in Figure 4.11 it is possible to see traces of hairs. This is because the replaced pixels are more homogeneous in color than the surrounding skin. To avoid this, more sophisticated methods for calculating the new values of the hair pixels can be used. The interpolation can be more complex, using higher orders, more points and two dimensions instead of one. This will be more time-consuming than a simple linear interpolation. Since the new pixel values are meant to be used for segmentation purposes only, and not for further analysis of the interior of the lesion, the linear interpolation showed satisfactory results. Lee et al. (1997) uses interpolation to calculate the output value, while Schmid (1999b) uses the value obtained in the morphological closing operation.

After segmentation, the hair pixels inside the lesion is replaced by their original value. This means that another hair removing algorithm is needed

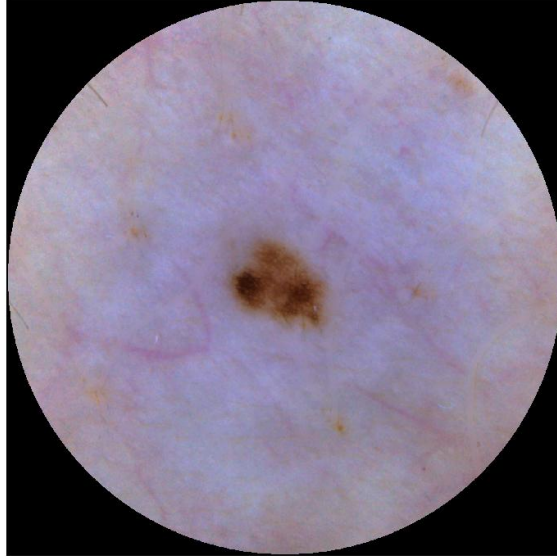


Figure 4.11: Final image with hairs and scales removed.

for removing hair inside the lesion. When replacing hair pixels inside the lesion, it is much more important that the new value is as similar as possible to the (unknown) true value. Wighton et al. (2008) proposes an algorithm for this, using *inpainting*, a method originally used to restore damaged artwork.

4.2.1 Improvement of the Hair-Removing Algorithm

The procedure can be improved by having more rotations. This is kept to a minimum to save time. When removing hair and scales for the purpose of a correct segmentation, there are two main objectives; removing the dark, thick hairs, since they will be identified as lesions by the segmentation algorithm, and removing the hairs near the border of the lesion, since they will be identified as parts of the lesion. When these two objectives are attained, further improvement of the hair removal algorithm must be carefully weighed against the time use.

One might ask whether it is necessary to use a 1700×1700 pixel image for the removal of hairs. The answer seems to be no. It is possible to downsample the image and still detect the hairs. Downsampling the images

to every third row and every third column showed promising results, and improved the computation time significantly.

Chapter 5

Segmentation of Skin Lesions

Before doing any analysis on a skin lesion, it first has to be recognized and segmented from the surrounding skin. This can of course be done manually or semi-automatically (Cheng et al., 2008; Menzies et al., 2005; Stanley et al., 2007), but it is time-consuming and subjective (see Chapter 6). There are a number of different methods for unsupervised or semi-automatic segmentation of skin lesions. The Laplacian filter, described in Burdick (1997, p.105), is used in Burroni et al. (2004, 2005), but without reports on the success rate. The JSEG-algorithm, motivated by Fisher's multiclass linear discriminant analysis (described in Johnson and Wichern (2002, p.628)), is used in Celebi et al. (2007), non-linear diffusion is used in Fleming et al. (1998) and in Schmid (1999c). Thresholding is used by Ganster et al. (2001) and by Lee (2001), the latter in combination with histogram analysis. Local thresholding in combination with histogram analysis is used by Maglioganis et al. (2006). Segmentation based on histogram analysis in combination with scale-space filtering is used by Lim and Lee (1990). Histogram analysis is used in the pre-processing in Zagrouba and Barhoumi (2004, 2005).

This thesis proposes a fully automatic lesion segmentation program based on histogram analysis. Schmid (1999c, p. 62) points out that histogram analysis is the fastest and least complex way to segment an image. Before segmentation, the image has been through an extensive pre-processing including

- Conversion from DNG to ppm
- Filtering

- Centering
- Illumination Correction
- Removal of Artifacts (scales and hair)

5.1 Color Space and Gray Conversion

The hairless three-layer RGB truecolor image is now ready to get segmented, but first a proper choice of gray-conversion must be made. It is, of course, possible to use all three color layers, doing three separated histogram analysis and merging the results together, but as shown below, the results from histogram analysis on a grayscale image is satisfying. Some methods for unsupervised segmentation of skin lesions use only the blue wavelengths (Elbaum et al., 2001), the blue channel of RGB (Ganster et al., 2001; Rajab et al., 2004) or give more weight to the blue channel of RGB (Lee, 2001, p.49). The reason for this choice is that the blue wavelengths are absorbed by the melanin in the epidermis, and a melanocytic lesion will therefore appear much darker than the surrounding skin. The drawback is that non-melanocytic lesions may not be segmented correctly. Blum et al. (2004) exclude the non-melanocytic lesions. Differentiating between melanocytic and non-melanocytic lesions can be a challenge even for trained dermatologists (Stolz et al., 2002, p.42). Menzies (1999) criticizes the exclusion of non-melanocytic lesions.

The image is converted into a one-layer grayscale image using the principal component transformation. The choice fell on the PCT-method after in addition considering the RGB-to-luminance method, and the three color layers separately. The Histogram Pursuit (HP) algorithm, presented in Gomez et al. (2007), and developed further as the Independent Histogram Pursuit (IHP) in Gómez et al. (2008), is an interesting alternative to the widely used PCT-method.

5.2 Threshold Segmentation

In threshold segmentation, the pixels in an image are divided into two or more groups based on their value. For lesion segmentation, the final goal is two groups; skin and lesion. Threshold segmentation can be divided into

two sub-groups; global thresholding and local thresholding (also referred to as adaptive thresholding). In global thresholding, the same threshold value is used for all pixels in the image, while in local thresholding, the threshold value varies over the image. The method presented in this thesis uses global thresholding. The disadvantage of global thresholding is that the spatial information is ignored. The result is that pixels clearly belonging to the skin, because of their location far from the lesion, are categorized as lesion pixels. To make up for this disadvantage, post-processing is applied to the binary image resulting from the thresholding.

5.2.1 Histogram Analysis

Histogram analysis is one way to do threshold segmentation. Several of the methods referred to earlier in this chapter, rely on the analysis of bimodal or multimodal histograms (Lim and Lee, 1990; Magliogannis et al., 2006; Zagrouba and Barhoumi, 2004; Gomez et al., 2007; Lee, 2001, p. 48). Most images of skin lesions will produce bimodal or multimodal histograms, depending on the number of bins and the smoothing, but it is important to include a method for threshold segmentation for unimodal histograms. The histogram analysis presented in this thesis gives a suggestion for segmentation when the histogram is unimodal.

The pixel values of a digital image can be considered as being observations of an underlying probability distribution. The distribution is unique for each image, and unique for each color layer. By regarding the pixel values as observations from a probability distribution, the histogram can be regarded as an approximation to the real, underlying distribution. The histograms presented here are really frequency histograms. The height of each bin represents the number of pixels belonging to this bin. For a histogram to be regarded as a probability distribution, it should be normalized so it will integrate to one (Martinez and Martinez, 2002, p.114). Normalizing the histogram will not effect the shape of the histogram or the relative height of the bins. Since the histogram analysis done in this thesis relies solely on the shape of the histogram, normalization is not performed.

Since the number of pixels (over 2 millions for each color layer), is so high, one can expect the histogram to be close to the distribution.

When an image has two clearly separated areas, one for skin and one for the lesion, and each area is more or less homogeneous, the histogram will

show two distinct peaks, as in Figure 5.1.

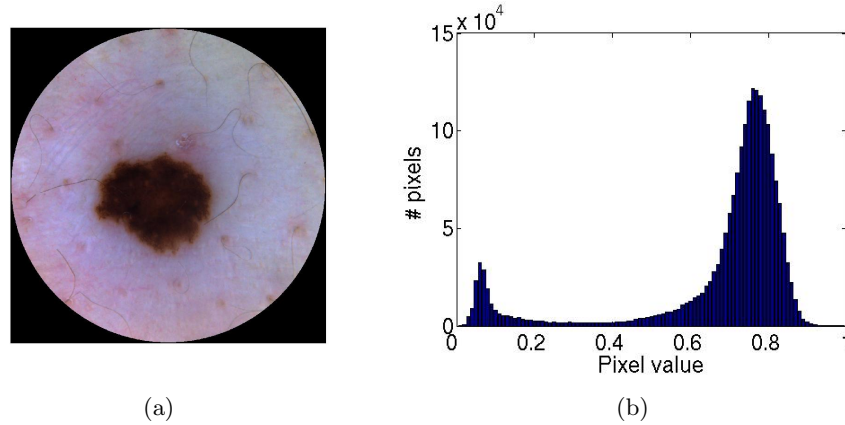


Figure 5.1: (a) Lesion and skin clearly separable by the human eye. (b) The mode representing the lesion is easily detected.

The leftmost mode represents the lesion, which consists of low pixel values, the mode to the right represents the skin. The classical histogram analysis in its simplest form is now to choose a threshold value between these two modes, often the minimum or the point in the middle of the peaks. This threshold is then used to make a binary image. All pixels below the threshold are considered to belong to the lesion, all pixels above the threshold are considered to belong to the skin. This leads to correct segmentation in the above mentioned cases, but for images where the lesion is light colored or non-homogeneous, or the skin is non-homogeneous, the mode to the left will totally disappear, as in Figure 5.2(b).

Even when the left part of the histogram in Figure 5.2 is zoomed in on, as in Figure 5.3 it is not easy to detect the lesion mode.

Number of bins and smoothing

The number of bins in the histogram is a parameter that affects the output of a histogram analysis, and there is no objective way to choose this number. The number of bins must be high enough for the histogram to show all the important details, but low enough not to overfit the sample to the distribution. The simplest and most effective thing to do is to choose one common number of bins for all images. When MatLab computes the best threshold value for converting grayscale images into binary images using

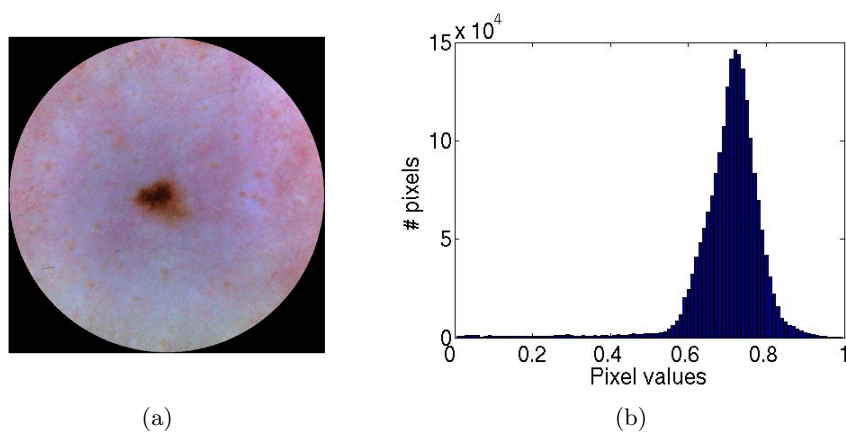


Figure 5.2: (a) Lesion and skin clearly separable by the human eye. (b) The mode representing the lesion is not easily detected.

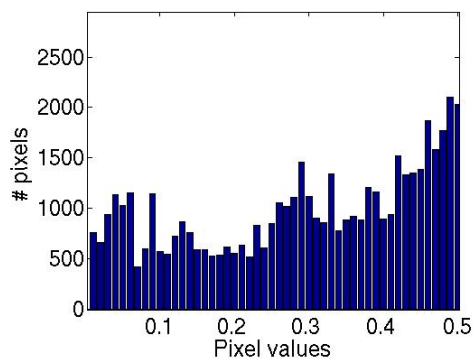


Figure 5.3: The left half of Figure 5.2(b).

Otsu's method, the number of bins is 256. This speeds up the computation, because then double numbers can be converted into the 8-bit integer class *uint8*. In the following method, the number of bins is 100, chosen on the basis of the 68 images in the training sample. The optimal histogram for threshold segmentation has, as mentioned above, two modes. For most images investigated in this thesis, 100 bins will produce two or more modes, and in very rare cases the histogram will be unimodal.

There are ways to automatically choose number of bins in a histogram. One of simplest is Sturge's Rule: If n is the number of observations in a random sample, then the number of bins, k , in the histogram is (Martinez and Martinez, 2002, p.264)

$$k = 1 + \log_2 n.$$

For the images in this thesis, Sturge's Rule gives 22 as number of bins, which is a number too low.

Scott's Rule is more elaborate, and is based not only on the number of observations, but also on the sample standard deviation (see Definition 10).

If n is the number of observations in a random sample and s is the standard deviation of the sample, then the optimal bin width, \hat{h} , is (Martinez and Martinez, 2002, p.265)

$$\hat{h} = 3.5 \cdot s \cdot n^{-1/3}.$$

For skewed data, the bin width obtained by Scott's Rule can be multiplied with a skewness factor (Martinez and Martinez, 2002, p.266);

$$\text{skewness factor} = \frac{2^{1/3}\sigma}{e^{5\sigma^2/4}(\sigma^2 + 2)^{1/3}(e^{\sigma^2} - 1)^{1/2}}.$$

Instead of letting the bin width depend on the sample standard deviation, the Freedman-Diaconis Rule lets the bin width depend on the interquartile range (IQR), defined in Definition 14.

The optimal bin width according to the Freedman-Diaconis Rule is (Martinez and Martinez, 2002, p.266)

$$\hat{h} = 2 \cdot IQR \cdot n^{-1/3}.$$

While the number of bins using Sturge's Rule is too low, using Scott's Rule, Scott's Rule multiplied by the skewness factor, or the Freedman-Diaconis Rule produces bin widths that are much too narrow, and hence too many bins. The number of bins using Scott's Rule are around 300 for the images in the training sample, while the Freedman-Diaconis Rule suggests the number of bins to be around 500 for the training sample.

Sturge's Rule, Scott's Rule and the Freedman-Diaconis Rule can not be used uncritically. There are several assumptions on the underlying distribution that have to be made (see Martinez and Martinez (2002, p.263) for details). Even if these assumptions are fulfilled, the rules are based on normal distribution, and is therefore not necessarily fit for histogram analysis of skin-lesion images.

The histogram can also be smoothed to suppress insignificant local minima and maxima. Gómez et al. (2008) and Zagrouba and Barhoumi (2004) do this by applying 15 iterations of a mean filter with a window size of 3 bins. Smoothing the histogram proved to be unnecessary for the algorithm presented in this thesis.

5.3 Segmentation Algorithm

Several segmentation algorithms based on histogram analysis focus on finding the correct skin mode and lesion mode (Gómez et al., 2008; Lee, 2001, p.48). While the skin mode is relatively easy to detect in most cases, finding the true lesion mode can be a much more difficult task. The problem is that a lesion can contain pixels with colors from a wide range, distributed in such a manner that the histogram for lesion pixels will be multimodal. Therefore, this algorithm concentrates on finding the peak and the left bound of the skin mode, and makes an estimate of the location of the lesion mode. Finally, the lower limit of the bin situated at equal distance from the estimated location of the lesion mode and the peak of the skin mode is taken as the global threshold value.

5.3.1 Skin Mode

The skin mode is found simply by finding the global maximum of the histogram. In most cases, the skin mode is much bigger than the lesion mode, but as seen in Figure 5.5(b), the lesion mode can be quite big. Therefore,

only the 75 rightmost bins are taken into consideration when searching for the skin-mode peak. Several images give histograms with skin modes with two peaks, as in Figure 5.4(b) and Figure 5.4(d).

After the peak of the skin mode has been found, the next search is for the left bound of the mode. The bound is a local minimum situated between the peak of the skin mode and the first bin. Since the skin mode can have two peaks, the search for a local minimum starts 10 bins to the left of the skin-mode peak. All pixels to the left of the bound are taken into consideration when estimating the location of the lesion mode. 10 bins is considered to be the minimum distance between the peak and bound of the skin mode. This number is found empirically on the basis of the 68 images in the training sample. The smallest distance between the peak of the skin mode and the bound was 19 bins, as seen in Figure 5.4(b).

A local minimum is regarded as the bound of the skin mode if it fulfills the criterion that the current bin is smaller than the four bins to the left and the four bins to the right. A local minimum fulfilling this criterion is found where a visual inspection would place the bound in most cases. There are cases where no bound is found because the transition between the skin mode and the rest of the histogram does not create a local minimum that fulfills the bound-criterion. This is the case for Figure 5.5(e) with the histogram shown in Figure 5.5(f). When this happens, a lower limit is created, and the minimum between the lower limit and the skin-mode peak is regarded as the bound. The lower limit must be present, or else the search for a minimum will in most cases result in bin number one. The lower limit is found in two steps. First, the cumulative sum of the number of pixels in the bins is calculated. When the cumulative sum has reached 0.1% of the total number of pixels in the histogram, the search for a peak begins. The reason for not starting the search for a peak at the first bin is to avoid regarding small fluctuations in the bottom of the intensity range as significant peaks. A local maximum is considered to be a peak if the two bins to the right of the current bin are smaller than the current bin. The first peak will be the lower limit. If such a peak is not found, the histogram is considered unimodal (see Section 5.3.3), as seen in Figure 5.5(d).

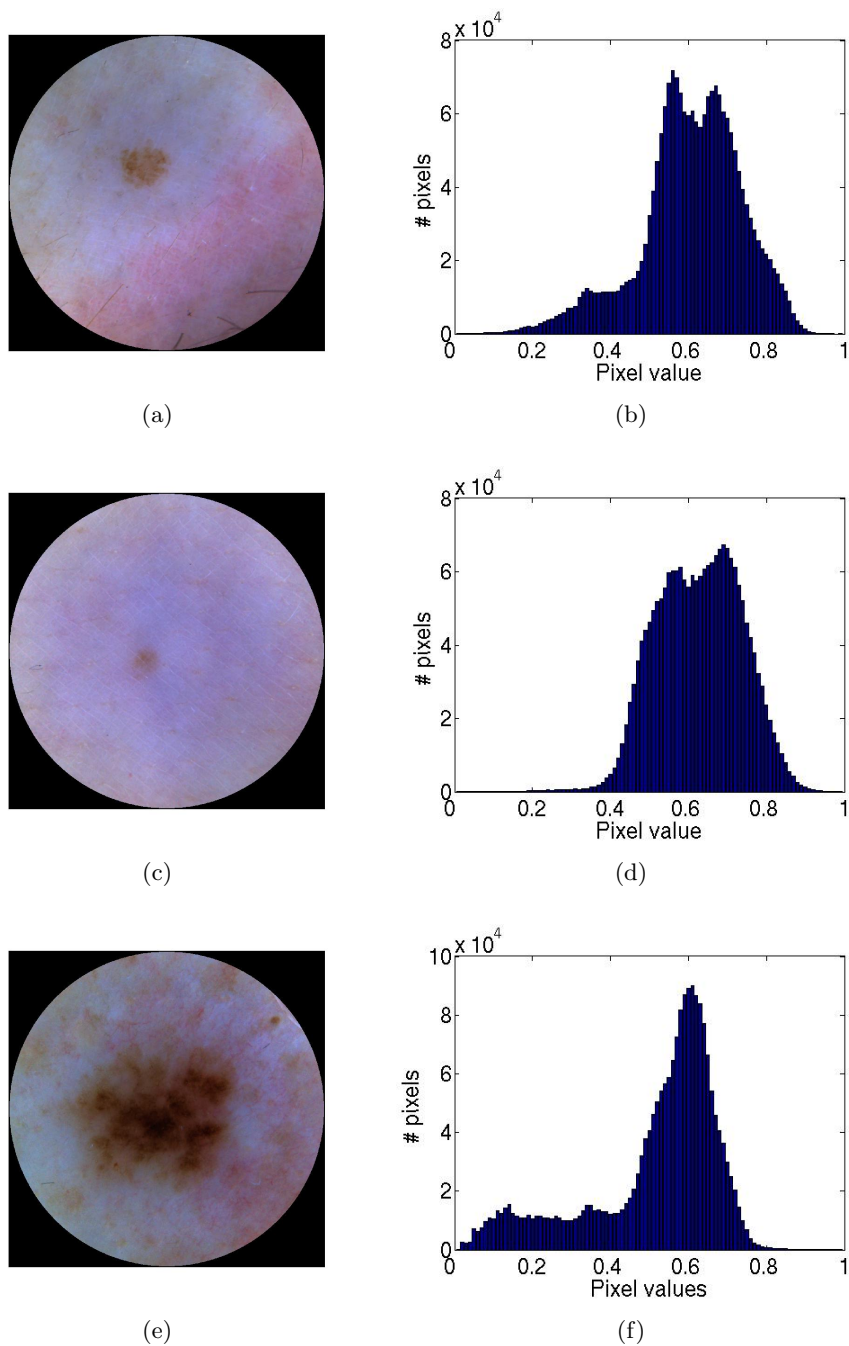


Figure 5.4: Images challenging for histogram analysis. (a)-(b) The global maximum and the bound of the skin mode is no more than 19 bins apart. (c)-(d) The skin mode has a double peak, where the peak to the right is the global maximum. (e)-(f) Multiple lesion modes.

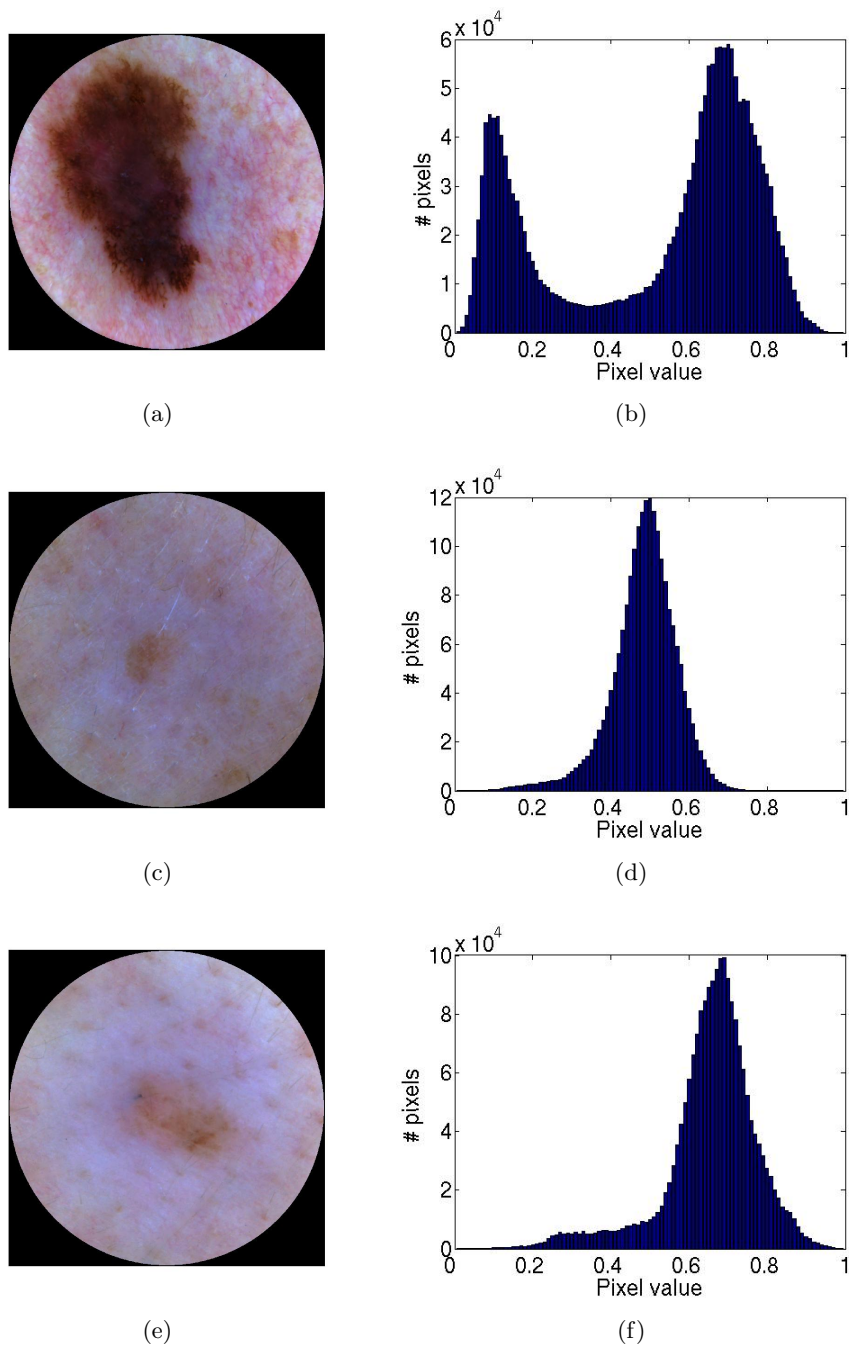


Figure 5.5: Images challenging for histogram analysis. (a)-(b) The lesion mode peak is nearly as high as the skin mode peak. (c)-(d) Unimodal histogram. (e)-(f) No obvious lesion mode.

5.3.2 Lesion Mode

As already pointed out, the lesion might consist of several modes. Instead of searching for the biggest mode (Lee, 2001, p.48), or smoothing the histogram until a pre-set number of modes has appeared (Gomez et al., 2007; Gómez et al., 2008; Zagrouba and Barhoumi, 2004), the mean of the pixels believed to be lesion pixels is estimated. The lesion pixels are all pixels to the left of the bound of the skin mode.

Regarding the pixel values of the grayscale image situated to the left of the bound in the histogram as samples from an underlying (unknown) distribution, an estimator for the expected value of this distribution can be calculated.

Estimator

An estimator is a function of the random variables X_1, X_2, \dots, X_n . A point estimator is a function with a scalar output. Choosing the right estimator for the expected value of an unknown distribution might seem like a shot in the dark, but with a few quite mild assumptions on the unknown distribution, the task gets a lot easier. In fact, the only assumptions needed are that both mean and variance exist, and that the variance is finite.

An estimator for the mean is the first method of moments estimator, μ'_1 . This estimator is found by equating the first sample moment and the first population moment (Definition 17);

$$m_1 = \mu'_1$$

$$\frac{1}{n} \sum_{i=1}^n X_i = EX = \mu_X.$$

The best estimator for a parameter is of course the estimator closest to the parameter. Since the parameter is unknown, some other criterion has to be used for finding a good estimator. There is not one objective criterion, but a widely used one is that the parameter should be *unbiased*.

Definition 22 (Bias). *The bias of a point estimator $\hat{\theta}$ of a parameter θ is the difference between the expected value of $\hat{\theta}$ and θ :*

$$Bias(\hat{\theta}) = E\hat{\theta} - \theta. \quad (5.3.1)$$

The first method of moments estimator is unbiased since

$$E\mu'_1 = E\left(\frac{1}{n} \sum_{i=1}^n X_i\right) = \frac{1}{n} E\left(\sum_{i=1}^n X_i\right) = \frac{1}{n} n EX_1 = \mu_X,$$

(for proof, see Casella and Berger (2002)) and the bias is then

$$Bias(\mu'_1) = E\mu'_1 - \mu = \mu - \mu = 0.$$

The pixel values come from a sample truncated at value of the skin-lesion bound. By making additional assumptions on the distribution, for example that the distribution is symmetrical, the effect of the truncation could be taken into consideration when estimating the expected value. By investigating the histograms in Figure 5.4 and Figure 5.5, it is difficult to find any assumptions that would be valid in all (or nearly all) cases.

5.3.3 Unimodal histograms

If an image produces a unimodal histogram (as defined at the end of Section 5.3.1), there is no obvious way to estimate the mean of the lesion mode or the whereabouts of the skin-mode bound. For unimodal histograms, the location of skin-mode bound is set to 1/3 of the distance between the first bin and the skin-mode peak. The mean of the lesion mode is estimated as above. The reason for choosing 1/3 is based on the location of the skin-mode bound and the skin-mode peak in the multimodal histograms of the training sample.

5.3.4 Global Thresholding

Before making the grayscale image into a binary image, a morphological opening operation with a structuring element of the shape of a disk with a radius of 20 pixels is applied to the grayscale image. This is necessary for correct segmentation of skin lesions that have honeycomb-like network, as seen in Figure 5.6.

After finding the location of the skin mode (the global maximum), and the location of the lesion mode (the population mean), the global threshold value is the lower limit of the bin situated at equal distance from the lesion mode and the skin-mode peak. The normalized grayscale image, the source

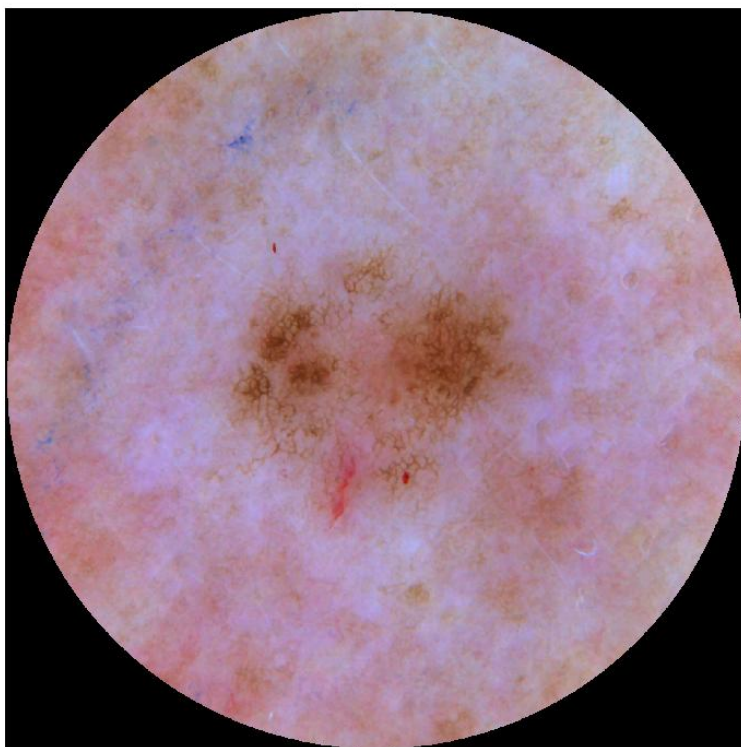
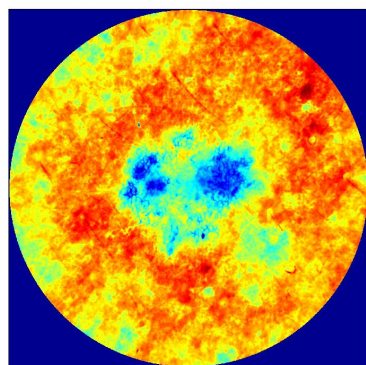
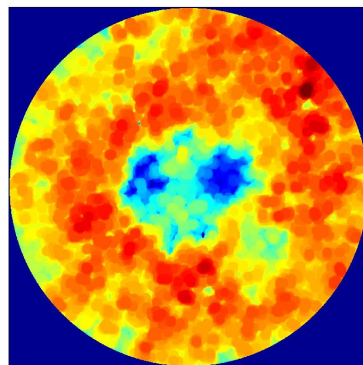


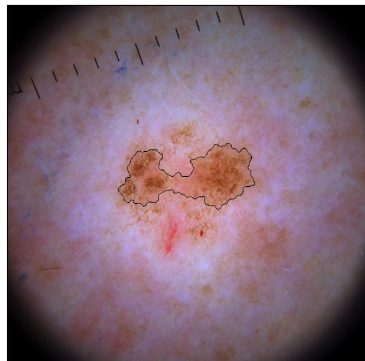
Figure 5.6: Lesion with honeycomb-like network.



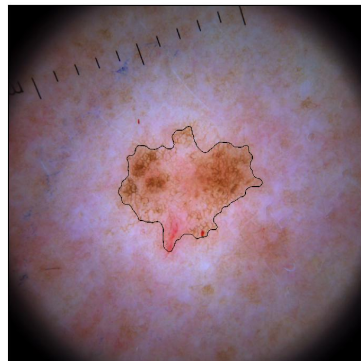
(a)



(b)



(c)



(d)

Figure 5.7: The first principal component of the image in Figure 5.6 (a) before and (b) after opening. Segmentation (c) without and (d) with opening.

of the histogram, is now made into a binary image, by using the calculated threshold. The global threshold value lies in most cases to the left of the skin-mode bound. One might think that the threshold value therefore is too high. There are pixels belonging to the lesion that are light enough to resemble skin pixels, and pixels belonging to the skin that are dark enough to resemble lesion pixels. This means that the bound of the lesion mode (if present) lies to the right of the bound of the skin mode. By choosing a the global threshold value as done here, pixels belonging to the skin will appear in the lesion group (black pixels), but will be removed in the post-processing (see Section 5.4).

5.4 Separation of Lesion and Skin

The binary image resulting from the global thresholding is first inverted, all black pixels are turned white, and all white pixels are turned black. This is done because a binary object is defined as a cluster of white pixels on a black background. In the post-processing, 4-connection is used for all operations.

In addition to the lesion, other objects might be present in the binary image. The skin lesion must be recognized and the other objects removed. One can suspect the lesion for being the largest object in the binary image. This is true in most cases, but if the dermoscope has been tilted, the skin near the boundary of the illuminated disk can appear very dark. The illumination correction is not able to correct poor illumination beyond a certain angle between the dermoscope and the skin. Therefore, first all objects in contact with the black mask is removed, as seen in Figure 5.8.

The next step in the post-processing is to fill holes. A hole is a black area surrounded by white pixels that cannot be reached from any of the four corners of the image. The filling is necessary when there are light areas inside the lesion, as seen in Figure 5.9.

After filling the holes, the image is opened, using a disk of radius 10 pixels as structuring element. This is done to remove thin arms and break isthmuses created by the closing operation applied before the conversion to binary image, as seen in Figure 5.10.

All objects are then labeled, and the largest object is considered to be the lesion. In some cases there is more than one lesion in an image. Only the largest lesion will then be detected. An alternative is to allow user input

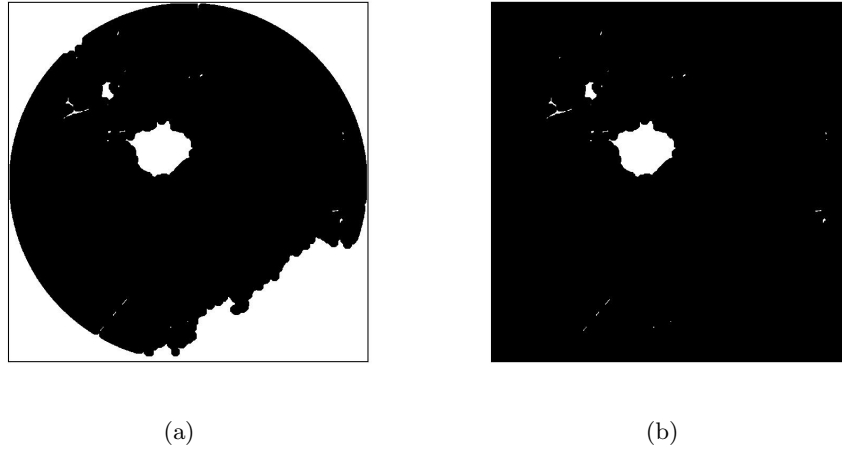


Figure 5.8: The binary image of Figure 5.4(a) (a) before and (b) after the objects in connection with the mask have been removed.

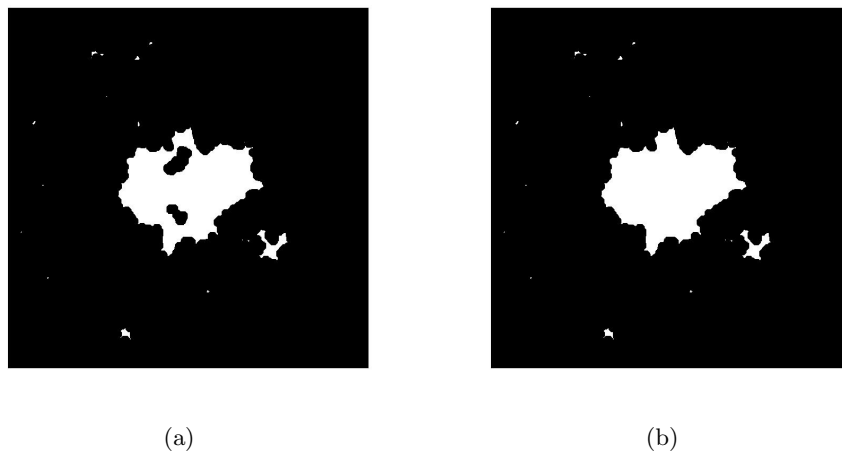
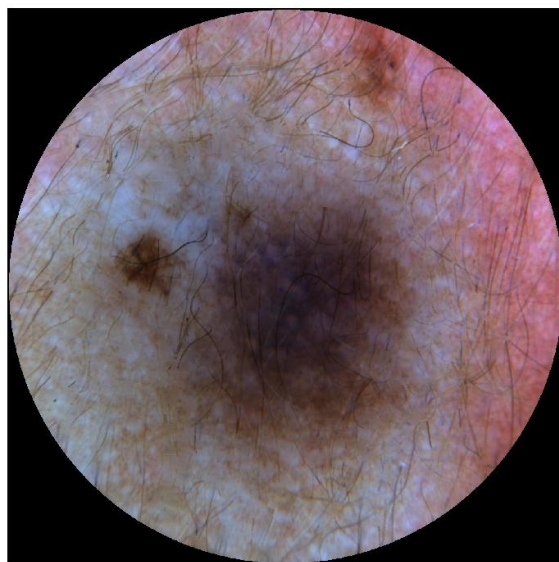
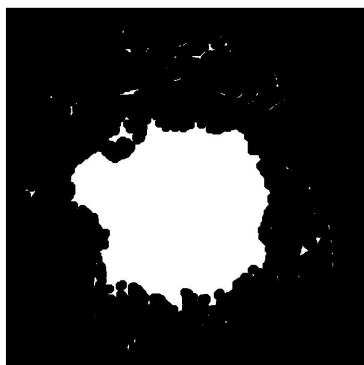


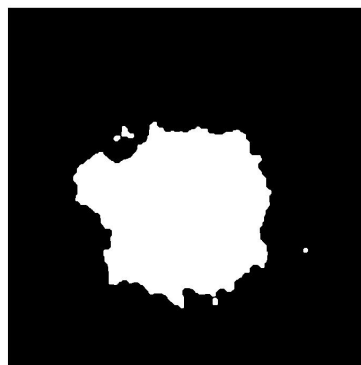
Figure 5.9: The binary image of Figure 5.6 (a) before and (b) after filling the holes.



(a)



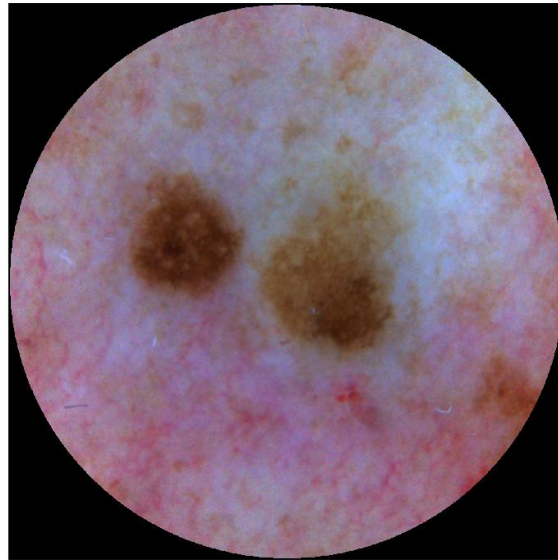
(b)



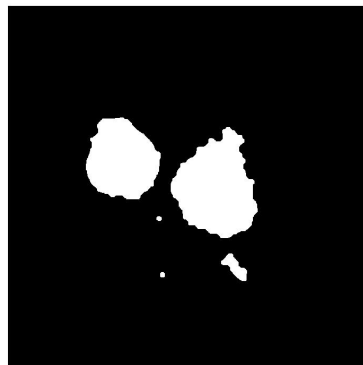
(c)

Figure 5.10: (a) The image on which the segmentation is based. (b) The binary image before and (c) after the opening.

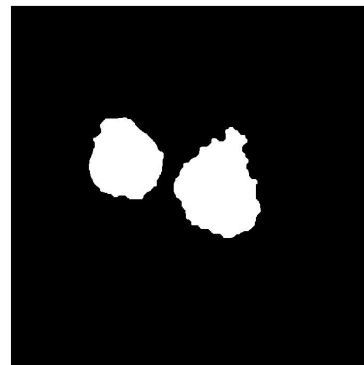
on number of lesions. Figure 5.11 shows an image with two skin lesions.



(a)



(b)



(c)

Figure 5.11: (a) The image on which the segmentation is based. (b) After the post-processing. (c) The two largest objects.

If the largest object, considered to be the lesion, covers less than 0.1% of the area of the image, the post-processing is done over again, but now the objects in direct contact with the mask are not removed.

5.5 Drawing of Borders

Even if the lesion now is separated from the skin, the borders need to be drawn. The purpose of drawing the borders is for evaluation of the segmentation. This can be done by an algorithm made by Fred Godtlielsen and Tor-Arne Øigård, described below.

- The binary image is filtered by an average filter with a 15×15 square neighborhood. The result is a blurring of the binary image, as seen in Figure 5.12(b). Only the transition between black and white is blurred. The regions of black and the regions of white keep their color.
- The absolute difference between the filtered image and 0.5 is calculated. The result, as seen in Figure 5.12(c), is an image where the previous all black and all white regions are gray (pixel value 0.5), while the blurred edges of the lesion are close to black.
- A binary image with 0.1 as threshold is created. Only the darkest pixels of the image in Figure 5.12(c) are set to 0. The result, seen in Figure 5.12(d), is a thin and smooth connected line marking the boundary of the black region in Figure 5.12(a).

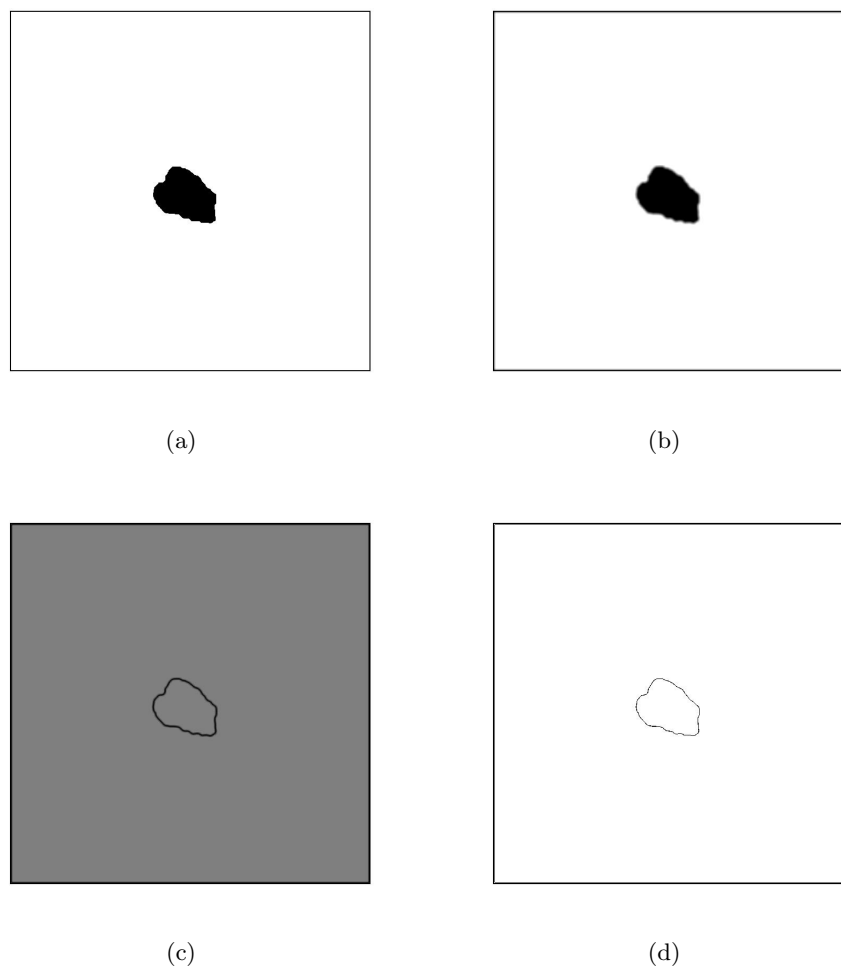


Figure 5.12: (a) The binary image after post-processing. (b) After filtration. (c) The absolute difference. (d) The final borders.

The borders in Figure 5.12(d) are superimposed on the original image of the skin lesion, as seen in Figure 5.13.

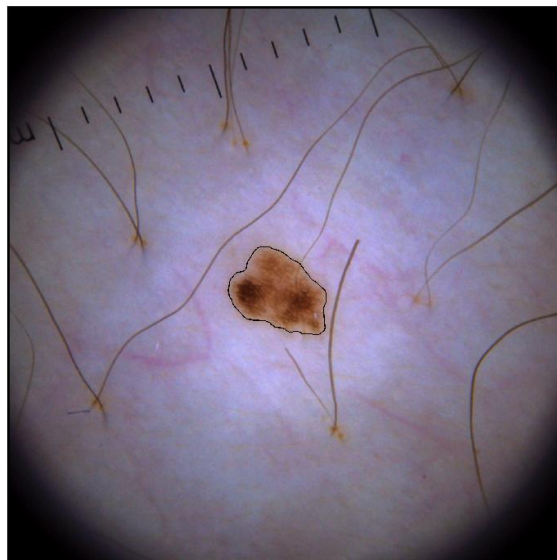
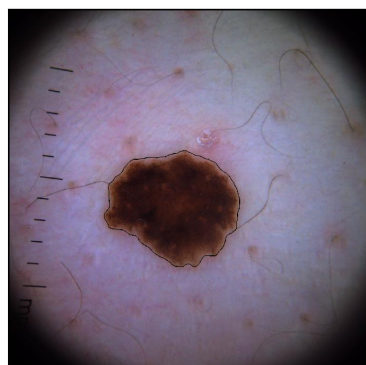


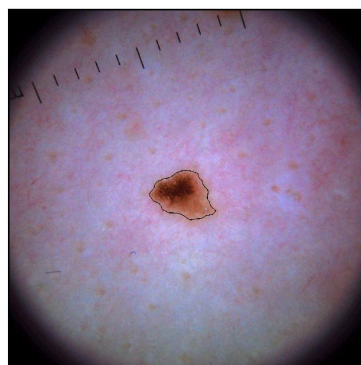
Figure 5.13: Original image with borders superimposed.

There are several other methods to draw a border around an object in a binary image, one of them is *outlining*, a morphological method, described in (Burdick, 1997, p.159). Godtlielsen and Øigård's method gives a smoother border, and is therefore preferred for evaluation since it resembles hand-drawn borders.

The results of the segmentation of Figure 5.1(a), 5.2(a), 5.4(a), 5.4(c), 5.4(e), 5.5(a), 5.5(c), 5.5(e), 5.6, 5.10(a) and 5.11(a) are seen in Figure 5.14 and Figure 5.15.



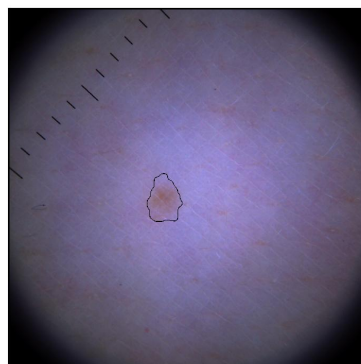
(a)



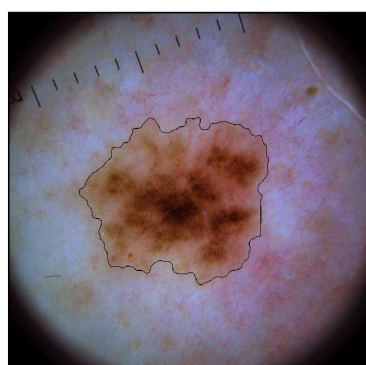
(b)



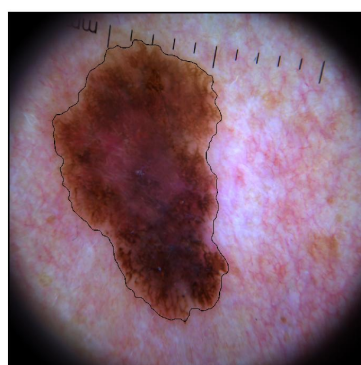
(c)



(d)

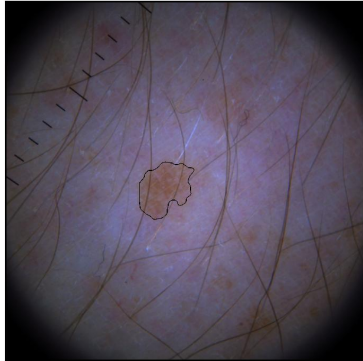


(e)

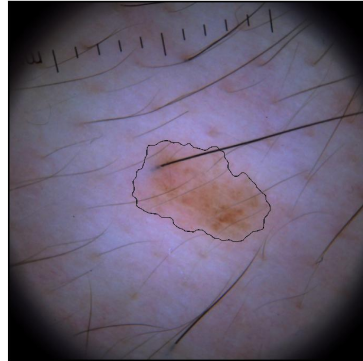


(f)

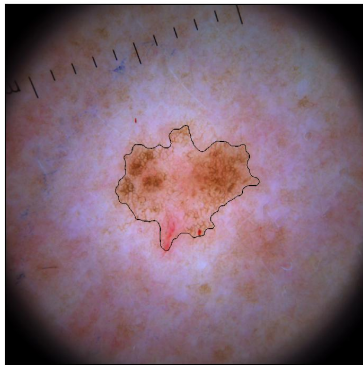
Figure 5.14: Images with borders superimposed, based on segmentation of (a) Figure 5.1(a), (b) Figure 5.2(a), (c) Figure 5.4(a), (d) Figure 5.4(c), (e) Figure 5.4(e) and (f) Figure 5.5(a).



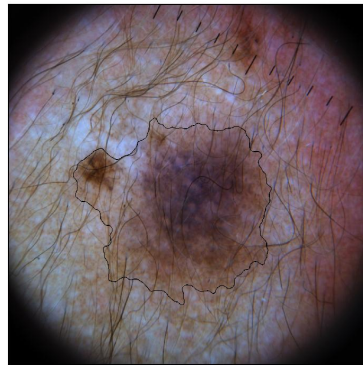
(a)



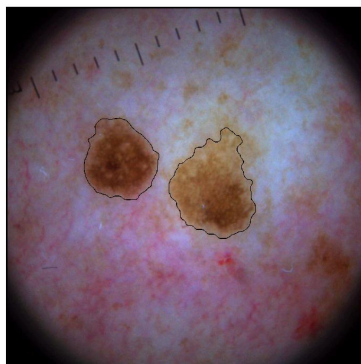
(b)



(c)



(d)



(e)

Figure 5.15: Images with borders superimposed, based on segmentation of (a) Figure 5.5(c), (b) Figure 5.5(e), (c) Figure 5.6, (d) Figure 5.10(a) and (e) Figure 5.11(a).

Chapter 6

Evaluation of Segmentation Algorithms

The most intuitive way to evaluate a segmentation algorithm for skin lesions is to compare the segmented images with segmentation done by expert dermatologists. This is a popular evaluation method, used by Celebi et al. (2007), Fleming et al. (1998), Ganster et al. (2001), Hance et al. (1996), Lim and Lee (1990), Magliogannis (2003), Gómez et al. (2008) and Magliogannis et al. (2006). Different error metrics are used. Using segmentation done by expert dermatologists or other physicians is criticized in several studies. The main drawback of using hand drawn borders as ground truth for evaluating a segmentation algorithm, is that a hand-drawn border is based on subjective criteria, and the difference between borders drawn by dermatologists are too big. Even more alarming is that the same dermatologist is not able to reproduce her own drawing (Schmid-Saugeon et al., 2003; Guillod et al., 2002; Schmid, 1999c, p.171).

Both Guillod et al. (2002) and Schmid-Saugeon et al. (2003) note that using hand-drawn borders as gold standard is somewhat contradictory to the objectives of automated image segmentation. The use of automated image segmentation is motivated by the need of objective and reproducible borders, the contrary of what borders drawn by dermatologists achieve.

Zhang et al. (2008) do a study on unsupervised evaluation methods, but the conclusion is that unsupervised methods perform poor when comparing segmentation obtained from different algorithms, or when comparing human segmentation versus unsupervised segmentation. One approach might be to

use synthesized images where the true segmentation is objective (Rajab et al., 2004), but the evaluation based on synthetic images can seldom be generalized (Zhang et al., 2008).

The evaluation of the performance of the segmentation algorithm is done by *subjective evaluation* or visual assessment. Each segmented image is evaluated by a dermatologist and given a rating. This form of evaluation is not ideal, but can give a sense of the performance of the algorithm. The method is used by Schmid-Saugeon et al. (2003), Schmid (1999b) and Lee (2001) (here by the author himself). The only objective way to evaluate the segmentation algorithm is to see whether the segmentation leads to correct classification (melanoma/non-melanoma) of the lesion.

The evaluation of the segmentation in this thesis is done by two dermatologists, each given the same sample of 80 segmented skin lesions. The dermatologist is asked to give each segmented image a rating from 1 to 4 (good-acceptable-poor-bad). The 80 images are taken from the test sample. This means that the parameters used in the segmentation algorithm (the number of bins in the histogram) are chosen independently of the 80 images that are evaluated. This is true also for the parameters of the pre-processing. All the 80 images come from the sample from Germany, where the patients are not referred. The images are the 80 first images in the test sample, after exclusions have been done according to the exclusion criteria in Section 1.5.

6.1 Comparison With Other Algorithms

Rosado et al. (2003) do a meta-analysis of 30 studies on computer diagnosis of melanoma, in 20 of them the segmentation was done automatically. The conclusion is that too many of the articles provide too little information on the methods used, for being able to compare the performance of the different algorithms. The result is a 9-point requirement list. Even if this meta-analysis focuses on classification and not segmentation, several of the points in the requirement list are necessary to follow to be able to do a comparison on segmentation algorithms. Menzies et al. (2005) recommend the same 9 requirements.

The criteria recommended by Rosado et al. (2003) that are clearly transferable to segmentation algorithms are (1) Selection of lesions should be random or consecutive, (2) Inclusion and exclusion criteria should be clearly

stated and (8) Classification (segmentation) should be carried out on an independent test set.

Since no segmentation algorithm is parameter-free, the technical information about the images is needed to do a comparison of algorithm performance. The format of the image, the size, the spatial resolution, the depth (bit/pixel) and the color space is a minimum of information required to be able to do a comparison on segmentation algorithms. In addition all pre-processes, such as filtering, removal of artifacts and gray-conversion (if any), must be described. When this information is missing, the comparison of algorithms is either very difficult or very time consuming. Since there is no common or objective method for evaluation of the segmentation, a comparison of several algorithms must run the algorithms on the same sample of images. An algorithm adjusted to one specific format, size, resolution, depth and color-space will necessarily perform poorer when these assumptions of the images are changed. Running other algorithms on ones own sample for comparison of performance gives ones own algorithm a benefit too big to be ignored.

Chapter 7

Discussion

A segmentation algorithm based on a new approach for histogram analysis is presented and tested with great success. The algorithm is simple, efficient and intuitive. Specifically this thesis presents two major algorithms; one for removal of hair and one for skin lesion segmentation.

7.1 Hair Removal

The hair-removal algorithm performs good in all the investigated cases. The performance of a hair-removal algorithm depends on two things; if it removes enough hair for correct segmentation, and if it removes other non-artifacts (pieces of skin or lesion). To do an objective evaluation of the hair-removing algorithm, one could manually detect all the hair pixels, and compare the segmentation based on the image with automatic hair removal and manual hair removal. This would be a very time consuming evaluation method and is therefore not processed. Instead, the performance of the algorithm is visually inspected. Of the 68 images in the training sample and the 224 images in the test sample there were no indications that poor hair-removal was a source for incorrect segmentation. The indications would be hairs near the lesion that are not removed and therefore included in the lesion when segmentation is done, or parts of the lesion recognized as hair, replaced by skin-colored pixel values and therefore excluded when segmentation is performed. The hair removal affects the shape of the histogram. How it affects the histogram is not as easily predicted, but it is clear that a lot of unremoved hair would possibly move the skin-mode bound and affect the

lesion-mode mean.

7.1.1 Comparison With Other Algorithms

Comparing the algorithm presented in this thesis with the algorithms presented in Schmid-Saugeon et al. (2003) and Lee et al. (1997) (described in Chapter 4.2), the use of morphological operations stands out as the common basis for all three algorithms. Schmid-Saugeon et al. (2003) and Lee et al. (1997) perform morphological operations on grayscale images, and use a pre-defined threshold value to compare the original image with the processed image. The algorithm presented in this thesis applies morphological operations to binary images, and avoids the hard-thresholding. Another advantage of using binary images is that morphological operations run much faster on binary images than on grayscale images. The algorithm of Schmid-Saugeon et al. (2003) is not recommended if histogram analysis is performed for lesion segmentation (Schmid, 1999a).

The DullRazor algorithm was tested on the 68 images in the training sample, but had a poorer performance than the algorithm presented in this thesis. To remove enough hair by DullRazor, the threshold had to be set so low that a lot of non-hair pixels were included. It must be noted that the size of the images originally used for evaluation of DullRazor were much smaller (512×486 pixels) than the images used in this thesis (1700×1700 pixels) and adjusting the size of the structuring element on DullRazor might lead to better performance.

7.2 Segmentation Algorithm

The segmentation algorithm presented provides a fast and stable segmentation based on histogram analysis and global thresholding. The novelty in the histogram analysis is that instead of searching for the two modes representing the skin and the lesion, this algorithm finds the skin mode and the valley separating the skin mode from the multiple lesion modes. The pixel values are regarded as random samples from the underlying, unknown distribution of the lesion, and the expected value of this distribution is estimated by calculating the sample mean. The only assumptions made on the distribution are that both the mean and variance exist, and that the variance is finite. The global threshold value is set to the lower limit of the

bin located at equal distance from the lesion-mode mean and the skin-mode peak. In most cases, looking at the histogram, the threshold is to the right of the valley separating the lesion modes from the skin mode. At first, this might seem like it should lead to over-segmentation, setting the threshold on the skin mode. Generally, it does not, because the limit between lesion values and skin values is not clear cut. There are pixels belonging to the lesion with values that are more skin-like, and skin pixels with lesion-like values.

7.2.1 Comparison with other algorithms

As discussed in Section 6.1, an unbiased comparison of the performance of several segmentation algorithms is difficult due to the lack of information published. It is possible to compare an algorithm with others on the method level.

Some skin lesion segmentation algorithms rely on the calculations of skin or background color (Celebi et al., 2007). This is done by taking the mean of a sample of pixels believed to belong to the skin. The drawback is obvious; for non-homogeneous skin, the value of the background color relies heavily on the pixel sample. Since this calculation is done before segmentation, the skin-pixel sample cannot be too big, or else it is a risk that the sample is taken from the lesion.

Other algorithms (Elbaum et al., 2001; Ganster et al., 2001; Rajab et al., 2004; Blum et al., 2004; Lee, 2001, p.49) restrict the segmentation to melanocytic lesions only, either by relying on blue color bands or wavelengths, or by explicitly excluding non-melanocytic lesions from the study. Even an expert dermatologist has problems distinguishing certain types of non-melanocytic skin lesions from the melanocytic group of lesions, so this rules out a small but important class of lesions.

Among the algorithms based on histogram analysis an assumption of a multimodal histogram is made in Lim and Lee (1990), Zagrouba and Barhoumi (2004), Magliogannis et al. (2006). The algorithm presented in this thesis has a solution to analysis of unimodal histograms, and therefore includes a bigger group of skin lesions.

The segmentation algorithm presented in this thesis is close to parameter-free. The only parameters present in the histogram analysis is the number of bins in the histogram and the shape of the local minimum regarded as

skin-mode bound. These two parameters might have to be adjusted for samples of images with different spatial resolution or size, but should otherwise be independent on the lesions in the sample.

Gomez et al. (2007) requires the number of classes in the image as input. In Fleming et al. (1998) the number of regions is a pre-set parameter. Lim and Lee (1990) introduce a parameter when searching for valley location. Xu et al. (1999) use image-dependent parameters, which makes the segmentation algorithm only semi-automatic. The histogram analysis in Zagrouba and Barhoumi (2004) relies heavily two parameters. The values of these parameters are adjusted to the sample used in Zagrouba and Barhoumi (2004) and is not necessarily the optimal values for other samples.

7.2.2 Challenges

Though the algorithm delivers good results for the big majority of the tested images, there are two groups of lesions that are not correctly segmented. One group is when there are areas belonging to the lesion that are lighter than the skin and are situated on the boundaries of the lesion. In these cases, the light area will be regarded as skin. The other group is images with more than one skin lesion. If the lesions are well separated, the problem can be solved by letting the physician give number of lesions as input. When the lesions lie close together, the result is an over-segmentation. It may be possible in these cases to divide the image pixels into three classes, based on the location of the skin mode, the first and the second minimum.

All images in the training sample and the test sample come from people with white skin. Although this is the group most likely to develop melanoma, there is no guarantee that a person with darker skin never will develop melanoma. The segmentation algorithm has to be tested on images taken from person with darker skin, to verify that the algorithm is skin-color independent. If it shows that the algorithm works only on images from white-skinned people, it has to be adjusted.

7.3 Expert Evaluation of Segmentation Algorithm

80 of the images from the test sample were evaluated by two dermatologists. The exact results can be seen in Appendix B. Two prints of each image were presented; one print of the original image, and one print of the original

image with the borders superimposed. The dermatologists were asked to give each border a rating from 1 to 4, where 1 = “good”, 2 = “acceptable”, 3 = “poor” and 4 = “bad”.

Of the 80 images, there were only one image where the rating done by the dermatologist differed more than 1. In this special case, dermatologist I gave the segmentation rating 1, while dermatologist II gave the rating 4. Probably the dermatologists have disagreed on which object in the image was the actual lesion. This image is excluded from the evaluation.

Among the remaining 79 images, the two dermatologists gave the rating “good” or “acceptable” to 69 (87%) of the images. 7 (9%) of the images were given the rating “poor” or “bad” by both dermatologists. The remaining 3 (4%) of the images were given the rating “acceptable” by one dermatologist and “poor” by the other.

The images in the test sample that were not evaluated by dermatologists showed similar performance, as would be expected, since the training sample and the test sample are independent.

The expert evaluation of the segmentation algorithm is promising. A high number of borders are rated “good” or “acceptable”. A real evaluation is not available until an automated diagnosis algorithm is developed, and it is possible to check whether the segmentation leads to correct diagnosis or not.

7.4 Further Work

The segmentation algorithm presented in this thesis has two major limitations; lesions with diameter greater than approximately 16 mm are excluded, and lesions lighter than the surrounding skin will not be correctly segmented. Both obstacles are possible to overcome. For big lesions, it is possible to add an optional input, “big”, to the program, where the removal of objects in contact with the mask will not be done. Whether this gives a satisfying result or not has to be investigated on a sample of images with big lesions. Light lesions might be detected by locating the right bound of the skin mode, and then estimate the location of the lesion bound as done in Section 5.3.2. A method for including light areas situated on the edge of a dark lesion is also needed.

From the expert evaluation of the 80 test images, the tendency is that

the segmentation algorithm fails when the lesion consists of several darker objects separated by skin. Criteria for including objects located near the lesion is a possible solution to this problem.

The hair-removing algorithm performs the way it should, removing the hairs that otherwise would affect the segmentation. The hair pixels are replaced by skin-colored pixels for hairs outside the lesion, and by lesion-colored pixels for hairs inside the lesion. A more elaborate method than linear interpolation is needed if the hair-removing algorithm should be used for removing hairs for analysis of the lesion.

Testing the segmentation algorithm on a sample with more malign melanomas needs to be done to assure that the algorithm performs good in all cases.

The evaluation of the performance of the segmentation algorithm is limited. At least one more dermatologist should evaluate the borders, and more lesions from the test sample should be included in the evaluation.

Appendix A

MatLab Code for Unsupervised Segmentation of Skin Lesions

Presented here are all the programs used to do the unsupervised segmentation of skin lesions as described in the thesis.

Main Program

The main program *segmentfunc* has an RGB image as input, and returns the borders of the lesion in that image. If more than one lesion is present, the number of lesions can be specified as an optional input. Several other programs are called in the main program.

segmentfunc

This program reads the image file, centers the image and corrects for non-central and non-uniform illumination. It calls other programs for removal of artifacts and segmentation.

circlefunc

This program creates a black mask with a disk-shaped white hole in the middle. The radius of the disk must be specified in the input.

rgb2pc

This program calculates the three principal components obtained from an RGB image. An optional input vector specifies which pixels are to be used in the transformation.

scaleremovalfunc

This program identifies the pixels belonging to the scales printed on the glass of the dermoscope. The scale pixels are replaced by skin-valued or lesion-valued pixels.

ladyshave

This program identifies the pixels belonging to hairs in the image. The hair pixels are replaced by skin-valued or lesion-valued pixels.

bordersfunc

This program calculates a global threshold value on the basis of histogram analysis. A binary image with white background and black objects is returned.

lesionfunc

Using the binary image obtained in *bordersfunc* as input, this program removes the objects that are not lesions, and finally draws a border around the lesion object.

Correction Matrix

In *segmentfunc*, a matrix named *corr_ppm* is loaded. This matrix is made using the program *correctionfunc*. This program calls *interpolfunc* and *vector2matrix*.

correctionfunc

This program has an RGB image as input, and calculates a vector of the pixel value mean as a function of distance from the center of the image.

interpolfunc

This program calculates the interpolation polynomials on Newton's form based on an input vector and interpolation points chosen by the user.

vector2matrix

This program creates a disk-shaped object placed on a black background with pixel values according to the values of the input vector.

11/13/08 12:44 PM /Applications/matlab/Masteroppgave/segmentpdf.m

1 of 3

```

1 function segment = segmentfunc(xx,nr)
2
3 % This is the main program of the lesion segmentation program. The
4 % program reads the ppm-file, centers the image and corrects for
5 % non-central and non-homogeneous illumination. The resulting image is then
6 % sent for scale-removal, hair-removal and segmentation. The input is an
7 % RGB image of class uint8 or uint16. An optional input is the number of
8 % lesions. The default is 1 lesion.
9
10 %% The Images
11
12 if nargin == 1           % If the number of lesions is not specified
13     nr = 1;             % in the input, it is set to 1.
14 end
15
16 x00 = imread(xx);
17
18 % The maximum value of the data class is found for gray-conversion.
19 if length(class(x00)) == 5 & class(x00) == 'uint8'
20     maxclass = 255;
21 elseif length(class(x00)) == 6 & class(x00) == 'uint16'
22     maxclass = 65535;
23 else
24     output = 'class not supported'
25     return
26 end
27
28 [n m l] = size(x00);
29
30 r0 = 800; rdiff1 = 400; rdiff2 = 200; rdiff3 = 50;
31 r1 = r0+rdiff1; r01 = r0+rdiff2; r02 = r0+rdiff3;
32
33 % Crops a (2*r1 x 2*r1)-square from the center of the rectangular image.
34 x00 = x00(floor(n/2)-r1+1:floor(n/2)+r1,floor(m/2)-r1+1:floor(m/2)+r1,:);
35
36 %% Filtering
37
38 % The each color layer of the image is filtered with a median filter,
39 % using a (5 x 5) pixel square as neighborhood. The filtering is done
40 % before conversion to double to save computation time.
41
42 x01 = ordfilt2(x00(:,1),13,ones(5,5));
43 x02 = ordfilt2(x00(:,2),13,ones(5,5));
44 x03 = ordfilt2(x00(:,3),13,ones(5,5));
45 x00 = cat(3,x01,x02,x03); x00 = double(x00);
46
47 %% Masking
48
49 % The illuminated disk is the only part of the image that is of any
50 % interest. Therefore a black mask with a disk-shaped hole is superimposed
51 % on the image. The disk-shaped hole has different radiuses for different
52 % purposes.
53
54 mask = circlefunc(r0);
55 r = find(mask == 1); q = find(mask == 0);
56
57 mask2 = circlefunc(r1);
58 r2 = find(mask2 == 1);
59
60 mask3 = circlefunc(r01);
61 r3 = find(mask3 == 1);

```


11/13/08 12:44 PM /Applications/matlab/Masteroppgave/segmentpdf.m

2 of 3

```
62
63 mask4 = circlefunc(r02);
64 r4 = find(mask4 == 1);
65
66 %% Finding the center of the illuminated disk
67
68 % The rgb-image is converted into a grayscale image, by calculating the
69 % luminance for each pixel triplet. A binary image with the 0.5-quantile as
70 % threshold is created. The object with the largest area is identified,
71 % and all other objects are removed.
72
73 x = rgb2gray(x00/maxclass);
74 light = quantile(x(:,0.5));
75 bw = im2bw(x,light);
76
77 [labeled, num] = bwlabel(bw);
78 data = regionprops(labeled,'Area','PixelList');
79 main = find([data.Area] == max([data.Area]));
80 bw = zeros(size(bw));
81 g = data(main).PixelList;
82 for j = 1: length(g(:,1))
83     bw(g(j,2),g(j,1)) = 1;
84 end
85
86 % The center of the square made by the leftmost, rightmost, uppermost and
87 % lowermost pixel of value 1 is found. This is the center of the
88 % illuminated disk.
89
90 [t t1] = find(bw == 1,1);
91 [t t2] = find(bw == 1,1,'last');
92 t = t1 + floor((t2-t1)/2);
93
94 [s s1] = find(bw' == 1,1);
95 [s s2] = find(bw' == 1,1,'last');
96 s = s1 + floor((s2-s1)/2);
97
98
99 % Cuts out a (2*r01 x 2*r01)-square with the center coinciding with the
100 % center of the illuminated disk.
101 a = s-r01+1; b = 2*r01+a-1; c = t-r01+1; d = 2*r01+c-1;
102 x00 = x00(a:b,c:d,:);
103
104
105 %% Finding the center of illumination
106
107 % A binary image with the 0.75-quantile as threshold is created. The binary
108 % image is then morphologically opened to avoid outlying pixels of value 1.
109
110 x = rgb2gray(x00/maxclass);
111 light = quantile(x(r3),0.75);
112 bw = im2bw(x,light);
113
114 se = strel('square',5);           % Creates the structuring element.
115 bw = imopen(bw,se);
116
117 % The center of the square made by the leftmost, rightmost, uppermost and
118 % lowermost pixel of value one is found. This is the center of
119 % illumination.
120
121 [t t1] = find(bw == 1,1);
122 [t t2] = find(bw == 1,1,'last');
```

11/13/08 12:44 PM /Applications/matlab/Masteroppgave/segmentpdf.m

3 of 3

```

123 t = t1 + floor((t2-t1)/2);
124
125 [s s1] = find(bw' == 1,1);
126 [s s2] = find(bw' == 1,1,'last');
127 s = s1 + floor((s2-s1)/2);
128
129 %% Correction Matrix
130
131 % The correction matrix is cut to a (r02xr02)-matrix with center coinciding
132 % with the center of illumination.
133
134 load corr_ppm          % Loads the genreal correction matrix.
135
136 % The correction matrix is cropped to the right size. The min-max makes
137 % sure that the bounds of the new correction matrix is inside the borders
138 % of the general correction matrix.
139 [w w] = size(corr_ppm);
140 a = min(max(w/2-r02+r01-s+1,1),301);
141 c = min(max(w/2-r02+r01-t+1,1),301);
142 b = 2*r02+a-1; d = 2*r02+c-1;
143
144 corr_ppm = corr_ppm(a:b,c:d);
145
146 % Multiplies the rgb-image with correction matrix.
147 x00 = x00(rdif2-rdfff3+1:end-(rdiff2-rdfff3),rdiff2-rdfff3+1:end-(rdiff2-rdfff3),:);
148 x0 = x00.*cat(3,mask4,mask4,mask4).*cat(3,corr_ppm,corr_ppm,corr_ppm);
149
150 %% Removal of scales and hair
151
152 x0 = scaleremovalfunc(x0,maxclass); % Returns an image without scales.
153 x0 = ladyshave(x0,maxclass); % Returns an image without hairs.
154
155 % The image is cut to its final size; (r0xr0).
156 x0 = x0(rdfff3+1:end-rdfff3,rdfff3+1:end-rdfff3,:).*cat(3,mask,mask,mask);
157 x00 = x00(rdfff3+1:end-rdfff3,rdfff3+1:end-rdfff3,:);
158
159 % The image is segmented.
160 border = bordersfunc(x0,r); % Returns a binary image.
161
162 % The lesion is identified, and the borders are drawn.
163 lesion = lesionfunc(x00,border,r,nr); % Returns the borders.
164
165 segment = lesion;

```

11/13/08 2:13 PM /Applications/matlab/Masteroppgave/circlepdf.m

1 of 1

```
1 function circle = circlefunc(r);
2
3 %% This function creates an (r x r) binary image with black corners and a
4 %% white disk of radius r
5
6 y = -r:1:r;
7
8 x1 = ceil(sqrt(r^2 - y.*y) + r);
9 x2 = floor(-sqrt(r^2 - y.*y) + r);
10
11 x = logical(ones(2*r,2*r));
12
13 for i = 1: 2*r
14     x(i,x1(i):end) = 0;
15     x(i,1:x2(i)) = 0;
16 end
17
18 x(1:r,:) = x(end:-1:r+1,:);
19 x(:,r+1:end) = x(:,r:-1:1);
20 circle = x;
```

11/13/08 1:41 PM /Applications/matlab/Masteroppgave/rgb2pc_pdf.m

1 of 1

```
1 function pc = rgb2pc(x0,r)
2
3 %% This function converts an RGB image to three grayscale images, based on
4 %% the principal component transform. The input is the RGB image and a
5 %% vector, r, that indicates which pixels that should be used in the
6 %% conversion. The default of r is all pixels.
7
8 [n m l] = size(x0);
9
10 if nargin == 1
11     r = 1: n*m;
12 end
13
14 x = zeros(n,m,l);
15
16 % Generates a (3 x r) matrix from the r pixels of each color layer.
17 x01 = x0(:,:,1); x02 = x0(:,:,2); x03 = x0(:,:,3);
18 Xr = [x01(r)'; x02(r)'; x03(r)'];
19
20 % Calculates the eigenvalues and the corresponding eigenvectors of the
21 % covariance matrix.
22 [V,D] = eig(cov(Xr));
23 e1 = V(:,3); e2 = V(:,2); e3 = V(:,1);
24
25 % Calculates the three principal components on the basis of the
26 % eigenvalues.
27
28 xr1 = e1'*Xr;
29 x1 = zeros(n,m);
30 x1(r) = xr1;
31 x(:, :, 1) = x1;
32
33 xr2 = e2'*Xr;
34 x2 = zeros(n,m);
35 x2(r) = xr2;
36 x(:, :, 2) = x2;
37
38 xr3 = e3'*Xr;
39 x3 = zeros(n,m);
40 x3(r) = xr3;
41 x(:, :, 3) = x3;
42
43 % Returns the PCT matrix.
44 pc = x;
```

11/13/08 2:10 PM /Applications/matlab/Masteroppgave/scaleremovalpdf.m

1 of 4

```

1 function scaleremoval = scaleremovalfunc(x00,maxclass)
2
3 %% This function removes the scales from a dermoscopic image, and replaces
4 %% the scale-pixels with skin-colored pixels. The input is a truecolor
5 %% RGB image. This image is converted to a grayscale image and then
6 %% converted to a binary image. The pixels belonging to a scale is
7 %% recognized by being situated in a structure that is long and thin. A
8 %% scale object must be of a certain size (100 to 900 pixels) and
9 %% eccentricity (minimum 0.95) to be accepted. The 11 scales must be
10 %% situated in a certian position to be accepted. The scale pixels are
11 %% replaced by a value obtained by linear interpolation. The output is the
12 %% RGB image with replaced scale-pixels.
13
14 [n m l] = size(x00);
15
16 % Adds a black frame of size f around the image.
17 f = 20; x0 = zeros(n+2*f,m+2*f,l); x0(f+1:n+f,f+1:m+f,:) = x00;
18 [n m l] = size(x0);
19
20 % Creates a disk of radius r6 pixels, all scales are outside this disk.
21 r6 = 450; diff = n/2-r6; circ = logical(zeros(n,m));
22 circ(diff+1:n-diff,diff+1:n-diff) = circlefunc(r6);
23
24 % Creates a black mask with a disk-shaped hole. The pixels belonging to the
25 % black mask are not replaced, since they are not used in the final
26 % segmentation.
27 r0 = 800; mask = zeros(n,m); diff = n/2-r0;
28 circ2 = circlefunc(r0); mask(diff+1:n-diff,diff+1:m-diff) = circ2;
29 qq = find(mask == 0);
30
31
32 deg = [0 30 60]; % Angles of rotation.
33
34 xgray = rgb2gray(x0/maxclass); % Converts to gray.
35 num_scales = 10; % The minimum number of scales.
36 xx = x0;
37
38 % Identifies zero and non-zero pixels. The zero pixels will not be checked
39 % for scale objects.
40 p = find(x0(:,:,1) > 0); q = find(x0(:,:,1) == 0);
41
42 se = strel('square',7); % Structuring element
43
44 % If the scales are not found in the first binary image, a new binary image
45 % with a different threshold will be made.
46 thresh = [0.4 0.6];
47 for w = 1:2
48     for d = 1:length(deg)
49
50         x0rot = imrotate(x0,deg(d),'crop'); % Rotates the full image.
51         xp1 = x0rot(:,:,1); xp2 = x0rot(:,:,2); xp3 = x0rot(:,:,3);
52
53         xp11 = imrotate(xgray,deg(d),'crop'); % Rotates the grayscale image.
54         xp11 = xp11 - min(xp11(p)); xp11 = xp11/max(xp11(p)); % Normalizes.
55
56 % Creates binary image with 0.4 (or 0.6) as threshold. The pixels possibly
57 % belonging to a scale object are black (0). The pixels that are black in
58 % the original image, or are situated inside the central disk, are set to
59 % 1.
60
61     bw = logical(zeros(n,m));

```

11/13/08 2:10 PM /Applications/matlab/Masteroppgave/scaleremovalpdf.m

2 of 4

```

62     bw(p) = im2bw(xp11(p),thresh(w));
63     bw(q) = 1;  bw(circ == 1) = 1;
64
65     bb_v = logical(zeros(n,m)); bb_h = logical(zeros(n,m));
66     %% Each row in the binary image is checked for scale pixels.
67     % For each black pixel, check if there are white pixels to the left AND to
68     % the right. If this is true, check if all pixels in the neighborhood
69     % above OR below are black. If this is true, the current pixel is stored
70     % as a possible vertical scale-pixel. The same procedure is repeated for
71     % the horizontal direction.
72     for i = f+1: n-f
73         t = find(bw(i,:) == 0);
74         if length(t) > 0
75             for j = 1: length(t)
76                 if sum(bw(i,t(j)-10:t(j))) > 0 & sum(bw(i,t(j):t(j)+10)) > 0
77                     if sum(bw(i-10:i,t(j))) == 0 | sum(bw(i:i+10,t(j))) == 0
78                         bb_v(i,t(j)) = 1;
79                     end
80                 end
81                 if sum(bw(i-10:i,t(j))) > 0 & sum(bw(i:i+10,t(j))) > 0
82                     if sum(bw(i,t(j)-10:t(j))) == 0 | sum(bw(i,t(j):t(j)+10)) == 0
83                         bb_h(i,t(j)) = 1;
84                     end
85                 end
86             end
87         end
88         clear t
89     end
90
91     %% Vertical scale objects
92     % Each vertical scale object is labeled, and its area and eccentricity is
93     % caclulated.
94
95     [labeled, num] = bwlabel(bb_v);
96     data = regionprops(labeled,'PixelList','Eccentricity','FilledArea');
97     E = [data.Eccentricity]; F = [data.FilledArea];
98
99     % All objects smaller than 100 pixels, bigger than 900 pixels, or with
100    % eccentricity smaller than 0.95 are removed.
101     t = find(F < 100 | F > 900 | E < 0.95);
102
103     for i = 1: length(t)
104         g = data(t(i)).PixelList;
105         for j = 1: length(g(:,1))
106             bb_v(g(j,2),g(j,1)) = 0;
107         end
108     end
109
110    % Counts the objects in the upper part of the scale matrix. If the number
111    % of objects is greater than or equal to the number of scale objects, the
112    % upper matrix is dilated. The pixels belonging to a scale is replaced by
113    % a value obtained from linear interpolation. If the number of objects is
114    % too low, the objects are removed.
115
116     bb_v(qq) = 0;
117     [labeled,numU] = bwlabel(bb_v(1:n/2,:));
118     if numU > num_scales-1
119         bb_v(1:n/2,:) = imdilate(bb_v(1:n/2,:),se);
120         for i = f+1: n/2
121             t = find(bb_v(i,:) == 1);
122             if length(t) > 0

```

11/13/08 2:10 PM /Applications/matlab/Masteroppgave/scaleremovalpdf.m

3 of 4

```

123         for j = 1: length(t)
124             left_r = find(bb_v(i,1:t(j)) == 0,10,'last');
125             right_r = find(bb_v(i,t(j):end) == 0,10);
126             r = [left_r(1) t(j)+right_r(end)-1];
127
128             xp1(i,t(j)) = interp1(r,xp1(i,r),t(j));
129             xp2(i,t(j)) = interp1(r,xp2(i,r),t(j));
130             xp3(i,t(j)) = interp1(r,xp3(i,r),t(j));
131         end
132         clear t
133     end
134 end
135 else
136     bb_v(1:n/2,:) = 0;
137 end
138
139 % Same procedure for the lower half of the matrix.
140 [labeled,numD] = bwlabel(bb_v(n/2+1:end,:));
141 if numD > num_scales-1
142     bb_v(n/2+1:end,:) = imdilate(bb_v(n/2+1:end,:),se);
143     for i = n/2+1: n-f
144         t = find(bb_v(i,:) == 1);
145         if length(t) > 0
146             for j = 1: length(t)
147                 left_r = find(bb_v(i,1:t(j)) == 0,10,'last');
148                 right_r = find(bb_v(i,t(j):end) == 0,10);
149                 r = [left_r(1) t(j)+right_r(end)-1];
150                 xp1(i,t(j)) = interp1(r,xp1(i,r),t(j));
151                 xp2(i,t(j)) = interp1(r,xp2(i,r),t(j));
152                 xp3(i,t(j)) = interp1(r,xp3(i,r),t(j));
153             end
154             clear t
155         end
156     end
157 else
158     bb_v(n/2+1:end,:) = 0;
159 end
160
161 clear E F
162
163 %% Horizontal scale objects
164 % Follows the same procedure as for vertical scale objects.
165
166 [labeled, num] = bwlabel(bb_h);
167 data = regionprops(labeled,'PixelList','Eccentricity','FilledArea');
168 E = [data.Eccentricity]; F = [data.FilledArea];
169 t = find(F < 100 | F > 900 | E < 0.95);
170
171 for i = 1: length(t)
172     g = data(t(i)).PixelList;
173     for j = 1: length(g(:,1))
174         bb_h(g(j,2),g(j,1)) = 0;
175     end
176 end
177
178 bb_h(qq) = 0;
179 [labeled,numL] = bwlabel(bb_h(:,1:m/2));
180 if numL > num_scales-1
181     bb_h(:,1:m/2) = imdilate(bb_h(:,1:m/2),se);
182
183     for i = f+1: n-f

```

11/13/08 2:10 PM /Applications/matlab/Masteroppgave/scaleremovalpdf.m

4 of 4

```

184     s = find(bb_h(i,1:m/2) == 1);
185     if length(s) > 0
186         for j = 1: length(s)
187             up_r = find(bb_h(1:i,s(j)) == 0,10,'last');
188             low_r = find(bb_h(i:end,s(j)) == 0,10);
189             r = [up_r(1) i+low_r(end)-1]';
190             xp1(i,s(j)) = interp1(r,xp1(r,s(j)),i);
191             xp2(i,s(j)) = interp1(r,xp2(r,s(j)),i);
192             xp3(i,s(j)) = interp1(r,xp3(r,s(j)),i);
193         end
194     clear s
195     end
196 end
197 else
198     bb_h(:,1:m/2) = 0;
199 end
200
201 [labeled,numR] = bwlabel(bb_h(:,m/2+1:end));
202 if numR > num_scales-1
203     bb_h(:,m/2+1:end) = imdilate(bb_h(:,m/2+1:end),se);
204     for i = f+1: n-f
205         s = find(bb_h(i,m/2+1:end) == 1);
206         if length(s) > 0
207             for j = 1: length(s)
208                 up_r = find(bb_h(1:i,s(j)) == 0,10,'last');
209                 low_r = find(bb_h(i:end,s(j)) == 0,10);
210                 r = [up_r(1) i+low_r(end)-1]';
211                 xp1(i,s(j)) = interp1(r,xp1(r,s(j)),i);
212                 xp2(i,s(j)) = interp1(r,xp2(r,s(j)),i);
213                 xp3(i,s(j)) = interp1(r,xp3(r,s(j)),i);
214             end
215         clear s
216         end
217     end
218 else
219     bb_h(:,m/2+1:end) = 0;
220 end
221
222 x_x = cat(3,xp1,xp2,xp3);
223 if d > 1 % Rotates back to original position.
224     x_x = imrotate(x_x,-deg(d),'crop');
225 end
226
227 % Combines current color image with the ones from earlier rotations.
228 xx = max(xx,x_x);
229 end
230 % If scale objects have been found for the 0.4-threshold, the procedure is
231 % not repeated.
232 if sum(bb_v(:)) + sum(bb_h(:)) > 0
233     break
234 end
235 end
236
237 xx = xx(f+1:n-f,f+1:m-f,:); % Removes the frame.
238
239 % Returns the original-sized image with the scale replaced by skin-colored
240 % pixels.
241 scaleremoval = xx;

```


11/13/08 3:39 PM /Applications/matlab/Masteroppgave/ladyshave_pdf.m

1 of 4

```

1 function shaved = ladyshave(x00,maxclass)
2
3 %% This function removes hair in a dermosopic image, and replaces the hair
4 %% pixels with skin colored pixels. The hairs are divided into two
5 %% categories, thin and thick, and they are found in two directions,
6 %% vertical and horizontal. The image is rotated four times, each time
7 %% with an angle of 22.2 degrees. The input is an RGB image. The red layer
8 %% of the image is used as a grayscale image. The grayscale image is then
9 %% converted into a binary image, using 5 different thresholds. The
10 %% thresholds are calculated on the basis of Otsu's method. In the binary
11 %% image, the possible hair pixels are black. To be accepted as a hair
12 %% pixel, the pixel must belong to a long and thin structure. The hair
13 %% object must have a minimum major axis, and a minimum eccentricity. The
14 %% hair pixels are replaced by a value found by linear interpolation.
15
16 [n m l] = size(x00);
17
18 % Adds a black frame of width f around the image.
19 f = 20; x0 = zeros(n+2*f,m+2*f,l); x0(f+1:n+f,f+1:m+f,:) = x00;
20 [n m l] = size(x0);
21
22 % This defines the region of interest. Pixels outside this region need not
23 % be replaced.
24 r0 = 800;
25 mask = zeros(n,m); diff = n/2-r0;
26 circ = circlefunc(r0); mask(diff+1:n-diff,diff+1:m-diff) = circ;
27 qq = find(mask == 0);
28
29 deg = [0 22.5 45 67.5]; % Angle of rotation.
30
31 bb_2 = zeros(n,m);
32 xx = x0;
33 xp1 = x0(:,:,1)/maxclass;
34
35 % Finds the zero and non-zero pixels in the original image. The zero pixels
36 % are not taken into consideration.
37 p = find(xp1 > 0); q = find(xp1 == 0);
38 grayth = graythresh(xp1(p)); % Otsu's threshold.
39 w = [0.85 0.9 0.95 1 1.05];
40
41 for d = 1: length(deg) % For all rotations.
42
43 x = imrotate(x0,deg(d),'crop'); % Rotates the image
44 xp1 = x(:,:,1); xp2 = x(:,:,2); xp3 = x(:,:,3);
45 xp11 = xp1/maxclass;
46
47 bb_v = logical(zeros(n,m)); bb_h = logical(zeros(n,m));
48 bb_v20 = logical(zeros(n,m)); bb_h20 = logical(zeros(n,m));
49
50 for v = 1: length(w) % For all multiples of the Otsu's threshold.
51
52 % Creates a binary image where the threshold value is a multiple of Otsu's
53 % threshold. The possible hair pixels will appear black (0).
54 level = w(v).*grayth;
55 bw = im2bw(xp11,level); bw(q) = 1;
56
57 % To identify possible hair pixels in the vertical direction, a closing
58 % operation with a horizontal structuring element is applied to the binary
59 % image. The closing operation removes all objects thin enough to be a hair.
60 % The original binary image is then subtracted from the closed image, and
61 % the resulting image shows possible hair pixels as white pixels. An

```

11/13/08 3:39 PM /Applications/matlab/Masteroppgave/ladyshave_pdf.m

2 of 4

```

62 % opening operation with a vertical structuring element is applied, to
63 % remove objects that are too short to be a hair.
64     se = strel('line',21,0);
65     se2 = strel('line',11,90);
66     bw_closed = imclose(bw,se);
67     bw_diff = logical(bw_closed - bw);
68     bb_v = max(bb_v,imopen(bw_diff,se2));
69
70 % The same procedure is applied to identify possible hair pixels in the
71 % horizontal direction, but with transposed structuring elements.
72     se = strel('line',21,90);
73     se2 = strel('line',11,0);
74     bw_closed = imclose(bw,se);
75     bw_diff = bw_closed - bw;
76     bb_h = max(bb_h,imopen(bw_diff,se2));
77
78 % To identify thick hairs, the same procedure is applied, but with longer
79 % structuring elements.
80
81     se = strel('line',41,0);
82     se2 = strel('line',31,90);
83     bw_closed = imclose(bw,se);
84     bw_diff = bw_closed - bw;
85     bb_v20 = max(bb_v20,imopen(bw_diff,se2));
86
87     se = strel('line',41,90);
88     se2 = strel('line',31,0);
89     bw_closed = imclose(bw,se);
90     bw_diff = bw_closed - bw;
91     bb_h20 = max(bb_h20,imopen(bw_diff,se2));
92 end
93
94 %% Thin vertical hairs
95 % The possible hair objects are labeled. The eccentricity and major
96 % axis length of the corresponding ellipse are calculated.
97 [labeled, num] = bwlabel(bb_v);
98 data = regionprops(labeled,'PixelList','Eccentricity','MajorAxisLength');
99 E = [data.Eccentricity]; M = [data.MajorAxisLength];
100
101 % All objects with major axis shorter than 100 pixels or eccentricity less
102 % than 0.975 are removed.
103 t = find(M < 100 | E < 0.975);
104
105 for i = 1: length(t)
106     g = data(t(i)).PixelList;
107     for j = 1: length(g(:,1))
108         bb_v(g(j,2),g(j,1)) = 0;
109     end
110 end
111
112 clear E M t
113
114 %% Thick vertical hairs
115 % Same procedure as thin hairs, except the minimum length of the major
116 % axis.
117
118 [labeled, num] = bwlabel(bb_v20);
119 data = regionprops(labeled,'PixelList','Eccentricity','MajorAxisLength');
120 E = [data.Eccentricity]; M = [data.MajorAxisLength];
121 t = find(M < 250 | E < 0.975);
122

```

11/13/08 3:39 PM /Applications/matlab/Masteroppgave/ladyshave_pdf.m

3 of 4

```
123 for i = 1: length(t)
124     g = data(t(i)).PixelList;
125     for j = 1: length(g(:,1))
126         bb_v20(g(j,2),g(j,1)) = 0;
127     end
128 end
129
130 clear E M t
131
132 %% Thin horizontal hairs
133 % Same procedure as with vertical hairs.
134
135 [labeled, num] = bwlabel(bb_h);
136 data = regionprops(labeled,'PixelList','Eccentricity','MajorAxisLength');
137 E = [data.Eccentricity]; M = [data.MajorAxisLength];
138 t = find(M < 100 | E < 0.975);
139
140 for i = 1: length(t)
141     g = data(t(i)).PixelList;
142     for j = 1: length(g(:,1))
143         bb_h(g(j,2),g(j,1)) = 0;
144     end
145 end
146
147 clear E M t
148
149 %% Thick horizontal hairs
150 % Same procedure as with thick vertical hairs.
151 [labeled, num] = bwlabel(bb_h20);
152 data = regionprops(labeled,'PixelList','Eccentricity','MajorAxisLength');
153 E = [data.Eccentricity]; M = [data.MajorAxisLength];
154 t = find(M < 250 | E < 0.975);
155
156 for i = 1: length(t)
157     g = data(t(i)).PixelList;
158     for j = 1: length(g(:,1))
159         bb_h20(g(j,2),g(j,1)) = 0;
160     end
161 end
162
163 clear E M t
164
165
166 %% Dilation of the binary images
167 % The dilation is done to include the shadows in the hair objects.
168
169 bb_v = max(bb_v,bb_v20); bb_h = max(bb_h,bb_h20);
170 se = strel('square',7);
171 bb_v = imdilate(bb_v,se); bb_h = imdilate(bb_h,se);
172
173
174 %% Replacement of the hair pixels with skin-valued pixels
175 % Each vertical (horizontal) hair-pixel value is replaced by the value
176 % obtained from linear interpolation. The pixels used for interpolation are
177 % the 10th non-hair pixel to the right (above) and the 10th non-hair pixel
178 % to the left of (below) the current hair pixel. Only the pixels in the
179 % region of interest will be replaced.
180
181 bb_v(qq) = 0; bb_h(qq) = 0;
182
183 for i = f+1: n-f
```

11/13/08 3:39 PM /Applications/matlab/Masteroppgave/ladyshave_pdf.m

4 of 4

```

184
185     t = find(bb_v(i,:) == 1);
186     s = find(bb_h(i,:) == 1);
187
188     if length(t) > 0           % Vertical hairs.
189         for j = 1: length(t)
190             left_r = find(bb_v(i,1:t(j)) == 0,10,'last');
191             right_r = find(bb_v(i,t(j):end) == 0,10);
192             r = [left_r(1) t(j)+right_r(end)-1];
193
194             xp1(i,t(j)) = interp1(r,xp1(i,r),t(j));
195             xp2(i,t(j)) = interp1(r,xp2(i,r),t(j));
196             xp3(i,t(j)) = interp1(r,xp3(i,r),t(j));
197         end
198     end
199
200     if length(s) > 0           % Horizontal hairs.
201         for j = 1: length(s)
202
203             upper_r = find(bb_h(1:i,s(j)) == 0,10,'last');
204             low_r = find(bb_h(i:end,s(j)) == 0,10);
205             r = [upper_r(1) i+low_r(end)-1];
206
207             xp1(i,s(j)) = interp1(r,xp1(r,s(j)),i);
208             xp2(i,s(j)) = interp1(r,xp2(r,s(j)),i);
209             xp3(i,s(j)) = interp1(r,xp3(r,s(j)),i);
210         end
211     end
212
213     clear s t
214 end
215
216     x_x = cat(3,xp1,xp2,xp3);
217     % Rotates back to the original position.
218     x_x = imrotate(x_x,-deg(d),'crop');
219     % Combines the matrix with the hairs replaced with previous
220     % combinations of matrices with hairs replaced.
221     xx = max(xx,x_x);
222 end
223
224
225 xx = xx(f+1:n-f,f+1:m-f,:); % Removes the frame.
226
227 % Returns an RGB image with the hair pixels replaced by skin-valued pixels.
228 shaved = xx;

```

```
1 function borders = bordersfunc(x0,r,name)
2
3 %% This function segments the lesion from the background skin. First the
4 %% RGB-image is transformed into a grayscale image by the PCT-method. The
5 %% grayscale image is filtered with a median filter. A histogram with 100
6 %% bins is made. First the maximum of the histogram is found. This bin is
7 %% called the skin mode. Then the first local minimum to the left of the
8 %% skin mode is found. The mean of all pixel values to the left of the
9 %% local minimum is calculated. This mean is called the lesion mode. A
10 %% morphological opening operation is applied to the grayscale image. A
11 %% binary image is made from the opened grayscale image, where the
12 %% threshold is the middle point between the lesion mode and the skin mode.
13
14 xpc = rgb2pc(x0,r);           % Conversion to principal components.
15 x1 = xpc(:,1);               % 1st principal component.
16 x1 = ordfilt2(x1,13,ones(5,5)); % Filters to avoid outliers.
17 x1(r) = x1(r) - min(x1(r)); x1(r) = x1(r)/max(x1(r)); % Normalizes.
18
19 h = hist(x1(r),100);        % Creates histogram with 100 bins.
20
21 % The skinmode is the highest bin among the 75 rightmost bins.
22 skinmode = 24 + find(h(25:end) == max(h(25:end)));
23
24 % Finds the first bin, t, containing more than 2014 pixels (0.1% of the
25 % total number of pixels).
26 p = round(length(r)*0.001);
27 for t = 1: round(skinmode/3)
28     if sum(h(1:t)) > p
29         break
30     end
31 end
32
33 % Finds the first peak to the right of t, but with minimum 5.
34 t = max(t,5);
35 for i = t: skinmode-10
36     if h(i+1:i+2) < h(i)
37         break
38     end
39 end
40
41 % Check if the histogram is unimodal. The histogram is considered to be
42 % unimodal if no peak has been found between t and skinmode - 10. Then the
43 % skin mode bound is chosen to be skinmode/3.
44 if i == skinmode-10
45     minh = round(skinmode/3);
46     i = t;
47 else
48
49 % If a peak is found, the bound of the skin mode is searched for between
50 % the peak and the skinmode. A local minimum is considered to be the bound
51 % if the four bins to the left and the four bins to the right are taller
52 % than the current bin.
53
54     for j = skinmode-10:-1:i
55         if h(j-4:j-1) > h(j) & h(j+1:j+4) > h(j)
56             minh = j;
57             break
58         end
59     end
60
61 % If no bound is found, the bound is set to be the minimum bin between
```

11/13/08 3:57 PM /Applications/matlab/Masteroppgave/borderspdf.m

2 of 2

```
62 % the peak and the skinmode.
63
64 if j == i
65     minh = i + find(h(i:skinmode) == min(h(i:skinmode)),1,'last') - 1;
66 end
67 end
68
69 %% Threshold value
70 % All pixels belonging to the bins to the left of the bound are considered
71 % to belong to the lesion mode. The mean of these pixels are set as the
72 % location of the lesion mode peak.
73 xx = x1(r);
74 s = find(xx < 0.01*minh);
75 M = mean(xx(s));
76
77 % The threshold value is that of the pixels in the bin located at equal
78 % distance from the lesion mode peak and the skin mode peak.
79 thresh = M + (0.01*skinmode-M)/2
80
81 % The image is opened with a disk of radius 20 pixels as structuring
82 % element.
83 se = strel('disk',20);
84 x1 = imopen(x1,se);
85
86 % The binary image is created.
87 bw = im2bw(x1(r),thresh);
88 x1(r) = bw;
89 bw = reshape(x1,size(x1));
90
91 % The function returns a binary image with the same size as the input
92 % image.
93 borders = bw;
```

11/13/08 4:21 PM /Applications/matlab/Masteroppgave/lesionpdf.m

1 of 2

```
1 function lesion = lesionfunc(border,r,nr)
2
3 %% This function identifies the lesion and draws a border around it. The
4 %% input is a binary image where one or more of the black objects is a
5 %% lesion. The number of lesions present is also an input. The
6 %% post-processing done in this function is removing those objects and
7 %% pixels that are not skin lesions. When only the skin lesion objects
8 %% remain, a border is drawn.
9
10 bw = border;
11 bw2 = ones(size(border)); % The binary image is reversed.
12 bw2(border == 1) = 0;
13
14 % All objects in contact with the mask surrounding the disk-shaped middle
15 % of the binary image are removed. To do this, the mask is connected to one
16 % object, and the whole object starting in the upper left corner is removed.
17 bw2(1,:) = 1; bw2(end,:) = 1; bw2(:,1) = 1; bw2(:,end) = 1;
18 [labeled, num] = bwlabel(bw2,4);
19 data = regionprops(labeled,'PixelList');
20 g = data(1).PixelList;
21 for j = 1:length(g(:,1))
22     bw2(g(j,2),g(j,1)) = 0;
23 end
24
25 % The image is filled, meaning that all black regions that cannot be
26 % reached from any of the four corners without traversing white pixels are
27 % turned white.
28 bw2 = imfill(bw2,'holes');
29
30 % The image is opened, using a disk with 10 pixel radius as structuring
31 % element.
32 se = strel('disk',10);
33 bw2 = imopen(bw2,se);
34
35 % The objects are labeled.
36 [labeled, num] = bwlabel(bw2,4);
37 data = regionprops(labeled,'Area');
38
39 if nr == 1 % If there is only one lesion present.
40     mainArea = max([data.Area]) % The largest object is found.
41 % If the largest object covers less than 0.1% of the image, the
42 % post-processing is done without removing the objects in contact with the
43 % mask.
44     if mainArea < round(length(r)*0.001)
45         bw = border;
46         bw2 = ones(size(bw));
47         bw2(bw == 1) = 0;
48         bw2 = imfill(bw2,'holes');
49         figure, imagesc(bw2), colormap(gray)
50         se = strel('disk',10);
51         bw = imopen(bw2,se);
52         [labeled, num] = bwlabel(bw,4);
53         data = regionprops(labeled,'Area','PixelList');
54         mainArea = max([data.Area])
55     end
56     main = find([data.Area] == mainArea);
57
58 % A binary image with white background and a black object consisting of the
59 % pixels in the largest object is created.
60     bw2 = ones(size(bw));
61     g = data(main).PixelList;
```

11/13/08 4:21 PM /Applications/matlab/Masteroppgave/lesionpdf.m

2 of 2

```
62 for j = 1: length(g(:,1))
63     bw2(g(j,2),g(j,1)) = 0;
64 end
65 % If more than one lesion (nr) is present in the image, the nr largest
66 % objects are recognized. If an object covers less than 0.1% of the image,
67 % it is not considered to be a lesion.
68 else
69     bw = zeros(size(bw));
70     for t = 1: nr
71
72         [labeled, num] = bwlabel(bw2,4);
73         data = regionprops(labeled,'Area','PixelList');
74
75         main = find([data.Area] == max([data.Area]));
76
77         Area =[data.Area];
78         mainArea = Area(main)
79         if Area(main) < round(length(r)*0.001)
80             break
81         end
82         g = data(main).PixelList;
83         for j = 1: length(g(:,1))
84             bw(g(j,2),g(j,1)) = 1;
85         end
86
87         for j = 1: length(g(:,1))
88             bw2(g(j,2),g(j,1)) = 0;
89         end
90     end
91 end
92
93 %% Drawing of borders
94 % The binary image with only the lesion object present is filtered with an
95 % average filter. The absolute value between the filtered image and 0.5 is
96 % calculated for each pixel. A binary image with 0.1 as threshold value is
97 % created from the absolute difference image. The black pixels in this
98 % binary image are the border pixels.
99
100 H = fspecial('average',15);
101 blurred = filter2(H,bw2);
102 x2 = abs(blurred - 0.5);
103 c = im2bw(x2,0.1);
104
105 % The binary image consisting of black border pixels on a white background
106 % is returned.
107 lesion = c;
108
```



```
1 function correction = correctionfunc(x0,r0)
2
3 %% This function creates a (2*r0x2*r0) correction matrix based on a single
4 %% RGB image of class uint8. The correction matrix is disk-shaped. The
5 %% input is the RGB image and the radius of the wanted correction disk.
6
7 x0 = imread(x0);           % Reads the image.
8 x0 = double(x0);          % Converts to double.
9
10 % A square image is cropped from the original rectangle image. The size is
11 % (2*(r0+100)x2*(r0+100)).
12 [n m l] = size(x0);
13 r01 = r0+100;
14 x0 = x0(floor(n/2)-r01+1:floor(n/2)+r01,floor(m/2)-r01+1:floor(m/2)+r01,:);
15
16 % Creates black masks.
17 mask = circlefunc(r0);
18 r = find(mask == 1);
19
20 mask2 = circlefunc(r01);
21 r2 = find(mask2 == 1);
22
23 % Converts the rgb-image to a grayscale image.
24 x = rgb2gray(x0/255);
25
26 %% Finds the center of illumination.
27
28 % Creates a binary image using the 0.75-quantile as threshold. Opens the
29 % binary image to avoid outlying pixels of value 1.
30 se = strel('square',5);
31 light = quantile(x(r2),0.75);
32 xbw = im2bw(x,light);
33 xbw = imopen(xbw,se);
34
35 % Finds the center of the square made up by the leftmost, rightmost,
36 % uppermost and lowermost pixel of value 1.
37 [t t1] = find(xbw == 1,1);
38 [t t2] = find(xbw == 1,1,'last');
39 t = t1 + floor((t2-t1)/2);
40
41 [s s1] = find(xbw' == 1,1);
42 [s s2] = find(xbw' == 1,1,'last');
43 s = s1 + floor((s2-s1)/2);
44
45 % Cuts the image to the size of (r0xr0) with the center of illumination
46 % coinciding with the center of the new image.
47 a = s-r0+1; b = 2*r0+a-1; c = t-r0+1; d = 2*r0+c-1;
48 x = x(a:b,c:d);
49 x0 = x0(a:b,c:d,:);
50
51 xpc = rgb2pc(x0,r);       % The principal components are calculated.
52 x = xpc(:,1);            % The first PC is used as grayscale image.
53
54 % Creates a vector, s, containing the mean values of the pixels belonging
55 % to the right half of the image, according to their distance from the
56 % center.
57
58 for r = 1:r0
59
60     y = -r:1:r;
61     x1 = ceil(sqrt(r^2 - y.*y) + r);
```

11/13/08 1:23 PM /Applications/matlab/Masteroppgave/correctionpdf.m

2 of 2

```
62
63 for i = 1: 2*r
64     if x1(i) < r0
65         t1 = x(i+r0-r,x1(i)+r0-r:x1(i)+r0-r+1);
66     else
67         t1 = x(i+r0-r,x1(i)+r0-r-1:x1(i)+r0-r);
68     end
69     t(i) = sum(t1).*0.5;
70 end
71 s(r) = mean(t);
72 end
73
74 % Smooths the vector by moving average with an 11 pixel long window.
75 s = smooth(s,11);
76
77 % Reverses the graph.
78 s = s/max(s);
79 one = ones(r0,1);
80 s = one./s;
81
82 % Sends the graph to interpolation.
83 interpol = interpofunc(s);
84
85 % Creates the correction matrix.
86 matrix = vector2matrix(interpol);
87
88 correction = matrix;
89
```

```
1 function interpol = interpolfunc(s)
2
3 %% Based on a vector,s, and interpolation points, t1-t5, chosen by the
4 %% user, this function returns a new vector based on Newton's
5 %% interpolation. The input is the vector, s.
6
7 r0 = length(s);
8
9 figure, hold on, plot(s), axis([1 1000 0 5])
10
11 % The user chooses the points of interpolation.
12 t1 = input('Choose first point for interpolation ');
13 t2 = input('Choose second point for interpolation ');
14 t3 = input('Choose third point for interpolation ');
15 t4 = input('Choose fourth point for interpolation ');
16 t5 = input('Choose fifth point for interpolation ');
17
18 %% Interpolation by Newton's interpolation algorithm.
19 % The interpolation is done separately on two parts of the vector; first
20 % the part from t1+1 to t3, with interpolating points t1+1,t2 and t3; then
21 % the part from t3+1 to the end of the vector, with interpolating points
22 % t3+1, t4 and t5.
23
24 % Part one.
25
26 w1 = [t1+1 t2 t3];
27 n = length(w1);
28 y(1) = 1;
29 y(2:n) = s(w1(2:n));
30 c = zeros(1,n);
31
32 c(1) = y(1);
33 for k = 2: n
34     d = w1(k)-w1(k-1);
35     u = c(k-1);
36     for i = k-2: -1: 1
37         u = u*(w1(k)-w1(i)) + c(i);
38         d = d*(w1(k)-w1(i));
39     end
40     c(k) = (y(k)-u)/d;
41 end
42
43 r = t1+1:1:t3;
44 p(1:t1) = 1;
45 one = ones(1,length(r));
46 w11 = w1(1)*one;
47 w12 = w1(2)*one;
48
49 p(r) = c(1) + c(2).*(r-w11) + c(3).*(r-w11).*(r-w12);
50
51 % Part two.
52
53 w2 = [t3+1 t4 t5];
54 n = length(w2);
55 y = s(w2);
56 c = zeros(1,n);
57
58 c(1) = y(1);
59 for k = 2: n
60     d = w2(k)-w2(k-1);
61     u = c(k-1);
```

11/13/08 1:31 PM /Applications/matlab/Masteroppgave/interpolf.m

2 of 2

```
62 for i = k-2:-1: 1
63     u = u*(w2(k)-w2(i)) + c(i);
64     d = d*(w2(k)-w2(i));
65 end
66 c(k) = (y(k)-u)/d;
67 end
68
69 r = t3+1:r0;
70 one = ones(1,length(r));
71 w21 = w2(1)*one;
72 w22 = w2(2)*one;
73
74 % The resulting two parts are combined to one vector. The interpolated
75 % vector is shown in the same window as the input vector.
76 p(r) = c(1) + c(2).*(r-w21) + c(3).*(r-w21).*(r-w22);
77 plot(p,'r')
78
79 % Returns the interpolated vector.
80 interpol = p;
```

```
1 function matrix = vector2matrix(s)
2
3 %% This function creates a matrix from a vector. The matrix has the
4 %% dimension (2*r0x2*r0), where r0 is the length of the vector. The pixel
5 %% values of the matrix corresponds to the pixel values of the graph, where
6 %% position in the graph corresponds to distance from origo in the matrix.
7 %% The result is a disk on black (0) background.
8
9 r0 = length(s);
10 x3 = zeros(r0*2,r0*2);
11
12 for r = 1: r0
13
14     y = -r:1:r;
15
16     x1 = ceil(sqrt(r^2 - y.*y) + r);
17     x2 = floor(-sqrt(r^2 - y.*y) + r);
18
19     t1 = find(x1 == 0);
20     x1(t1) = 1;
21     t2 = find(x2 == 0);
22     x2(t2) = 1;
23
24     for i = 1: 2*r
25         if x1(i) < r0
26             x3(i+r0-r,x1(i)+r0-r:x1(i)+r0-r+1) = s(r);
27         else
28             x3(i+r0-r,x1(i)+r0-r-1:x1(i)+r0-r) = s(r);
29         end
30         if x2(i) < r0
31             x3(i+r0-r,x2(i)+r0-r:x2(i)+r0-r+1) = s(r);
32         else
33             x3(i+r0-r,x2(i)+r0-r-1:x2(i)+r0-r) = s(r);
34         end
35     end
36 end
37
38 x4 = imrotate(x3,90,'crop');
39
40 r1 = find(x3 == 0);
41 x3(r1) = x4(r1);
42
43 matrix = x3;
44
```


Appendix B

Instructions

The instructions given to the dermatologists for evaluation of the segmentation.

Results

The evaluation done by dermatologist I and dermatologist II.

I denne permen finner du en samling bilder av føflekker med tilhørende forslag til segmentering. Til venstre er bildet av føflekken slik det ble tatt gjennom dermatoskopet. Over bildet står et tall som du finner igjen i tabellen der du skal skrive inn vurderingen din. Til høyre finner du det samme bildet med grensen til føflekken tegnet opp. Du skal vurdere denne grensa på en skala fra 1 til 4 (1, 2, 3 eller 4), som har følgende verdier

- 1: good
- 2: acceptable
- 3: poor
- 4: bad

Takk for hjelpen.

Kajsa Møllersen

image	grade				image	grade			
expert	I	II	accepted			I	II	accepted	
91	1	1	x		124	1	1	x	
93	1	1	x		126	1	1	x	
95	2	2	x		130	1	1	x	
97	2	2	x		132	3	3	-	
99	1	2	x		133	1	1	x	
101	2	1	x		135	2	2	x	
103	1	1	x		136	2	2	x	
105	2	2	x		137	2	2	x	
107	2	2	x		139	1	1	x	
108	1	1	x		140	3	3	-	
110	2	3	/		144	1	2	x	
112	2	2	x		146	3	3	-	
113	2	2	x		150	3	4	-	
114	3	3	-		152	2	2	x	
116	2	2	x		154	3	4	-	
117	2	1	x		156	1	1	x	
118	1	1	x		157	1	1	x	
119	2	1	x		159	2	2	x	
121	2	1	x		160	2	3	/	
123	2	1	x		163	1	2	x	

image	grade		accepted		image	grade		accepted
	I	II				I	II	
165	1	1	x		195	1	2	x
167	1	1	x		197	1	1	x
168	1	1	x		198	1	2	x
169	1	1	x		199	2	2	x
170	1	1	x		200	1	1	x
172	2	1	x		202	2	1	x
174	1	1	x		204	1	1	x
175	2	2	x		205	1	2	x
176	1	1	x		206	1	1	x
177	1	1	x		208	2	2	x
179	2	1	x		210	1	2	x
181	2	2	x		212	1	1	x
182	4	1			213	1	1	x
184	2	3	/		214	1	2	x
186	1	1	x		216	1	2	x
188	1	1	x		218	2	1	x
189	1	1	x		220	2	2	x
190	2	1	x		222	1	1	x
191	1	1	x		224	1	2	x
193	3	3	-		225	1	2	x

Bibliography

- A. Blum, H. Luedtke, U. Ellwagner, and R. Schwabe and G. Rassner. Digital image analysis for diagnosis of cutaneous melanoma. development of a highly effective computer algorithm based on analysis of 837 melanocytic lesions. *British Journal of Dermatology*, 151:1029–1038, 2004.
- Howard E. Burdick. *Digital Imaging: theory and applications*. The McGraw-Hill Companies, 1997.
- M. Burrioni, P. Sbano, G. Cevenini, M. Risulo, G. Dell’Eva, C. Miracco, M. Fimiani, L. Andreassi, and P. Rubegni. Dysplastic naevus vs. *in situ* melanoma: digital dermoscopy analysis. *British Journal of Dermatology*, 152:679–684, 2005.
- Marco Burrioni, Rosamaria Corona, Giordana Dell’Eva, Francesco Sera, Riccardo Bono, Pietro Puddu, Roberto Perotti, Franco Nobile, Lucio Andreassi, and Pietro Rubegni. Melanoma computer-aided diagnosis: Reliability and feasibility study. *Clinical Cancer Research*, 10:1881–1886, 2004.
- George Casella and Roger L. Berger. *Statistical Inference*. Duxbury, second edition, 2002.
- M. Emre Celebi, Y. Alp Aslandogan, William V. Stoecker, Hitoshi Iyatomi, Hiroshi Oka, and Xiaohe Chen. Unsupervised border detection in dermoscopy images. *Skin Research and Technology*, 13:454–462, 2007.
- Yue (Iris) Cheng, Ragavendar Swamisai, Scott E. Umbaugh, Randy H. Moss, William V. Stoecker, Saritha Teegala, and Subhasini K. Srinivasan. Skin lesion classification using relative color features. *Skin Research and Technology*, 14:53–64, 2008.

- CRN. Cancer registry of norway: Cancer in norway 2006. Technical report, Institute of Population-based Cancer Research, 2006.
- DermLite. URL <http://www.dermlite.com/>.
<http://www.dermlite.com/dl2hr.html>.
- Marek Elbaum, Alfred W. Kopf, Harold S. Rabinovitz, Richard G. B. Langley, Hideko Kamino, Martin C. Mihm, Arthur J. Sober, Gary L. Peck, Alexandru Bogdan, Dina Gutkowitz-Krusin, Michael Greenebaum, Sunguk Keem, Margaret Oliviero, and Steven Wang. Automatic differentiation of melanoma from melanocytic nevi with multispectral digital dermoscopy: A feasibility study. *Journal of the American Academy of Dermatology*, 44:207–218, 2001.
- Matthew G. Fleming, Carsten Steger, Jun Zhang, Jianbo Gao, Armand B. Cognetta, Ilya Pollak, and Charles R. Dyer. Techniques for a structural analysis of dermatoscopic imagery. *Computerized medical imaging and graphics*, 22:375–389, 1998.
- Harald Ganster, Axel Pinz, Reinhard Röhner, Ernst Wildling, Michael Binder, and Harald Kittler. Automated melanoma recognition. *IEEE Transactions on Medical Imaging*, 20:233–239, 2001.
- David Delgado Gomez, Line Harder Clemmensen, Bjarne K. Ersbøll, and Jens Michael Carstensen. Precise acquisition and unsupervised segmentation of multi-spectral images. *Computer Vision and Image Understanding*, 106:183–193, 2007.
- David Delgado Gómez, Constantine Butakoff, Bjarne K. Ersbøll, and William Stoecker. Independent histogram pursuit for segmentation of skin lesions. *IEEE Transactions on Biomedical Engineering*, 55:157–161, 2008.
- Joel Guillod, Philippe Schmid-Saugeon, David Guggisberg, Jean Philippe Cerottini, Ralph Braun, Joakim Krischer, Jean-Hilaire Saurat, and Knut Murat. Validation of segmentation techniques for digital dermoscopy. *Skin Research and Technology*, 8:240–249, 2002.
- Gregory A. Hance, Scott E. Umbaugh, Randy H. Moss, and William V. Stoecker. Unsupervised color image segmentation: with application to

- skin tumor borders. *IEEE Engineering on Medicine and Biology*, 15:104–111, 1996.
- Richard A. Johnson and Dean W. Wichern. *Applied Multivariate Statistical Analysis*. Pearson Education International, fifth edition, 2002.
- David Kincaid and Ward Cheney. *Numerical Analysis: Mathematics of Scientific Computing*. Brooks/Cole, third edition, 2002.
- Tim Lee, Vincent Ng, Richard Gallagher, Andrew Coldman, and David McLean. Dullrazor[®]: A software approach to hair removal images. *Comput. Biol. Med.*, 27:533–543, 1997.
- Tim Kam Lee. *Measuring border irregularity and shape of cutaneous melanocytic lesions*. PhD thesis, Simon Fraser University, 2001.
- Young Won Lim and Sang Uk Lee. On the color image segmentation algorithm based on the thresholding and the fuzzy c-means techniques. *Pattern Recognition*, 23:935–952, 1990.
- Ilias Magliogannis. Automated segmentation and registration of dermatological images. *J. Math. Model. Algorithms*, 2:277–294, 2003.
- Ilias Magliogannis, Elias Zafiroopoulos, and Christos Kyranoudis. Intelligent segmentation and classification of pigmented skin lesions in dermatological images. *Lecture Notes in Computer Science*, 3955:214–223, 2006.
- Wendy L. Martinez and Angel R. Martinez. *Computational Statistics Handbook with Matlab[®]*. Chapman & Hall, first edition, 2002.
- The MathWorks. URL [http://www.mathworks.com/
http://www.mathworks.com/access/helpdesk/help/toolbox/images/
/index.html?/access/helpdesk/help/toolbox/images/rgb2gray.html](http://www.mathworks.com/http://www.mathworks.com/access/helpdesk/help/toolbox/images/index.html?/access/helpdesk/help/toolbox/images/rgb2gray.html).
- Scott W. Menzies. Automated epiluminescence microscopy: human vs machine in the diagnosis of melanoma. *Arch. Dermatol.*, 135:1538–1540, 1999.
- Scott W. Menzies, Leanne Bischof, Hugues Talbot, Alex Gutenev, Michelle Avramidis, Livian Wong, Sing Kai Lo, Geoffrey Mackellar, Victor Skladnev, William McCarthy, John Kelly, Brad Cranney, Peter Lye, Harold

- Rabinovitz, Margaret Oliviero, Andreas Blum, Alexandra Virol, Brian De'Ambrosis, Roderick McCleod, Hiroshi Koga, Caron Grin, Ralph Braun, and Robert Johr. The performance of solar scan: an automated dermoscopy image analysis instrument for the diagnosis of primary melanoma. *Arch. Dermatol.*, 141:1388–1396, 2005.
- W. Keith Nicholson. *Linear Algebra with Applications*. PWS Publishing Company, third edition, 1995.
- Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9:62–66, 1979.
- M. I. Rajab, M. S. Woolfson, and S. P. Morgan. Application of region-based segmentation and neural network edge detection to skin lesions. *Computerized Medical Imaging and Graphics*, 28:61–68, 2004.
- Barbara Rosado, Scott Menzies, Alexandra Harbauer, Hubert Pehamberger, Klaus Wolff, Michael Binder, and Harald Kittler. Accuracy of computer diagnosis of melanoma: a quantitative meta-analysis. *Arch. Dermatol.*, 139:361–367, 2003.
- Pietro Rubegni, Marco Burroni, Gabriele Cevenni, Roberto Perotti, Giordana Dell'Eva, Paolo Barbini, Michele Fimiani, and Lucio Andreassi. Digital dermoscopy analysis and artificial neural network for the differentiation of clinically atypical pigmented skin lesions: A retrospective study. *Journal of investigative dermatology*, 119:471–474, 2002.
- Philippe Schmid. Segmentation of digitized dermatoscopic images by two-dimensional color clustering. *IEEE Transactions on Medical Imaging*, 18:164–171, 1999a.
- Philippe Schmid. Lesion detection in dermatoscopic images using anisotropic diffusion and morphological flooding. In *1999 International Conference on Image Processing*, volume 3, pages 449–453, 1999b.
- Philippe Schmid. *Segmentation and symmetry measure for image analysis: application to digital dermatoscopy*. PhD thesis, École Polytechnique Fédérale de Lausanne, 1999c.

- Philippe Schmid-Saugeon, Joël Guillod, and Jean-Philippe Thiran. Towards a computer-aided diagnosis system for pigmented skin lesions. *Computerized Medical Imaging and Graphics*, 27:65–78, 2003.
- R. Joe Stanley, William V. Stoecker, and Randy H. Moss. A relative color approach to color discrimination for malignant melanoma detection in dermoscopy images. *Skin Research and Technology*, 13:62–72, 2007.
- Wilhelm Stolz, Otto Braun-Falco, Peter Bilek, Michael Landthaler, Walter H. C. Burgdorf, and Armand B. Cognetta. *Color Atlas of Dermatoscopy*. Blackwell Publishing, second edition, 2002.
- Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers, and Keying Ye. *Probability & Statistics for Engineers & Scientists*. Prentice Hall, seventh edition, 2002.
- Paul Wighton, Tim K. Lee, and M. Stella Atkins. Dermoscopic hair disocclusion using inpainting. In *SPIE Medical Imaging*, volume 6914, page 691427, 2008.
- L. Xu, M. Jackowski, A. Goshtasby, C. Yu, D. Roseman, S. Bines, A. Dhawan, and A. Huntley. Segmentation of skin cancer images. *Image and Vision Computing*, 17:65–74, 1999.
- Ezzedine Zagrouba and Walid Barhoumi. A preliminary approach for the automated recognition of malignant melanoma. *Image Anal. Stereol.*, 23: 121–135, 2004.
- Ezzedine Zagrouba and Walid Barhoumi. An accelerated system for melanoma diagnosis based on subset feature selection. *Journal of Computing and Information Technology*, 13:69–82, 2005.
- Hui Zhang, Jason E. Fritts, and Sally A. Goldman. Image segmentation evaluation: a survey of unsupervised methods. *Computer Vision and Image Understanding*, 110:260–280, 2008.
- Zhao Zhang, William V. Stoecker, and Randy H. Moss. Border detection on digitized skin tumor images. *IEEE Transactions on Medical Imaging*, 19:1128–1143, 2000.