PREPROCESSING TECHNIQUES FOR AUTOMATED MELANOMA DETECTION ENHANCEMENT

A Thesis

Presented to

the Faculty of the Department of Telecommunication Engineering

University of Navarra

In Partial Fulfillment

of the Requirements for the Degree

Masters of Science in Telecommunication Engineering

By

Ander Galisteo Zabalo

August 2016

PREPROCESSING TECHNIQUES FOR AUTOMATED MELANOMA DETECTION ENHANCEMENT

Ander Galisteo Zabalo

APPROVED:

Adam Podhorski, Ph.D., Chair
Professor of TECNUN

Xabier Insausti, Ph.D.
Professor of TECNUN

Iñigo Gutiérrez, Ph.D, Deputy head teacher of Tecnun

Acknowledgements

I would like to express my gratitude to Dr. George Zouridakis. He gave me the support I needed to achieve the goals of this project. I am thankful to the people I knew in Houston: Jonathan for showing me that not only engineers can work with Matlab; Hamza for his support during the whole year; Joshua for all his help with scholarship paperwork; Aarushi for beeing a great friend, I wish you the best of lucks and finally Alice and Lee, who made Houston feel like home. I would like to express my sincere gratitude to my family: Gloria for taken care of everyone while I wasn't there; Maddi because I miss her so much during those months; I would like to express my gratitude to my lovely parents. They have been very patient, furthermore these last years. Also they gave me the opportunity to study in Tecnun, to go to USA twice and they still continue supporting me. While I was the one that supported this thesis, Irati has been the one that supported me since I was 15. Please, keep being my pillar for much more. Thanks to everyone in Tecnun that supported my sense of humor: Laura, Juncal, Guille, Jon, Kevin, Vanessa, Amaia, Lorea, Álvaro, Carlos, Laura, Iñaki, Gorka, Sergio,... Thanks.

Ander Galisteo

Abstract

During the last years, technology have experienced a prominent advance in medical devices: devices that are able to make a diagnosis. One of these devices is a melanoma detection application using camera images. Nevertheless, those images may suffer some artifacts that could be avoided or at least mitigated using preprocessing techniques. In this project three preprocessing techniques are evaluated: illumination nonuniformity removal, hair removal and superresolution.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

The skin is the largest organ in the body [34]. It works not only as a first barrier for the immune system but also as a sensor that measures temperature, pressure and humidity. It helps humans gather information from the surroundings and at the same time the skin protects humans from the environment. Nevertheless, although it is very important, the skin is affected by some diseases, cancer being one of the deadliest ones [33]. Although not one of the most common cancers, the presence of this cancer has increased during the two decades [3]. In fact, the average death rate for diagnosed melanomas is around 12% [35], but if melanoma is detected in early stages, the survival rate could be up to 95% [17],[16].

This is why early detection is key to prevent as many deaths as possible. The objective of several medical melanoma detection approaches has been to look for specific features doctors look for. For example, the ABCD rule [36] looks for: Asymmetry, Border smoothness, Color and Diameter or Differentiation.

In some cases, E is also added as "evolution". If the mole changes in size, shape and color, it is more likely to be a melanoma. Nevertheless, the average diagnostic accuracy for visual detection is only 58 % and about 30 % for inexperienced physician, i.e., general dermatologists not specialized in dermoscopy [8],[10]. The future of melanoma detection comes from the automatization of the detection process. Once it is known what to look for, for example the ABCD rule, the idea is to automatize the process in such a way that a machine is able to detect features that belong to melanomas and decide if a given mole is a melanoma or not. Furthermore, it will be even more beneficial if the machine is portable, affordable and easy to use.

For this reason, during the last 10 years, the Biomedical Imaging Lab at the University of Houston has developed a smartphone application that is able to detect melanomas [38] from pictures taken with the phone and only using the phone's computing capabilities [42],[43],[48]. This application has several advantages against other melanoma detection systems [14],[4]:

- Price: This are cheaper.
- Mobility: This system is portable.
- Noninvasive: Because the system relies on image processing (pictures of the skin) is a noninvasive system.

This idea has also been adapted to detect other types of lesions, such as Buruli ulcer [19]. This computerized analysis procedure used different stages to decide if a lesion is a melanoma: several segmentation algorithms with a scoring system to decide

the best one; feature extraction (based on color, shape, texture, structure,...) and selection; and partial classification and decision-fusion [49]. In this thesis another stage has been improved, preprocessing stage. More specifically superresolution, nonhomogeneous illumination removal and hair removal are implemented and tested.

The project is divided into 5 chapters:

- Chapter 2 shows the objectives of this project.
- Chapter 3 shows the theoretical bases of the algorithms used.
- Chapter 4 shows the conclusions, positive and negative aspects of the algorithms and future work to do.
- Chapter 5 shows the budget required for this project.

This project has been developed at the University of Houston under the supervision of Prof. George Zouridakis. He is a Professor and Director of the Biomedical imaging Lab; Engineering Technology; Computer Science; and Electrical and Computer Engineering at University of Houston, Houston, Texas. The supervisor in Spain has been Dr. Adam Podhorski.

Chapter 2

Objectives

The objective of this thesis is to find algorithms in the literature and evaluate their performance, more specifically:

- Superresolution techniques for augmenting the resolution of the images.
- Nonhomogeneous illumination correction.
- Hair removal.

This thesis analyzes each one of these issues, as well as implementation and improvement to currently existing methods and the performance analysis of each one of those methods.

Chapter 3

Methods

In this chapter, the different methods used for obtaining a superresolution image, along with procedures for nonhomogeneous illumination correction and hair removal are reviewed.

3.1 Superresolution

Superresolution (SR) is a technique that uses different sets of information to create high resolution pictures [31]. Those sets of information are usually low resolution pictures (in this thesis this is used) but in some cases a combination of low resolution pictures and infrared data or low resolution pictures and proximity data are used [26].

It must be taken into account that the superresolution explained here is different from the resolution that uses only one picture to create a higher resolution picture

[15]. The main difference is that those techniques are aimed to create pictures with more information than the original pictures, while the superresolution techniques that use only 1 image estimate the value of new pixels from currently existing pixels, but the information they contain is the same as the original one.

The main idea is that the bigger the resolution of the image, the more details of the mole are going to be available and the more accurate the diagnosis is going to be.

The first thing that was found when the literature was reviewed was that super-resolution was used almost exclusively for objects that were relatively far, usually when some details were wanted and they were too pixeled.

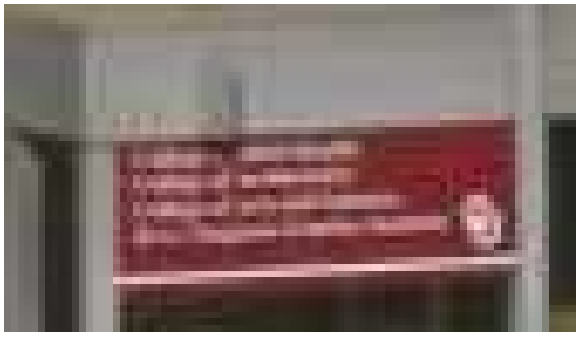


Figure 3.1: Example of a low resolution image in which the text is unreadable [7]

As an example, Figure 3.1 contains a part of a bigger image. Because this picture has been taken from a great distance, the text is unreadable. Taking several pictures (a total of 40) of the same scene and applying superresolution techniques, Figure 3.2 was obtained.

As it can be seen, although there are some artifacts, after superresolution has been applied, it is possible to read what is written.

Figure 3.2: Superresolution image after processing the picture in Figure 3.1

SR techniques combine information from different pictures to obtain one pictures with more information than the others. Because of this, it is assumed that different images contain different information. If the images do not contain different information, that means that they are the exact same picture and the result obtained is not going to be better than a picture obtained using single image superresolution techniques [15].

Different images are formed from taking several pictures of the same object by rotating and moving the camera. While this hypothesis is true when humans take a picture with a camera they are holding, it is not true when a machine does it or when low resolution images are simulated.

Usually, all the superresolution algorithms contain 2 steps: Image registration and Image combination. The first step aligns all the images because all the images are shifted and/or rotated respect to each other. Once they are aligned, information combination algorithms are used to obtain the high resolution images.

There are several techniques that implement superresolution. During the first

stages of the research, a user-friendly GUI was found. This GUI done by Ecole Polytechnique Federale de Lausanne (EPFL) contains several superresolution algorithms[41], [21], [25], [28], [39], [20], [47], [32].

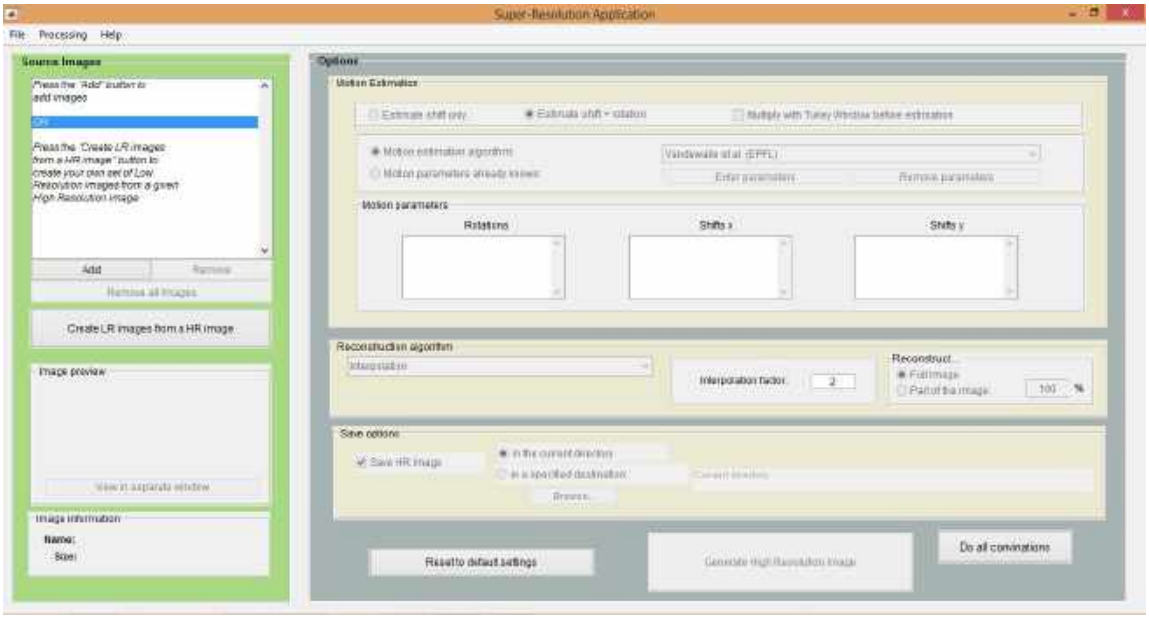


Figure 3.3: GUI for superresolution

In this GUI (Figure 3.3), several motion estimation and superresolution algorithms are available.

In order to see which combination of motion estimation and superresolution is the best for this particular case a synthetic melanoma was created. This synthetic melanoma, shown in Figure 3.4, contains several colors, high and low spatial frequency components and different shapes.

From this image, that was used as the ground truth, several simulated pictures were created. Those pictures are smaller, rotated and shifted versions of Figure 3.4. Notice that Figures 3.5, 3.6, 3.7 and 3.8 are very similar (because all of them come

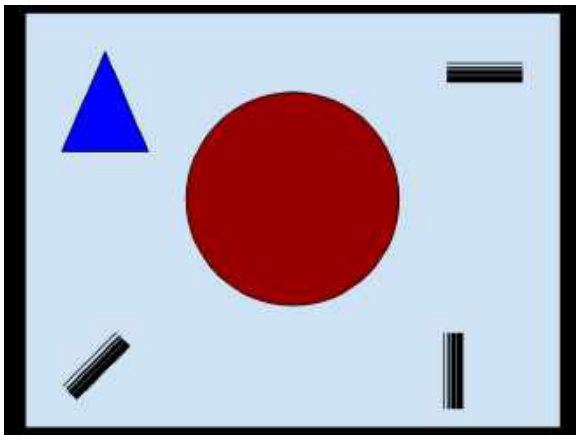


Figure 3.4: Synthetic HR image used in simulations

from Figure 3.4) but different (meaning that contain different information). The rotation and shift of the low resolution images was obtained randomly.

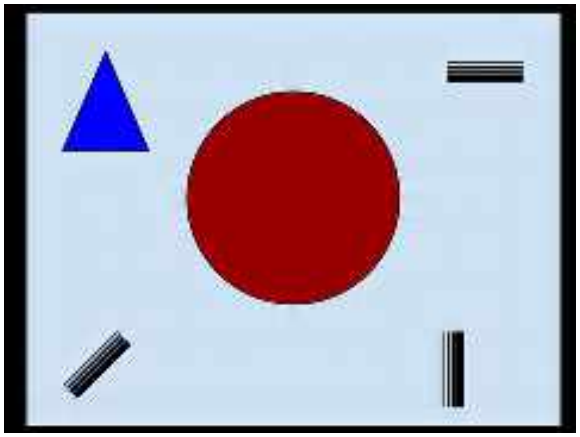


Figure 3.5: Synthetic LR image simulating a picture from Figure 3.4 1

Once the simulated pictures were obtained, the next step was to use all the possible combinations of motion estimation and superresolution algorithms.

Because there were 4 motion estimation and 9 different superresolution algorithms (5 simple algorithms and 1 algorithms with 2 optional features), there were a total

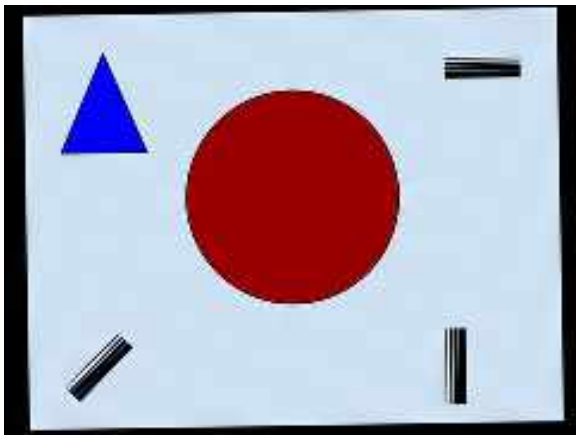


Figure 3.6: Synthetic LR image simulating a picture from Figure 3.4 2

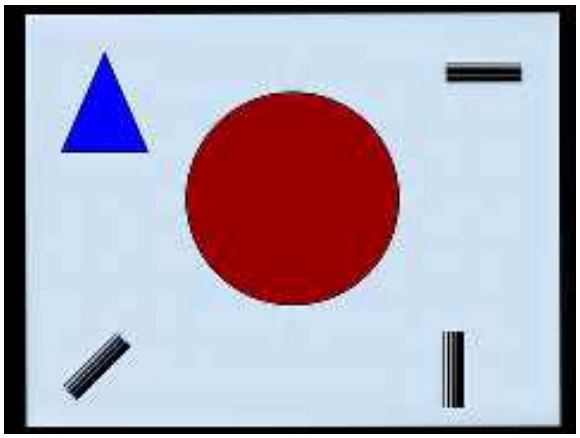


Figure 3.7: Synthetic LR image simulating a picture from Figure 3.4 3

of 36 possible combinations.

Once all the combinations were run, the results were compared with the original high resolution image using the Mean Root Square Error (MRSE) between the original and the results of the SR algorithm.

The best results were obtained by using the combination of Keren [21] motion estimation and interpolation algorithms. The Keren algorithm calculates the rotation and horizontal and vertical shifts required to align the images.

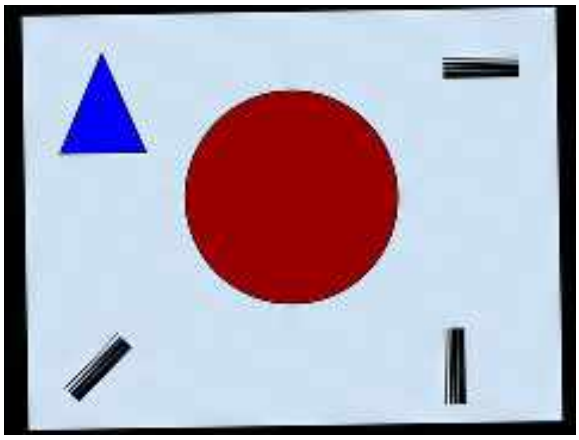


Figure 3.8: Synthetic LR image simulating a picture from Figure 3.4 4

Once the shift and rotations are known, the next method can be used, interpolation. Interpolation is usually used in image processing to calculate the value of some pixels when the image's size changes. For example, if larger images want to be created, that means it is necessary to first create new pixel values. The only information that can be used for creating new pixel values is the value of the pixels that are already known.

3.1.1 Keren subpixel displacement estimation and Interpolation

One way of creating these new pixels is to assign the value of the closest pixel. This is called nearest-neighbor interpolation.

Figure 3.9 shows an example of nearest-neighbor interpolation. For example for the pixel at $(2, 1)$ it can be seen that the value of the pixel is around 5, represented by the orange color. When the nearest-neighbor algorithm is used to increase the

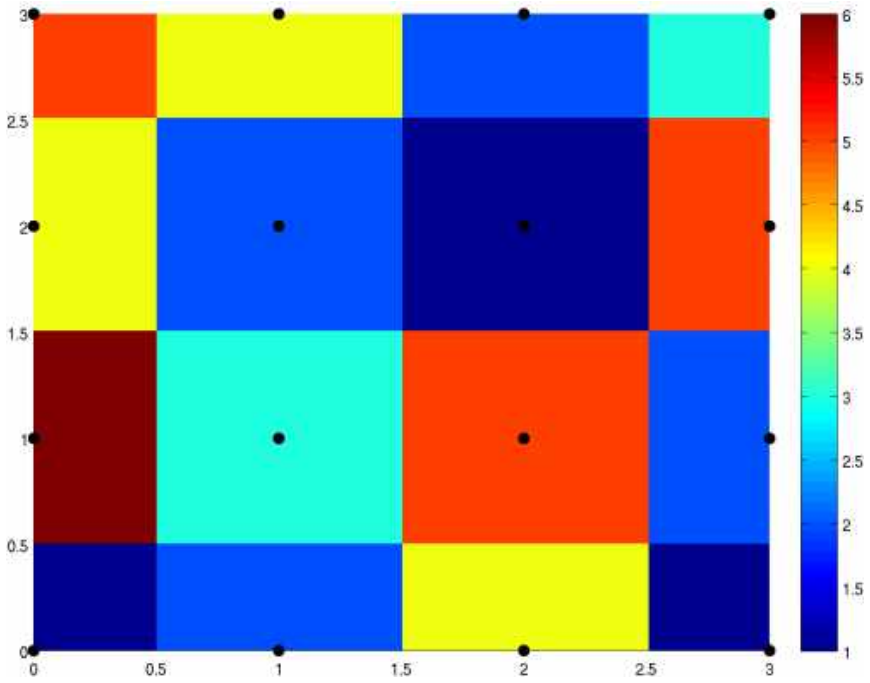


Figure 3.9: Nearest-neighbor interpolation example [46]

number of pixels of the images, all the new pixels created in $x = (1.5, 2.5)$ and $y = (0.5, 1.5)$ will have the same value as the pixel at $(2, 1)$, which is 5. This is why there is an orange square which has the pixel at $(2, 1)$ as the center; that square represents the points that are closer to $(2, 1)$ than to any other pixel. In the particular case that a pixel needs to be created in the exact boundary between two pixels, a decision needs to be made: give it the value of one pixel or the value of the other. As long as the boundary rule is the same in the entire picture, the result will not be the same, but will be equally valid.

Another possible method is to use bilinear interpolation. In the case of bilinear interpolation, the value of the new pixel is not given by the value of the closest pixel but as a combination of the values of the 4 closest pixels, depending on the distance

between the created pixel and the original ones.

Due to the fact that a grayscale image can be represented as a plane, it is easy to see that any new pixel that can be created is going to be contained in a square formed by the closest four pixels (obviously, a pixel is generated outside the edges of the images, this assumption is not going to be true). Those 4 pixels are the ones taken into account when the value of the new pixel is computed.

But, of course, each original pixel is not going to contribute the same to the value of the interpolated pixel; the distance between the interpolated pixel and the original pixel is going to determine the weight of this pixel to the final value of the interpolated one. In other words, the closer a pixel wants to be created from an already existing one, the more similar both are going to be.

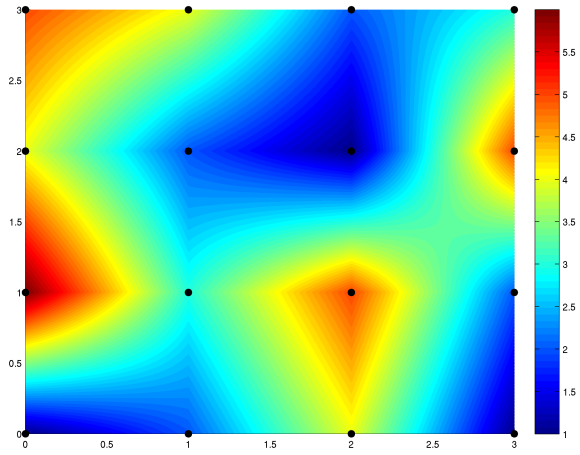


Figure 3.10: Bilinear interpolation example [45]

In Figure 3.10 an example of bilinear interpolation is shown. If a pixel is created exactly at the same distance from 4 pixels, its value is going to be an average of the 4. If a pixel is created really close to another pixel, their values are going to be very

similar. This is why the colors around an original pixel are very similar and they become more different once the generated pixel gets away from a pixel.

Finally, the last interpolation method that is going to be discussed here is bicubic interpolation. The main concept behind bicubic interpolation is the same as bilinear interpolation, but the difference is that while in the bilinear interpolation 4 pixels (2×2) were used to generate the new one, in the bicubic interpolation 16 (4×4) are used. Of course, in this case not only the distances between one pixel and the other are taken into account , but also the value of others pixels. This is why bicubic interpolation takes more time than bilinear, but also gives better results.

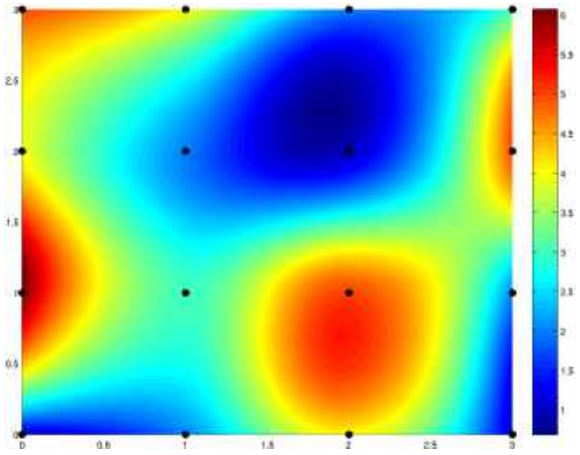


Figure 3.11: Bicubic interpolation example [44]

In Figure 3.11 it can be seen now that the transition between pixels is smoother. The function that represent the surface will have as an input the position and values of the pixels. It must be taken into account that the surface values will have exactly the value of the pixels at the position of the pixels.

When using interpolation with one image,"superpixeled" images are created, but

not SR images. The information contained inside the Low Resolution (LR) image and the SR image is the same. This makes sense, because only used the information contained in the LR image to create the new pixels has been used.

Usually, when image registration is performed, the images are rotated, shifted and scaled to align them to a reference, losing part of the information of the image. Some other times, it can be assumed that there is an infinite amount of space and one image after the other can be overlaid until all the information is on the same space (as it can be seeing in Figure 3.12).

Figure 3.12: Image registration assuming an infinite amount of space [30]

Another concept that needs to take into account is that interpolation does not only work with a regular data grid, it also works with irregular data grids too.

In all the examples shown before, the distance between all the original pixels was the same and this makes sense when images are used; the pixels in an image are regularly distributed.

Nevertheless, images can be represented as a set of points in a 3D space. The

horizontal and vertical distances could be the value of the x and y axis, while the value of the pixel's intensity could be the value of the z axis.

The benefit of thinking of pixels as points in the space is that as between two neighboring pixels there could be no pixel, but between two scattered neighboring points in a space there could be infinite points.

The registration parameters were calculated using Keren's subpixel displacement estimation algorithm [21]. The idea behind this algorithm is that two images of the same object are identical apart from the displacement and rotation:

$$g(x, y) = f(x\cos(\theta) - y\sin(\theta) + a, y\cos(\theta) + x\sin(\theta) + b) \tag{3.1}$$

Where a and b are the horizontal and vertical misalignments respectively, and θ is the rotation. If the first two terms of the Taylor series are computed:

$$g(x, y) \approx f(x + a - y\theta - x\theta^2/2, y + b + x\theta - y\theta^2/2) \tag{3.2}$$

And then expanding f to the first term of its own Taylor series:

$$g(x, y) \approx f(x, y) + (a - y\theta - x\theta^2/2)\frac{\partial f}{\partial x} + (b + x\theta - y\theta^2/2)\frac{\partial f}{\partial y} \tag{3.3}$$

So the error function between g and f can be approximated to:

$$E(a,b,\theta) = [\sum f(x,y) + (a - y\theta - x\theta^2/2)\frac{\partial f}{\partial x} + (b + x\theta - y\theta^2/2)\frac{\partial f}{\partial y} - g(x,y)] \tag{3.4}$$

Then, taking the derivation of this term for a, b and θ and making them 0, a 3 equation with 3 unknown system is obtained. In order to solve it this faster, a coarse-to-fine structure of the image is used, called a Gaussian pyramid. The algorithm is explained in Appendix A.

Due to the fact that the shift and rotation between images are known, all the points of all the images can be represented into one space. If the rotation and shift between the images is small in respect to the pixel shift, there will be clusters of points. If, on the other hand, the rotation and shifts are large, the points will be nonuniformly spread through all the space. Taking this into account, a regular data grid could be created, using interpolation.

Figures 3.13a, 3.13b, 3.13c and 3.13d are 4 low resolution images of the same object, in this case a castle.

Figure 3.14 shows the result of the superresolution interpolation algorithm. The result shows higher spatial detail than the original low resolution images.

(a) LR picture of a castle 1

(b) LR picture of a castle 2

(c) LR picture of a castle 3

(d) LR picture of a castle 4

Figure 3.13: LR pictures of a castle [5]

Figure 3.14: Castle SR result usign Keren estimation algorithm and Interpolation

3.1.2 Iterative process

Another superresolution algorithm used on lens free on-chip microscope [9] was explored, which was based on a different superresolution algorithm [6]. This algorithm works taking into account models of the scene captured and the camera.

Modeling of the scene:

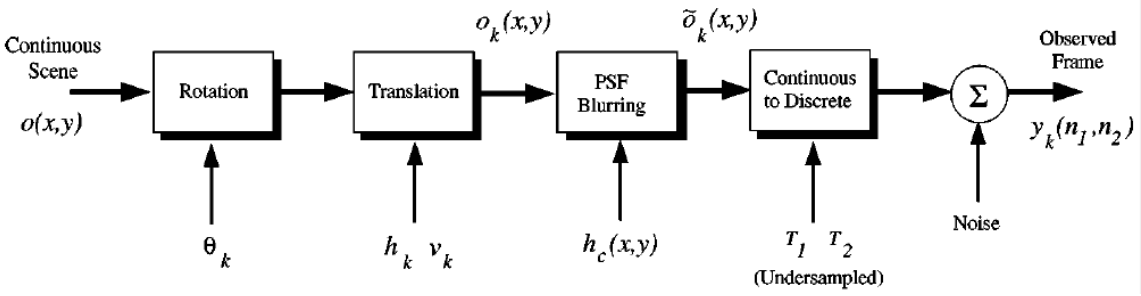


Figure 3.15: Continuous observation model resembling the physical process of image acquisition [6]

Figure 3.15 shows how the images are acquired. In the acquisition process there are several steps.

1. Translation and rotation of the scene.
2. Blurring of the scene due to optics.
3. Discretization of the continuous scene.
4. Noise (added by several factors).

The reality that is being captured is $o(x, y)$. After rotation and translation the scene would be

$$o_k(x, y) = o(xcos\theta_k - ysin\theta_k + h_k, ycos\theta_k + xsin\theta_k + v_k) \tag{3.5}$$

where θ_k represents the rotation and h_k and v_k the horizontal and vertical translations, respectively, for the k-th observable frame. Assuming that the scene $o(x, y)$ does not change while all the frames are being observed, after rotation and translation, a blurring effect is applied to the scene, modeled as

$$\tilde{o}_k(x, y) = o_k(x, y) * h_c(x, y) \tag{3.6}$$

where $h_c(x, y)$ is the continuous system Point Spread Function. Then the result is sampled and noise is added.

$$y_k(n_1, n_2) = \tilde{o}_k(n_1T_1, n_2T_2) + \eta_k(n_1, n_2) \tag{3.7}$$

where T_1 and T_2 are the horizontal and vertical sample spacing and the $\eta_k(n_1, n_2)$ is an additive noise term.

Defining the low resolution image $y_k(n_1, n_2) = N_1 \times N_2$.

Each pixel on the low resolution image can be defined as $\mathbf{y}_k = [y_{k,1}, y_{k,2}, y_{k,3} \dots y_{k,M}]^T$ where $M = N_1 N_2$. If p is the number of observable frames:

$$\mathbf{y} = [\mathbf{y}_1^T, \mathbf{y}_2^T, \mathbf{y}_3^T \dots \mathbf{y}_p^T]^T = [y_1, y_2, y_3 \dots y_{pM}]^T \tag{3.8}$$

where $\mathbf{y}$ contains all the observable pixels.

The the discrete model of the scene is defined, in order to relate the high resolution pictures with the continuous scene:

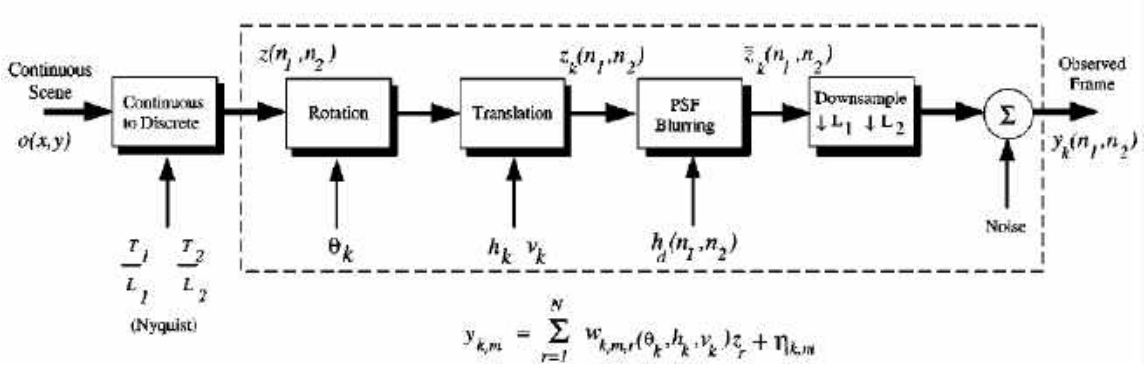


Figure 3.16: Equivalent discrete observation model illustrating the relationship between the ideally sampled image $\mathbf{z}$ and the observed frames $\mathbf{y}$ [6]

In this case too, several modules are defined:

1. Discretization of the continuous scene. Its output $z(n1, n2)$ is discrete but ideal.

2. Translation and rotation of the scene.

3. Blurring of the scene due to optics.

4. Downsampling from the high resolution to low resolution.

5. Noise (added by several factors).

The size $L_1N_1 \times L_2N_2 = N$ where L_1 and L_2 are positive integers that represent the size increasing factor horizontally and vertically, respectively.

Putting all the image in an array $\mathbf{z} = [z_1, z_2, \ldots, z_N]^T$.

Once again, θ_k represents the rotation and h_k and v_k the horizontal and vertical translations of the k-th observed low resolution frame, creating $z_k(n_1, n_2)$.

Next, the blurring is applied once again:

$$\tilde{z}_k(n_1, n_2) = z_k(n_1, n_2) * h_d(n_1, n_2) \tag{3.9}$$

Then the high resolution image is downsampled and noise added.

$$y_k(n_1, n_2) = \tilde{z}_k(Ln_1, Ln_2) + \eta_k(n_1, n_2) \tag{3.10}$$

This model can be re-written for all the pixels of all the low resolution frames as:

$$y_{k,m} = \sum_{r=1}^{N} w_{k,m,r}(\theta_k, h_k, v_k) z_r + \eta_{k,m} \tag{3.11}$$

for $m = 1, 2, ..., M$ and $k = 1, 2, ..., p$. $w_{k,m,r}$ represents the weight of the r-th high resolution pixel on the m-th low resolution pixel of the k-th observed frame.

The main problem here is how to calculate those weights. In order to do it, the system Point Spread Function (PSF) is going to be used, which has 2 components.

- The shape of the detector array.
- The optical transference function of the system.

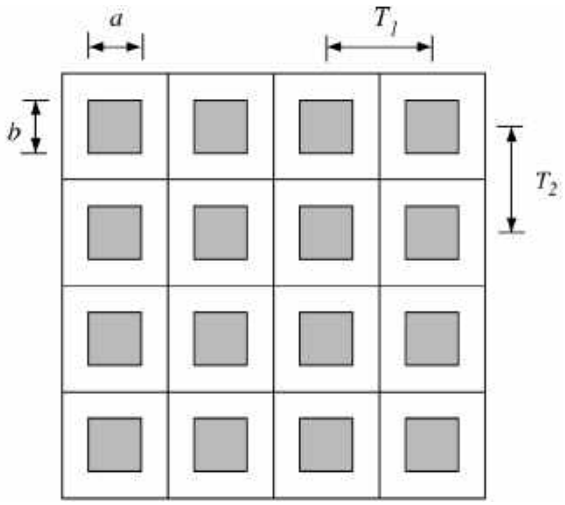


Figure 3.17: Detector's model [6]

The detector array model is shown in Figure 3.17.

The detector's mathematical model is given by

$$d(x,y) = \frac{1}{ab} rect(\frac{x}{a}, \frac{y}{b}) = \begin{cases} 1 \text{ for } |x/a| < 1/2 \text{ and } |y/b| < 1/2 \\ 0 \text{ otherwise} \end{cases} \tag{3.12}$$

$D(u,v)$ is defined as:

$$D(u,v) = \mathscr{F}\{d(x,y)\} \tag{3.13}$$

Then if u and v are the horizontal and vertical frequencies,

$$D(u,v) = \text{sinc}(au,bv) = \frac{\sin(\pi au)\sin(\pi bv)}{\pi^2 aubv} \tag{3.14}$$

The incoherent optical transfer function (OTF) of diffraction-limited optics with a circular exit pupil can be found as;

$$H_o(u,v) = \begin{cases} \frac{2}{\pi}\{\cos^{-1}(\frac{\rho}{\rho_c}) - \frac{\rho}{\rho_c}[1-(\frac{\rho}{\rho_c})^2]^{1/2}\} \text{ for } \rho < \rho_c \\ 0 \text{ otherwise} \end{cases} \tag{3.15}$$

where $\rho = (u^2+v^2)^{1/2}$ and $\rho_c = \frac{1}{\lambda f/\#}$ where $f/\#$ is the f number of the optics and λ is the wavelength to be considered.

So the total system frequency response will be $H(u,v) = D(u,v) * H_o(u,v)$ and the spatial response:

$$h_c(x,y) = \mathscr{F}^{-1}\{H(u,v)\} \tag{3.16}$$

And the discrete system:

$$h_d(n_1, n_2) = \frac{T_1 T_2}{L_1 L_2} h_c(\frac{n_1 T_1}{L_1}, \frac{n_2 T_2}{L_2}) \tag{3.17}$$

.

In this particular case, $f = 2.4$, $a = b = 1.12 \times 10^{-3}$ mm, $\lambda = 5.5 \times 10^{-6}$ mm and $T_1 = T_2 = 1.12 \times 10^{-3}$ mm.

This last value has been calculated dividing the sensor size by the number of pixels and shows that all the pixels are one after the other with no gap between them.

Of course, it is virtually impossible to calculate the discrete ideal scene $\mathbf{z}$, but it may be possible to calculate $\hat{\mathbf{z}}$ that is defined as

$$\hat{\mathbf{z}} = \text{argmin} C(\mathbf{z}) \tag{3.18}$$

where

$$C(\mathbf{z}) = \frac{1}{2} \sum_{m=1}^{pM} (y_m - \sum_{r=1}^{N} w_{m,r} z_r)^2 + \frac{\lambda}{2} \sum_{i=1}^{N} (\sum_{j=1}^{N} \alpha_{i,j} z_j)^2 \tag{3.19}$$

The first term of $C(\mathbf{z})$ represents the difference the observed low resolution pictures and the downsampled versions of the high resolution image. Of course, if the data in the high resolution image contains information from all the low resolution frames and the weights are correctly selected, theoretically it would be possible to

reconstruct the low resolution images from the high resolution one, making this term 0. Nevertheless, in reality there could be several images that could minimize this term due to the ill-posed nature of the inverse problem. In order to solve this, the second term was created, to regularize the result, where λ represents the weight of the second term. $\alpha_{i,j}$ is such that it will be minimized when the result is smooth, meaning that there will be little high resolution noise. $\alpha_{i,j}$ is defined as

$$\alpha_{i,j} = \begin{cases} 1 \text{ for } i = j \\ -1/4 \text{ for } j : z_j \text{ is a cardinal neighbor of } z_i \end{cases} \tag{3.20}$$

In order to optimize the cost term the first step is to differentiate the cost term for each one of the pixels z_k for $k = 1, 2, 3 \ldots, N$

$$g_k(\mathbf{z}) = \frac{\partial C(\mathbf{z})}{\partial z_k} = \sum_{m=1}^{pM} w_{m,k}(\sum_{r=1}^{N} w_{m,r} z_r - y_m) + \lambda \sum_{i=1}^{N} \alpha_{i,k}(\sum_{j=1}^{N} \alpha_{i,j} z_j) \tag{3.21}$$

In order to have the differentiation of all the pixels in one variable,

$$\mathbf{g}^n = \begin{bmatrix} g_1(\hat{\mathbf{z}}^n) \\ g_2(\hat{\mathbf{z}}^n) \\ \vdots \\ g_N(\hat{\mathbf{z}}^n) \end{bmatrix} \tag{3.22}$$

Now the conjugate gradient vector can be defined as:

$$\mathbf{d}^{n+1} = -\mathbf{g}^{n+1} + \beta^n \mathbf{d}^n \tag{3.23}$$

where

$$\beta^n = \frac{(\mathbf{g}^{n+1})^T \mathbf{g}^{n=1}}{(\mathbf{g}^n)^T \mathbf{g}^n} \tag{3.24}$$

and

$$\mathbf{d}^0 = -\mathbf{g}^0 \tag{3.25}$$

Now iterative process can be defined as:

$$\hat{\mathbf{z}}^{n+1} = \hat{\mathbf{z}}^n + \varepsilon^n \mathbf{d}^n \tag{3.26}$$

Where ε can be defined as the optimal step size obtained by minimizing $C(\hat{\mathbf{z}}^{n+1}) = C(\hat{\mathbf{z}}^n + \varepsilon^n \mathbf{d}^n)$ which yields to

$$\varepsilon^n = -\frac{\sum_{m=1}^{pM} \phi_m (\sum_{r=1}^{N} w_{m,r} \hat{z_r}^n - y_m) + \lambda \sum_{i=1}^{N} \bar{d}_i (\sum_{j=1}^{N} \alpha_{i,j} \hat{z}_j)}{\sum_{m=1}^{pM} \phi_m^2 + \lambda \sum_{i=1}^{N} \bar{d}_i^{\,2}} \tag{3.27}$$

where

$$\phi_m = \sum_{r=1}^{N} w_{m,r} d_r(\hat{\mathbf{z}}^n) \tag{3.28}$$

and

$$\bar{d}_i = \sum_{j=1}^{N} \alpha_{i,j} d_j(\hat{\mathbf{z}}^n) \tag{3.29}$$

Before starting the iterations some kind of high resolution image $\hat{\mathbf{z}}^0$ will be needed. This image could be obtained by taking the interpolation of one of the frames.

As an example, the images from Figure 3.14 were used with 10 iterations. In Figure 3.18 the estimated high resolution image, $\hat{\mathbf{z}}^{10} \approx \mathbf{z}$ was calculated

Figure 3.18: Iterative process' result

3.2 Non-homogeneous illumination correction

It is well known that optical images are often affected by inhomogeneous illumination. More specifically, it is known that at the center of the image, the illumination is higher than at the edges. As it can be seen in Figure 3.19, in the center of the image, the illumination is homogeneous, but it can clearly be seen that the top right corner is darker than in the center. The device that is used to take the images has some LEDs that are used to illuminate the skin but it makes the center of the image (where the useful information about the lesion is) brighter.

Figure 3.19: Melanoma with nonhomogeneous illumination

Nevertheless melanoma detection is done by looking for some features on the image. Some of those features could be affected by the nonhomogeneous illumination of the image [37], and this could make the algorithm to detect melanoma features on a healthy mole and the other way around.

The first thing that needs to be done to understand this well known procedure is to model the nonhomogeneous illumination [11]. This is why the following equation is described:

$$\mathrm{y}(x, y) = \mathrm{u}(x, y) * \mathrm{i}(x, y) \tag{3.30}$$

Where y represents the acquired image, u is the illumination mask and i is the real image without any non-homogeneous illumination problems.

In order to approach the problem, two possible solutions where used:

- Homomorphic filtering.
- Mathematical estimation of the nonhomogeneous illumination.

3.2.1 Homomorphic filtering

Homomorphic filtering is a technique used in image processing to enhance image contrast and normalize brightness [12]. It requires a mapping of the illumination to another non-linear domain. Then, after the filter is applied, the result is mapped back to the linear illumination space. It is used usually with grayscale biomedical images, such as MRIs and cell pictures.

As it can be seen in Figure 3.20, the illumination is homogeneous after the homomorphic filtering is applied.

Because RGB images are being used, the first step is to transform the image into a space where the color and the illumination components are separated. After that, only the illumination component is going to be used.

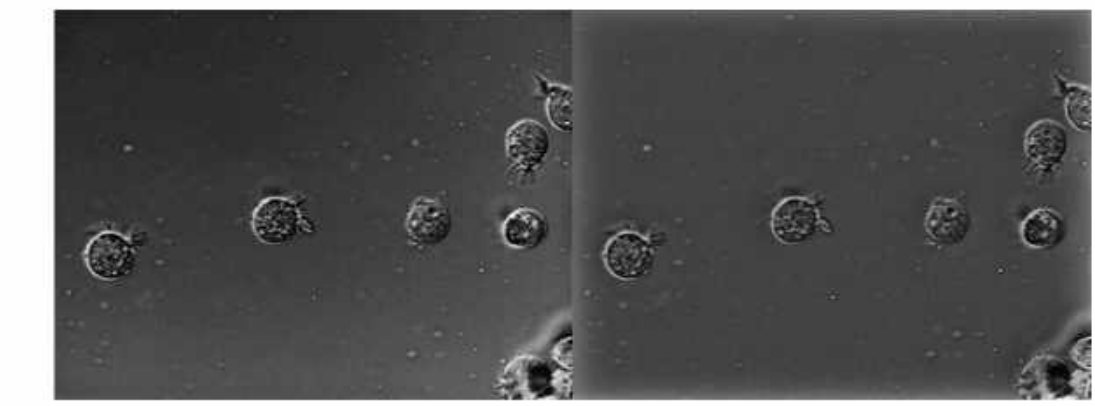


Figure 3.20: Right: Before filtering [29], Left: After filtering

In order to separate the illumination component from the color components, knowing the RGB components:

$$\begin{aligned} Y &= 0.299R + 0.587G + 0.114B \\ U &= -0.14713R - 0.28886G + 0.463B \\ V &= 0.615R - 0.55861G - 0.05639B \end{aligned} \tag{3.31}$$

where RGB represent the red, green and blue color channels respectively. Also, Y refer to the illumination and UV to the chrominance. In order to go from YUV to RGB the following transformation is used:

$$\begin{aligned} R &= Y + 1.13983V \\ G &= Y - 0.39465U - 0.5806V \\ B &= Y + 2.03211U \end{aligned} \tag{3.32}$$

Homomorphic filtering is applied to the illumination component. Consequently

when the final RGB image is created, the original U and V channels are used, without any filtering or modification.

In order to perform the homomorphic filtering, the following image model is going to be used:

$$\mathrm{I}(x, y) = \mathrm{L}(x, y) * \mathrm{R}(x, y) \tag{3.33}$$

where I is the image, L is the illumination of the scene and R is the reflectance of the scene. The first component is based on how the scene is being illuminated and the second one on how the scene is because of its physical structure, not related to how it is illuminated. Therefore, this model will work to remove the illumination nonhomogeneity. The only thing that needs to be completed is to remove the L component of the image, the illumination.

For now, it is assumed that the illumination is a low frequency component. For this particular case, the illumination of the scene does not change quickly, but the image does. This means that the illumination changes slower than the scene. A melanoma has a lot of small details, meaning that it has high frequency components.

So, it can be said that:

- L has low frequency components
- R has high frequency components

Nevertheless, a problem is faced at this point. Because illumination of the scene

and the reflectance of the object are multiplying, the space needs to be transformed before being able to separate both components.

This is why the following transformation is performed:

$$ln(\mathrm{I}(x,y)) = ln(\mathrm{L}(x,y) * \mathrm{R}(x,y)) = ln(\mathrm{L}(x,y)) + ln(\mathrm{R}(x,y)) \tag{3.34}$$

Then, the Fourier transform of both sides is done:

$$\begin{aligned} \mathscr{F}\{ln(\mathrm{I}(x,y))\} = \mathscr{F}\{ln(\mathrm{L}(x,y)) + ln(\mathrm{R}(x,y))\} = \\ \mathscr{F}\{ln(\mathrm{L}(x,y))\} + \mathscr{F}\{ln(\mathrm{R}(x,y))\} \end{aligned} \tag{3.35}$$

Now that both components are separated, a high-pass filter can be applied:

$$\begin{aligned} H(u,v) * \mathscr{F}\{ln(\mathrm{I}(x,y))\} = H(u,v) * \mathscr{F}\{ln(\mathrm{L}(x,y)) + ln(\mathrm{R}(x,y))\} = \\ = H(u,v) * \mathscr{F}\{ln(\mathrm{L}(x,y))\} + H(u,v) * \mathscr{F}\{ln(\mathrm{R}(x,y))\} \end{aligned} \tag{3.36}$$

where H is the filter and u and v the logarithmic frequencies. Because L only has low frequencies and R only high frequencies:

$$H(u,v) * \mathscr{F}\{ln(\mathrm{I}(x,y))\} = \mathscr{F}\{ln(\mathrm{R}(x,y))\} \tag{3.37}$$

Now applying the transformation back:

$$e^{\mathscr{F}^{-1}\{H(u,v)*\mathscr{F}\{ln(\mathrm{I}(x,y))\}\}} = e^{\mathscr{F}^{-1}\{\mathscr{F}\{ln(\mathrm{R}(x,y))\}\}} = e^{ln(\mathrm{R}(x,y))} = \mathrm{R}(x,y) \tag{3.38}$$

obtaining the image that was wanted.

The algorithm was tested. In order to do it, a synthetic image was created.

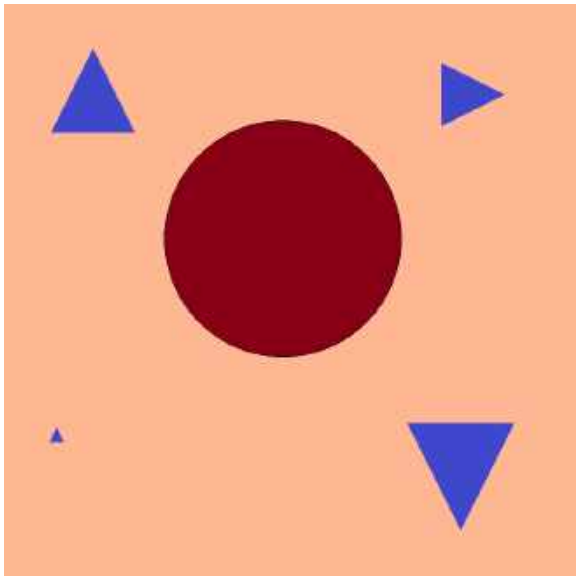


Figure 3.21: Synthetic image which contains high and low frequency components and different colors

Figure 3.21 shows the synthetic image used. It has some high frequency features, some low frequency features and different colors. This may not be the most accurate representation of a melanoma, but it is accurate enough to see if the homomorphic filtering works.

After creating the image, in which illumination is completely homogeneous, some illumination irregularities were simulated.

Basically 3 types of inhomogeneities were created: horizontal inhomogeneities, vertical inhomogeneities and circular inhomogeneities.

Because it is assumed that shadows are a multiplicative process, they must also

be added to the image by multiplying the mask to the image.

In Figures 3.22 and 3.23 two different inhomogeneities masks can be seen.

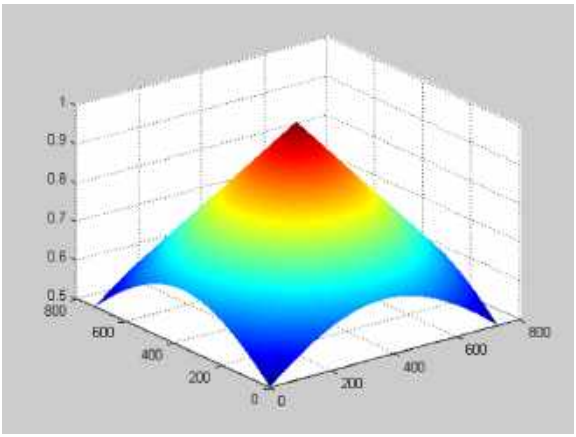


Figure 3.22: Circular shadow mask

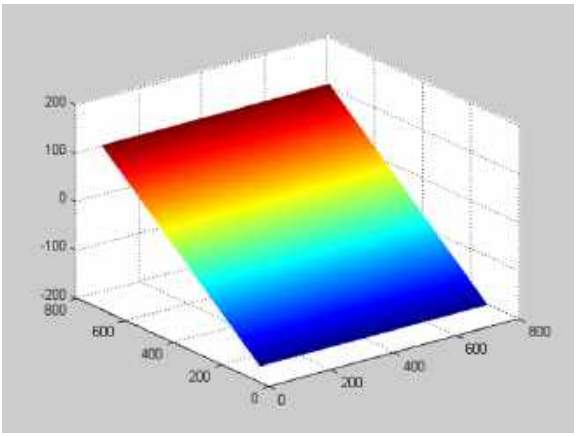


Figure 3.23: Horizontal shadow mask

When the inhomogeneities are added (Figures 3.22 and 3.23) to the original image (Figure 3.21) the following images (Figures 3.24 and 3.25) are obtained.

After applying the homomorphic filtering, the images obtained can be seen in Figures 3.26 and 3.27). As it can be seen, the homomorphic filtering does not only

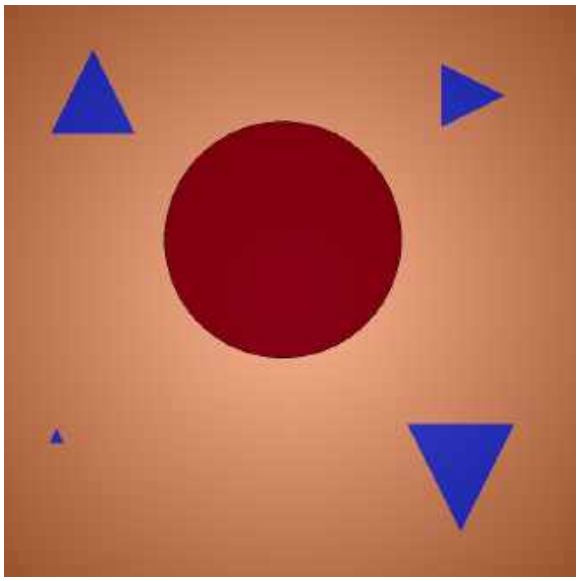


Figure 3.24: Synthetic melanoma with circular nonhomogeneous illumination mask

correct the mask, but also modifies the color of the background and the inside of the melanoma. Each effect can be explained separately.

The first one, the color difference that is seen, is due to the fact that the image is being filtered. In fact if any signal (audio, video, electromagnetic and even an image) goes through a passive filter, the energy at the output of the filter is always smaller or equal than the energy at the input of the filter. This is why the images are darker than what it could be expected.

Nevertheless, it can clearly be seen that in Figures 3.26 and 3.27 there is not only a change in the illumination pattern but also a "halo" around the melanoma. This is due to the fact that the homomorphic filtering has removed the low frequency components, but on the edges of the melanoma, the illumination pattern changes quickly.

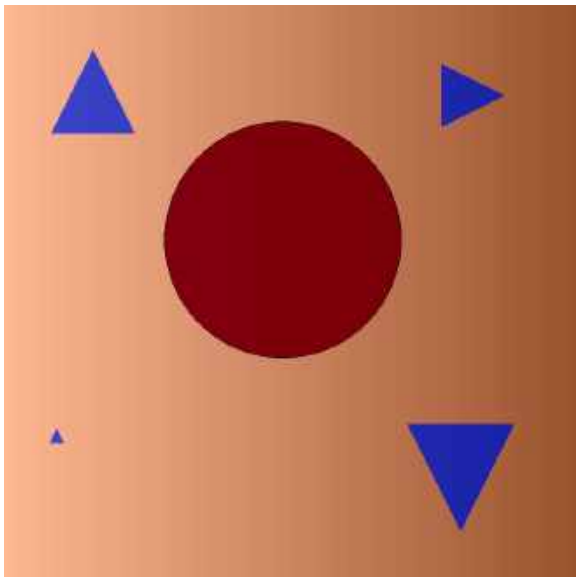


Figure 3.25: Synthetic melanoma with horizontal nonhomogeneous illumination mask

Also, in the center of the melanoma, some of the color is removed. This is due to the fact that one of the assumptions made was not correct.

It was assumed that the image was separable into two components (Equation 3.33) and that each of those components had a different origin. Nevertheless, although the inhomogeneous illumination can be considered as a low frequency component, the lesion has some high frequency and some low frequency components. What the homomorphic filtering does is to remove all the low frequency components, including the ones inside the lesion. This is why the image features are altered and why this method can not be used to remove the illumination nonhomogeneity.

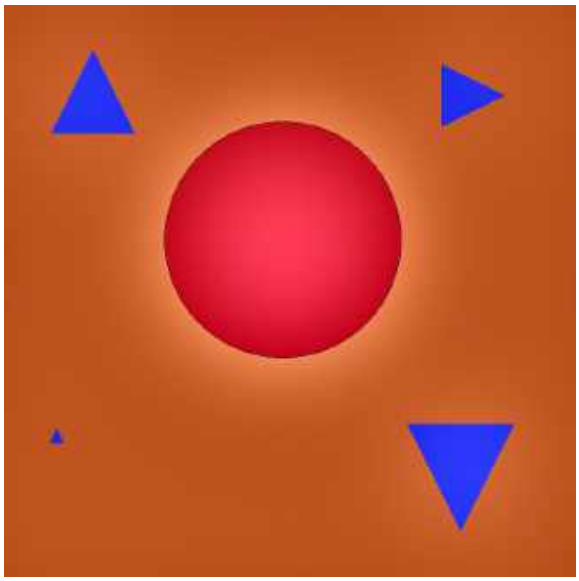


Figure 3.26: Result of the homomorphic filtering on a synthetic melanoma with circular nonhomogeneous illumination mask

3.2.2 Regression model

In the previous method, the inhomogeneous illumination mask was tried to be filtered [13]. In this method, the mask is going to be approached to a mathematical model, and then compensated (not removed).

The main idea here is that the image has two components:

$$\mathrm{I}(x, y) = \mathrm{M}(x, y) * \mathrm{R}(x, y) \tag{3.39}$$

where I represents the image captured by the camera, M represents the illumination mask and R represents the real background image without illumination homogeneity. What it wants to be done is to, somehow, estimate the illumination mask.

It is assumed that M mask is known:

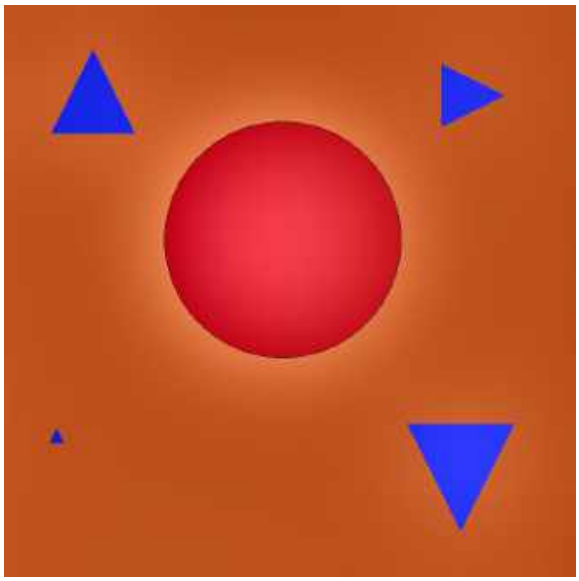


Figure 3.27: Result of the homomorphic filtering on a synthetic melanoma with horizontal nonhomogeneous illumination mask

$$\mathrm{I}(x,y)/\mathrm{M}(x,y) = \mathrm{M}(x,y) * \mathrm{R}(x,y)/\mathrm{M}(x,y) = \mathrm{R}(x,y) \tag{3.40}$$

In this case, it would be easy to compute the image with homogeneous illumination.

The problem relies on computing the M illumination mask. This is why some assumptions were taken:

- The image acquired has a melanoma and healthy skin.
- The healthy skin contains little details.
- The illumination changes slowly.

Mathematically, the first assumption could be written as

$$R = M \cup S \text{ and } M \cap S = \emptyset \tag{3.41}$$

where R is the real image without the non-homogeneous illumination, M is the melanoma and S is the healthy skin. As it can be seen, it is assumed that both sets are disjointed (no element is in both sets) and the union of both sets creates R (any element is M or S is in R).

The second assumption says that the pixels that belong to the healthy skin have mostly high frequency components.

The third one assures that the illumination has low frequency components.

Taking these three assumptions into account and due to the fact that the illumination mask wants to be computed, the following procedure can be used.

The main concept about this method is that the surrounding healthy skin is going to be used in order to compute the illumination mask.

The first step is to perform the segmentation of the melanoma and the healthy skin. In order to do it, a simple threshold method is used, already implemented in the Melanoma detection application [16].

Once the melanoma and the healthy skin are separated, the next step is to calculate the illumination mask. In order to do it, two different methods were used, splines and regression models.

It must be taken into account that any method used is going to predict accurately

the nonhomogeneous illumination mask of the healthy skin, because this has been the information used to predict the mask. What is interesting and useful is to predict the nonhomogeneous illumination of the melanoma using the healthy skin.

Splines are piecewise defined curves using polynomials. They were used in a previous project for calculating the surface of the retina having some points.

The main problem is that splines do not behave well under these requirements. In the retina surface calculation project [24], the amount of points per retina was small (around 18 per retina) and the points were almost uniformly distributed along the retina. This means that the gaps between the points were small, giving little freedom to the splines to have strange behaviors.

Nevertheless, this case is completely different. In this case, the image has up to hundreds of thousands of pixels (hundred of thousands of points the image is considered a set of points in the space) and in the middle of the image there is a large hole relative to the size of an image.

For this, the tests performed show that the nonhomogeneous illumination mask is correctly computed for the healthy skin, but not for the melanoma. The assumption taken was that this happens because splines work well when they are restricted. This means that they work good for creating curves between given points that are close to each other. If they are too separated, the spline has too much freedom and the behavior is no longer the one wanted.

On the other hand, it was tried to fit the healthy skin to a second order 2 dimensional polynomial surface. This means that all the pixels that belong to the healthy

skin are going to be taken and a shadow mask with the following formula is going to be calculated:

$$z(x, y) = a_0 + a_{10}x + a_{02}y + a_{20}x^2 + a_{11}xy + a_{02}y^2 \tag{3.42}$$

where the values $a_0, a_{10}, \ldots, a_{02}$ are chosen in such a way that the difference between the healthy skin and $z(x, y)$ is minimized.

In order to do this, the fit Matlab function was used. This function allows to do a regression model using different fit types. Each fit type is a different approach to do this. In this case, due to the fact that a second order polynomial is used, the fit type is going to be poly22.

The methods that are not polynomials work well with small gaps between values and when the gap is surrounded by points. In this case, this does not work, due to the fact that the gap is significant.

Also, one of the assumptions is that the nonhomogeneous illumination contains only low frequency components. Other methods, such as interpolation or splines, work in smaller areas, meaning that they will contain higher frequency components.

Along the polynomials, several tests were performed to see which algorithm to use. The fit function allows to have different degrees in x and y. As a requirement, due to symmetry, the degrees of freedom of x and y should be the same. This allows 5 possibilities, due to the fact that the maximum degree of the polynomial for the fit function is 5 in both dimensions.

Here, it is useful to think that the illumination mask is a metal surface. Practically, the higher the degree of the polynomial, the more "bending" capacity the surface has. This means that for smooth nonhomogeneous illuminations, the low degrees of freedom will be good enough to simulate the surface, while high spatial frequency illuminations will require higher order polynomials. In the same way, the smaller the degree of the polynomial, the more rigid the surface will be.

Nevertheless, not only the nonhomogeneous illumination of the healthy skin is desired, but also the illumination of the melanoma. Because the melanoma is not part of the healthy skin, the nonhomogeneous illumination needs to be computed from the healthy skin. If the mask computed is too rigid, that means that it is not going to have the shape of the nonhomogeneous illumination wanted. But, if it is too flexible, it is going to fit well in the healthy skin, but not on the melanoma. A similar thing that happened with splines (but not for the same reason).

This is why several tests were done, and as a result it was seen that the best possible combination was a second grade polynomial.

3.3 Hair removal

The last problem considered during the research has been hair removal in melanoma images.

Although not as dense and thick as other terrestrial mammals, humans still have hair all over their skin, sometimes really difficult to see unless a close look to the skin

is taken. Nevertheless, because the melanoma detection application takes really close pictures of the skin, in some cases the existing hairs can degrade the performance of the application. Therefore it was explored the idea of removing the hair from the pictures of the skin.

The removal of hair from the pictures is not an easy task. The greatest challenge is that it is easy to remove the hair from some pictures but very difficult to remove it from others. Not all skin, hair and melanomas are the same, and in some cases the shape of a hair could look very similar to a feature of a melanoma.

In this particular project, the objective is going to focus on removing long, coarse black hairs, the most problematic to see through but the easiest type to remove. An example is seen in the Figure 3.28.

Figure 3.28: Lesion with thick, black hairs [23]

All the hair removal algorithms have 3 steps:

- Hair detection
- Hair mask correction
- Hair subtraction and hole filling

There are several methods in the literature that can remove hair [1],[27],[40],[2]. The most used one and the one that the community considers the standard is DullRazor[22]. This method has some advantages and disadvantages compared to other methods:

Advantages:

- Well known and used since 10+ years
- Easy concept and easy implementation
- Good results with black thick hair

Disadvantages:

- No free code available.
- It does not work with thin bright hair.

In this thesis a modification of DullRazor is going to be used.

3.3.1 Hair detection

The first step is to detect the hair in the image. This can be done by trying to look for objects with the shape of a hair, blur the image so that the hair is removed, try to filter the hair as well as several other methods.

Trying to find structures similar to hair is unpractical, because hair could be found in different shapes and sizes, so it is difficult to try this approach. Moreover, in some cases the structure of a certain feature on a melanoma and the structure of the hair may be similar and the first one wants to be preserved while deleting the second one.

When the image is blurred (for example using a low pass filter), the high frequency components of the image are removed and the low frequency ones remain. In the case that the hair is responsible for the high frequency components and the rest of the melanoma forms the low frequency components, this approach could be useful. Nevertheless, this condition is usually not true, and when something like that is done, some features of the hair remain in the result while some details of the melanoma are deleted.

This is why DullRazor uses another approach (the same one followed here), which consist on using image closing. Image closing is defined as:

$$A \bullet B = (A \oplus B) \ominus B$$

where $\oplus$ is the dilation and $\ominus$ is the erosion operation. In order to use the image

closing, two elements are needed:

- The image where the closing operation wants to be applied on.
- The structure used to apply the closing operation.

There are several different structures that could be used to perform the closing operation: disks, rectangles, squares, spheres, octagons or lines. The structure that resembles hairs most to is a line. Depending on the picture, hairs can grow in any direction, therefore, several lines have been used: one for the horizontal components of the hair, another one for vertical components of the hair and two for diagonal components.

If A is the melanoma image with hair, B is the horizontal line structure, C is the vertical line structure and D is the diagonal line structure, the operation that is performed is:

$$\text{closed image} = (((A \bullet B) \bullet C) \bullet D)$$

With this, the image has been modified in such a way that the hairs are no longer there, although the rest of the image has been modified too. It must taken into account that the morphological closing operation can only be applied to grayscale or binary images. This is why the first step is to convert the image into grayscale, as it can be seen in Figure 3.29. The result of applying it to the example image is shown in Figure 3.30.

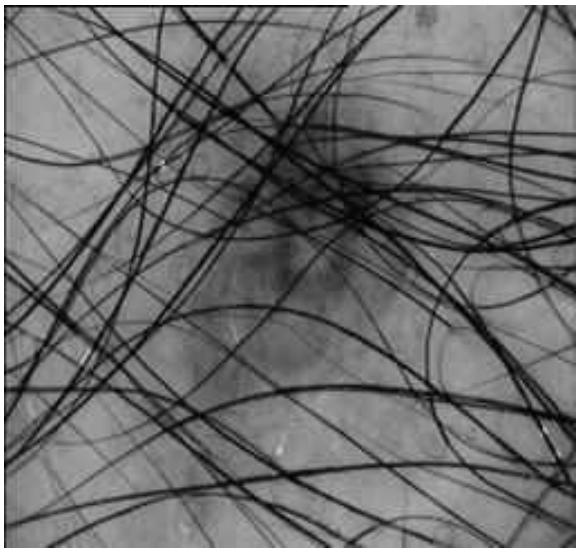


Figure 3.29: Grayscale melanoma image with thick, black hairs

The key here is that the hair part has been modified more than the rest of the image. Because of this, the result can be subtracted and thresholded. If the difference between the two images is significant, that means that there was an element (probably a hair) that has been removed. If the difference is small, that probably means that the image has been modified in a region were there was no hair. In order to see if a given pixel difference is small or significant, a thresholding method was used. The first step was to subtract both images (Figure 3.31) and then threshold the result (3.32).

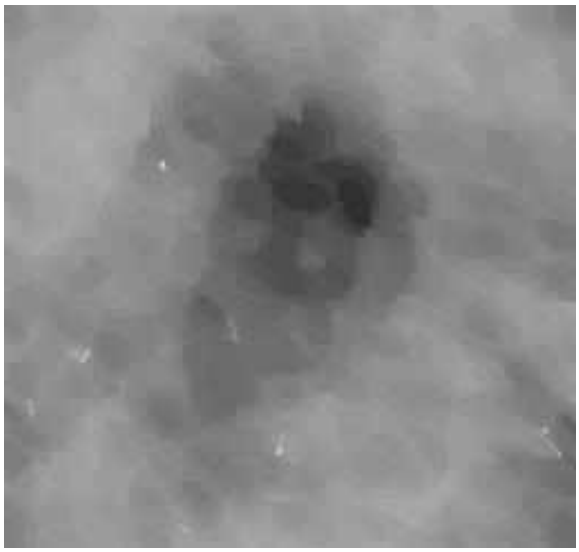


Figure 3.30: Morphological close over melanoma image with thick, black hairs

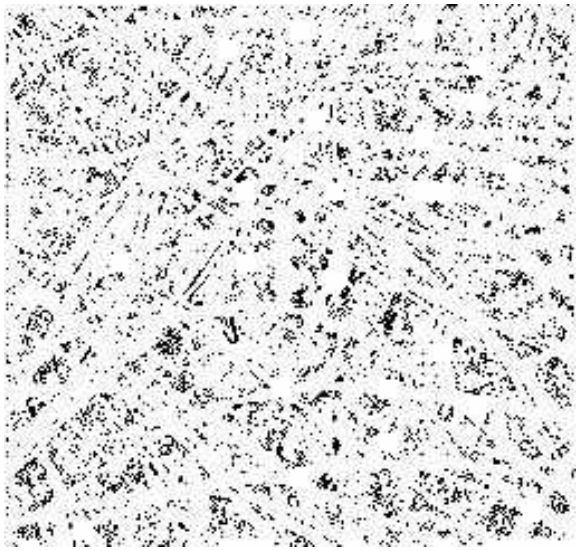


Figure 3.31: Difference of grayscale before and after morphological closing

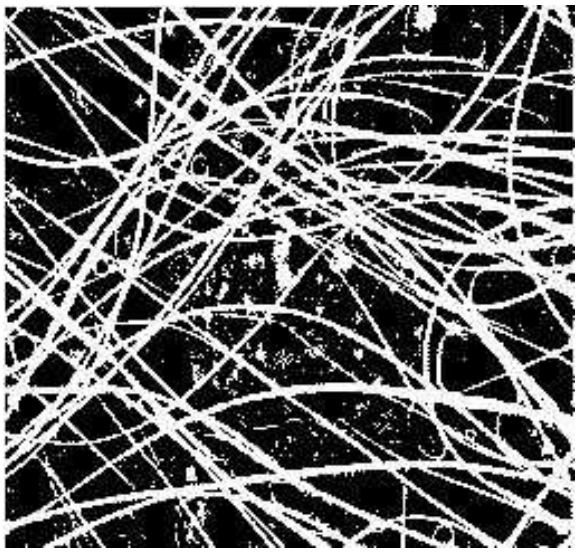


Figure 3.32: Thresholding of Figure 3.31

3.3.2 Mask correction

After this procedure, a binary mask was obtained, with 1s in the pixels that are considered hair and 0s in the pixels that are considered nonhair. Nevertheless, sometimes it can be seen that the mask contains small artifacts, small groups of pixel that are considered hair but are not, as seen in Figure 3.32. For example, this could be found when melanomas have hair-like features. In order to separate those two features, it can be taken into account that those features are usually small compared to hairs. Because of this, small groups of pixels were removed from the mask, as seen in Figure 3.33.

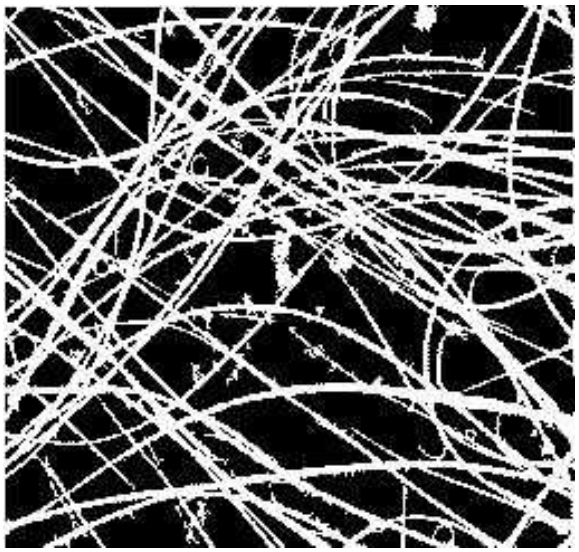


Figure 3.33: Cleaned Figure 3.32

Consequently, only large groups of pixels that the algorithm considers hair were taken into account. One final problem that was is that some times although the hair has been correctly detected, the exact boundary between the hair and the skin is not correct, creating some glitches that could be detected as melanoma features. An

example of this error can be found in Figure 3.34.

Figure 3.34: Removed hairs from 3.28 with glitches

In order to solve this, the hair mask is slightly dilated. Dilation could cause some skin/melanoma to be included as hair, but this is better than having some hairs detected as healthy skin. The resulting mask can be seen in Figure 3.35.

3.3.3 Gap filling

Once all the above procedures were completed, the final hair mask was obtained (Figure 3.35). The final step was to fill the gaps that the mask created with something that could be there(Figure 3.36). However, it is often challenging to determine what is under the hair. That information has been lost due to the hair. The best it can be done is to estimate the background.

Here is where DullRazor and the approach taken here differ. While DullRazor

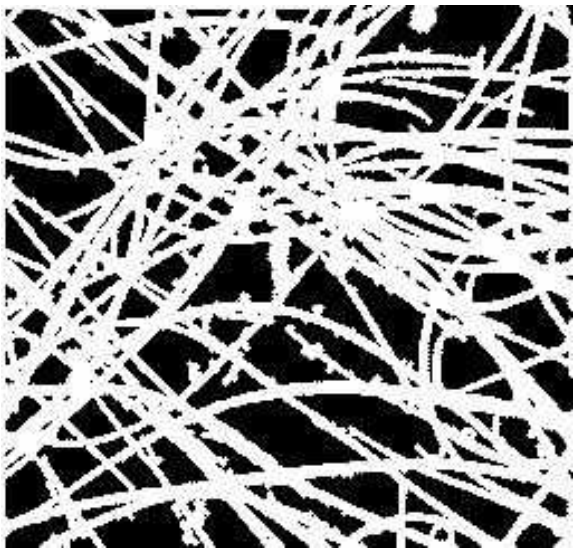


Figure 3.35: Dilated Figure 3.33

uses interpolation (Figure 3.37) to estimate the background, this approach uses RGB inpainting [18](Figure 3.38). This method was originally designed to run on a set of geodesic images.

In order to test the hair removal algorithm, some tests were performed. In those tests, lesions without hair were used. A random number of hairs uniformly distributed between 10 and 20, with uniformly random length from 1/4 of the length of the image to the full length of the image were included. The hair is simulated as a long, thin black line, with random rotation between 0 and 180 degrees.

Figure 3.36: Holes to be filled in Figure 3.28

Figure 3.37: Final Figure 3.28 using interpolation

Figure 3.38: Final Figure 3.28 using inpainting

Chapter 4

Results

In this chapter the results obtained when the algorithms were tested will be discussed. In some cases, the tests were done with real lesions. In others, simulated data were used. Also, in some simulations the results were given by a number, while in others, the final result is given by an image, which means that the result is subjective.

4.1 Superresolution

The first thing that was done was to test the different superresolution algorithms as it can be seen in Figure 4.1. The steps followed were:

- Create a synthetic melanoma.
- Simulate low resolution pictures of it.
- Run the superresolution algorithm on the low resolution pictures.

		Registration algorithms			
		KE	LU	MA	VA
SR algorithms	BP	3.518045	3.313514	4.358593	3.671430
	IN	2.903766	3.108306	3.941438	3.155352
	NC00	2.950938	3.275085	12.347428	3.222609
	NC01	2.915055	3.239756	12.352697	3.193301
	NC10	2.950938	3.258073	12.339806	3.264585
	NC11	2.915055	3.229613	12.343959	3.242935
	PG	11.580244	13.568094	12.331405	11.086319
	PO	3.821045	13.203370	4.954260	3.666100
	RS	3.272063	3.103303	3.690901	3.423080

Table 4.1: RMSE values for different registration and SR algorithms

- Compute the RMSE between the result and the high resolution original.

The RMSE results can be found in 4.1

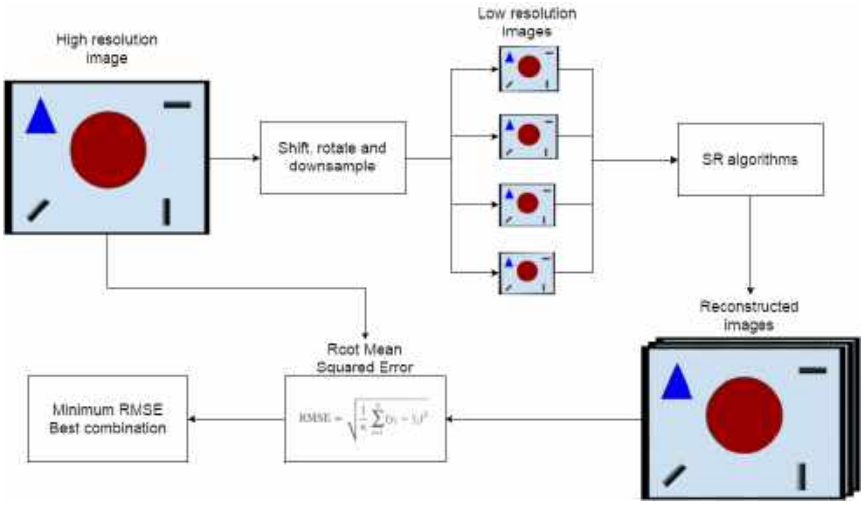


Figure 4.1: Superresolution testing scheme

4.1.1 Subjective results

One of the main problems with the research was the lack of data to work with. There are no libraries with more than a picture for every mole. This is why those algorithms have been run with acquired data. The data and the results can be found in appendix **??**.

4.2 Non-homogeneous illumination removal

In order to test the non-homogeneous illumination removal algorithm a synthetic skin lesion was used, which can be seen in Figure 4.3. Using this as a background, different nonhomogeneous illumination masks were added to this, which can be seen in Figure 4.4. Finally, the non-homogeneous illumination was removed using the algorithm and the results compared using RMSE, as it can be seen in Figure 4.2.

The original synthetic image can be seen in Figure 4.3.

Then, the different non-homogeneous illumination masks were added to the original synthetic melanoma. In Figure 4.4 circular centered, circular non-centered, horizontal, vertical and diagonal nonhomogeneous illumination masks can be seen.

The last step was to remove the non-homogeneous illumination. A linear regression was used as well as different polynomial levels. The results of each one of the degrees is shown below:

- For the 1st order polynomial: Figure B.1 and Table B.1.

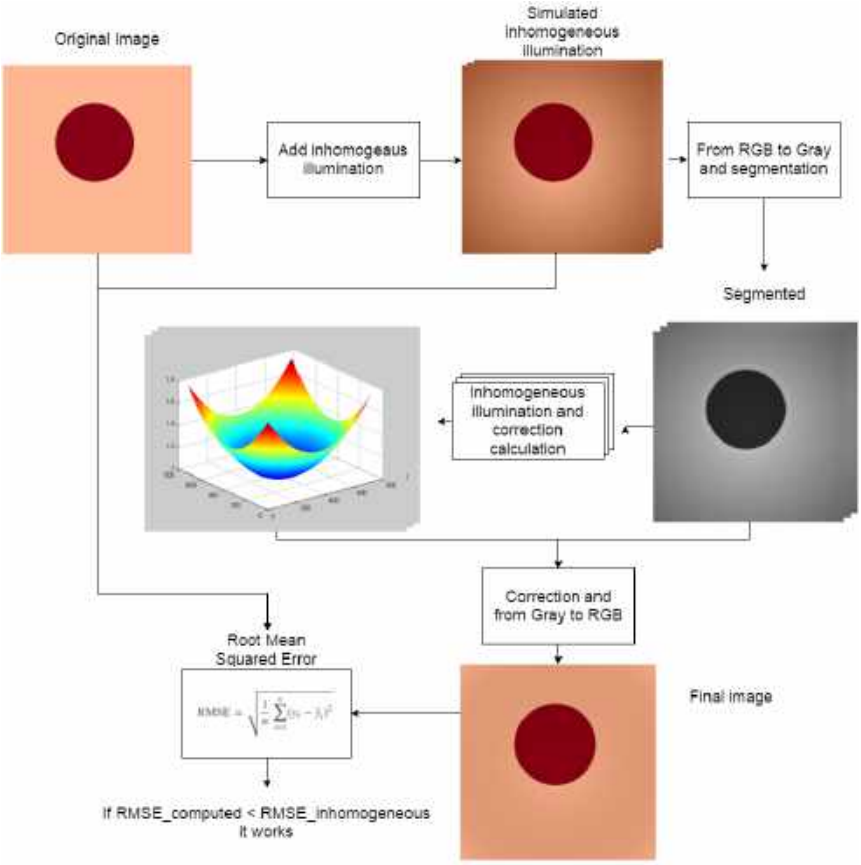


Figure 4.2: Nonhomogeneous illumination removal testing scheme

- For the 2nd order polynomial: Figure B.2 and Table B.2.
- For the 3rd order polynomial: Figure B.3 and Table B.3.
- For the 4th order polynomial: Figure B.4 and Table B.4.
- For the 5th order polynomial: Figure B.5 and Table B.5.

Results conclude that some of the polynomial degrees are too low to accurately represent the nonhomogeneous illumination and some of them are too high, creating strange artifacts.

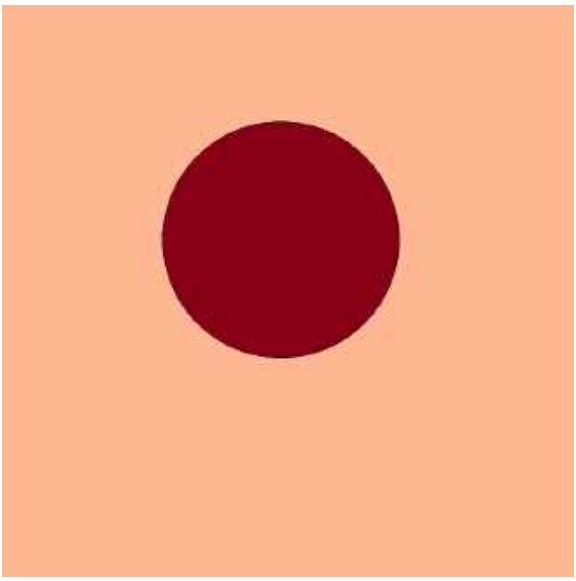


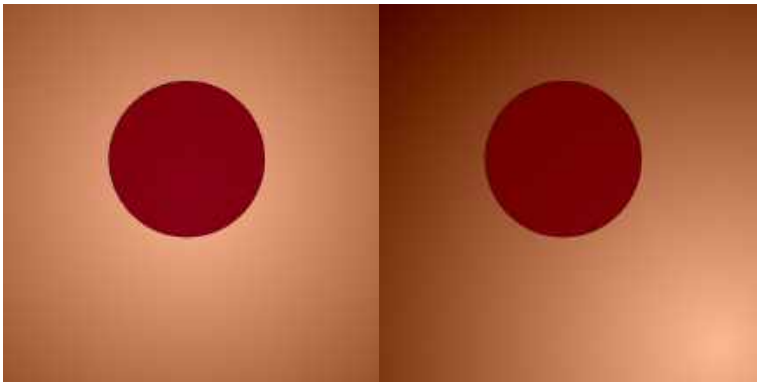
Figure 4.3: Synthetic melanoma

As it can be seen, the smallest RMSE belongs to the 2nd order polynomial linear regression, which means that this is the best choice.

4.3 Hair removal

In order to test the hair removal algorithm, some lesions (melanomas and healthy skin) were used. Those lesions contain no hair at all. The lesions used can be seen at Figure 4.5.

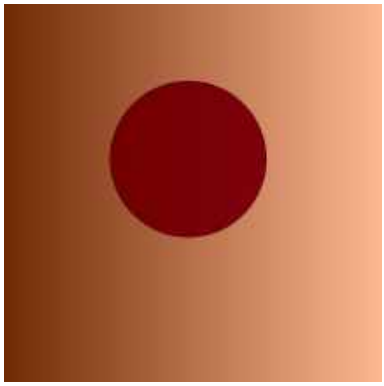
With the method explained in Chapter 3, synthetic hair was added to the images. Because the hair was added randomly, the experiment was repeated 10 times for each image, in order to see the performance of the algorithm under different circumstances. A summary of the method can be seen at Figure 4.6. The different outputs can be



(a) Synthetic melanoma with circular nonhomogeneous illumination mask

(b) Synthetic melanoma circular not centered nonhomogeneous illumination mask

(c) Synthetic melanoma diagonal nonhomogeneous illumination mask



(d) Synthetic melanoma horizontal nonhomogeneous illumination mask

(e) Synthetic melanoma vertical nonhomogeneous illumination mask

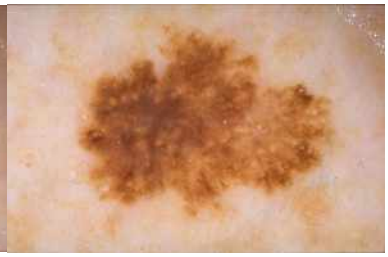
Figure 4.4: Synthetic melanoma and nonhomogeneous illumination masks

(a) Reference image 1

(b) Reference image 2

(c) Reference image 3

(d) Reference image 4



(e) Reference image 5

(f) Reference image 6

(g) Reference image 7

Figure 4.5: Reference images with no hair

seen in Figures 4.7,4.10,4.13,4.16,4.19,4.22 and 4.25.

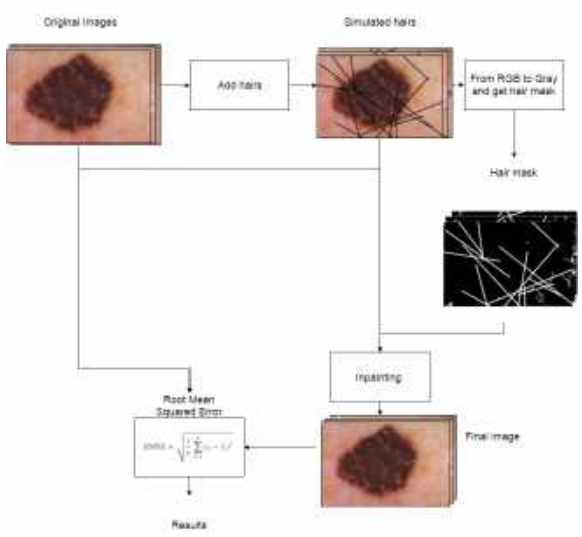


Figure 4.6: Testing procedure for the hair removal algorithm

After the synthetic hair was added, the hair mask of each image was calculated. The different outputs can be seen in Figures 4.8,4.11,4.14,4.17,4.20,4.23 and 4.26.

The last step was to remove the hair and fill the gaps. The different outputs can be seen in Figures 4.9,4.12,4.15,4.18,4.21,4.24 and 4.27.

Here the results of the hair removal algorithm are shown.

4.4 Synthetic hair added

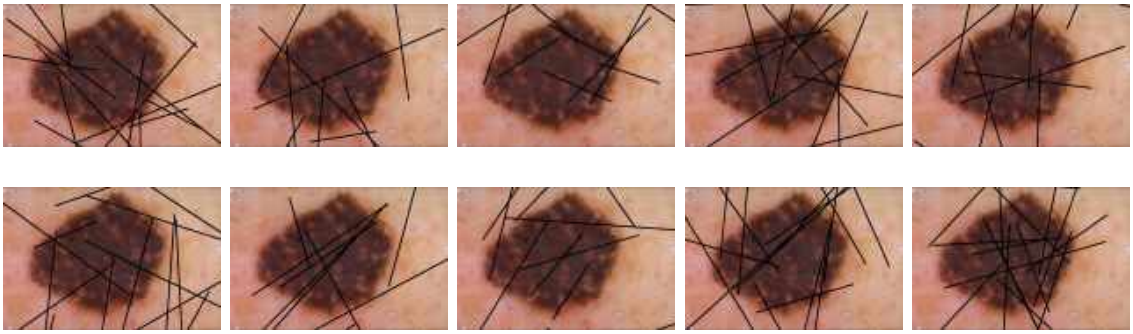


Figure 4.7: Reference image 1 with synthetic hair

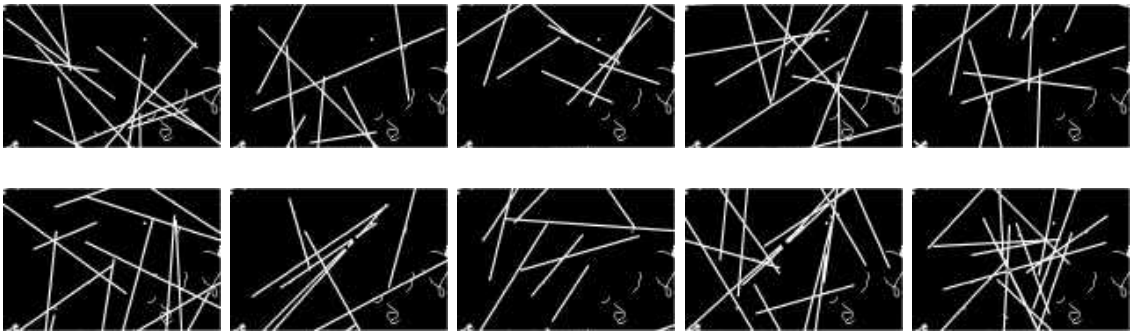


Figure 4.8: Reference image 1 hair masks

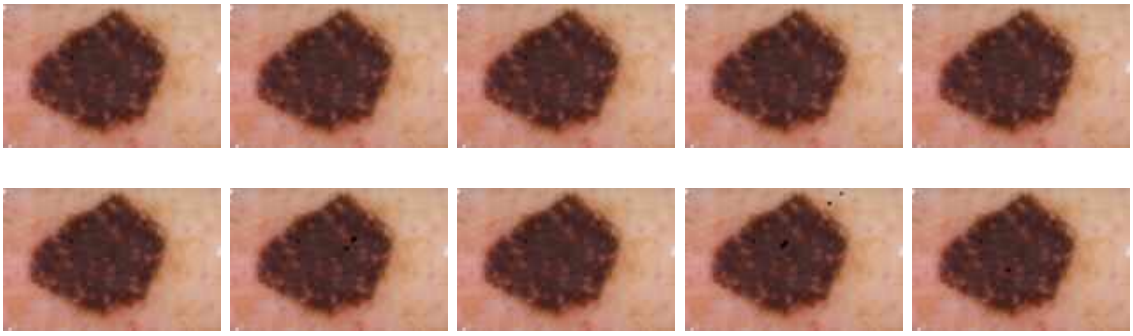


Figure 4.9: Reference image 1 without hairs

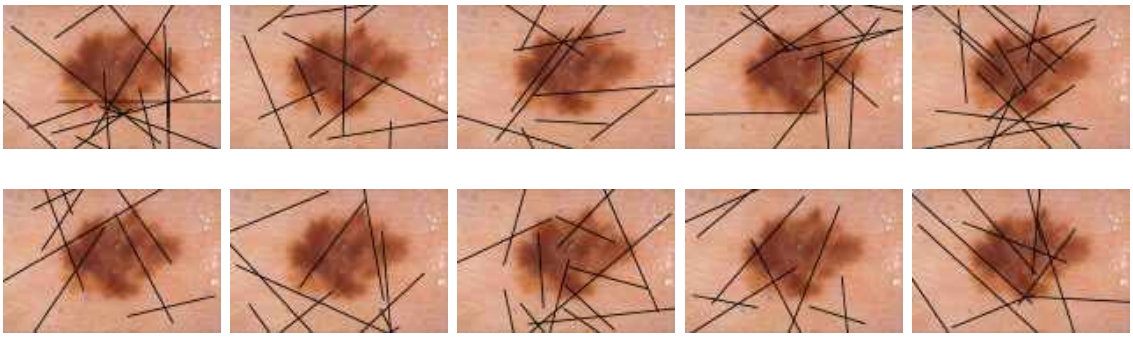


Figure 4.10: Reference image 2 with synthetic hair

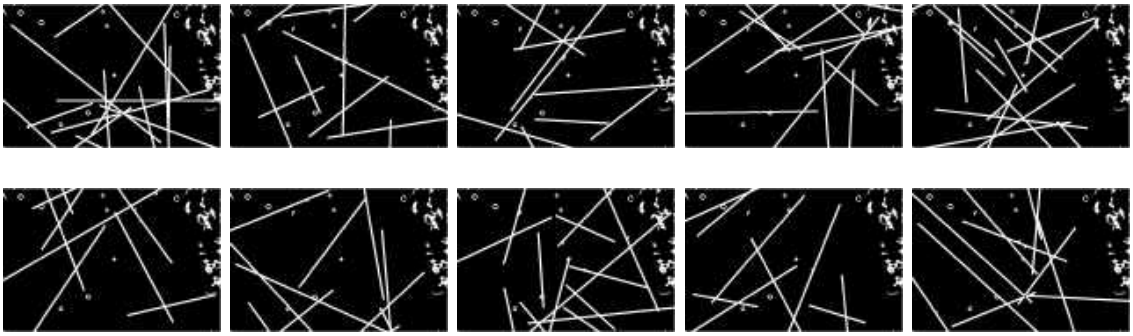


Figure 4.11: Reference image 2 hair masks

Figure 4.12: Reference image 2 without hairs

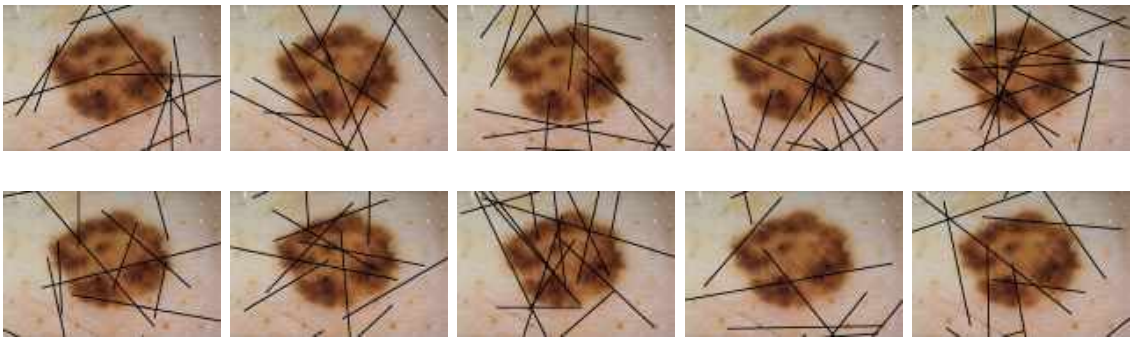


Figure 4.13: Reference image 3 with synthetic hair

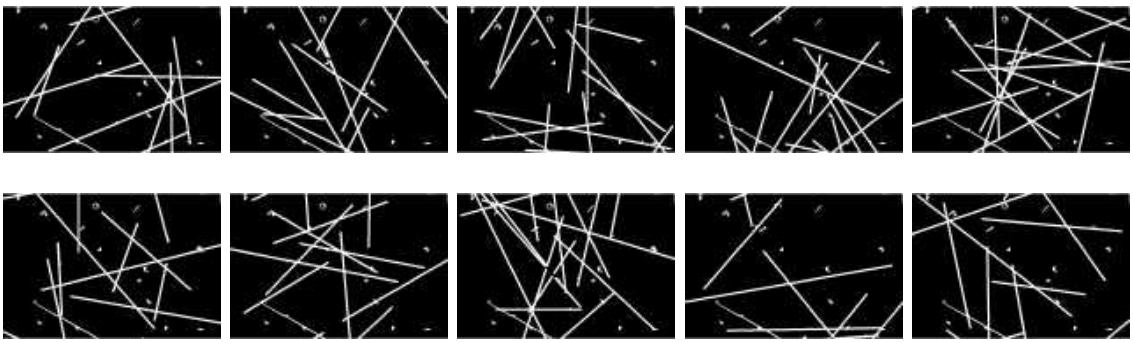


Figure 4.14: Reference image 3 hair masks

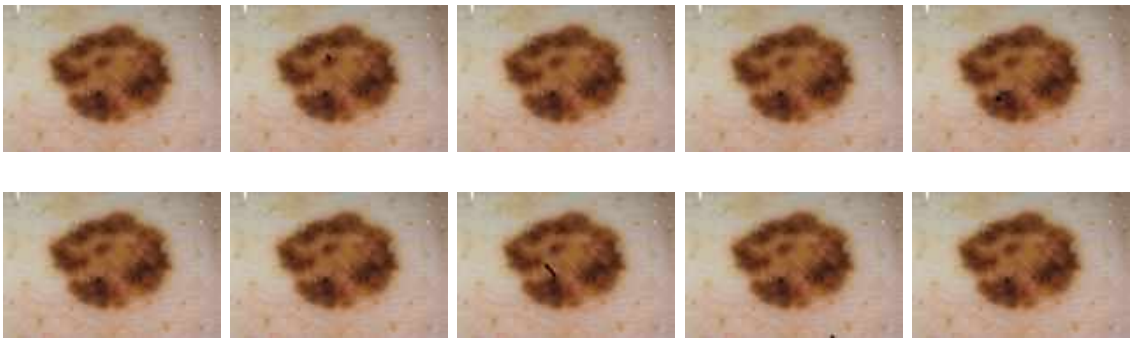


Figure 4.15: Reference image 3 without hairs

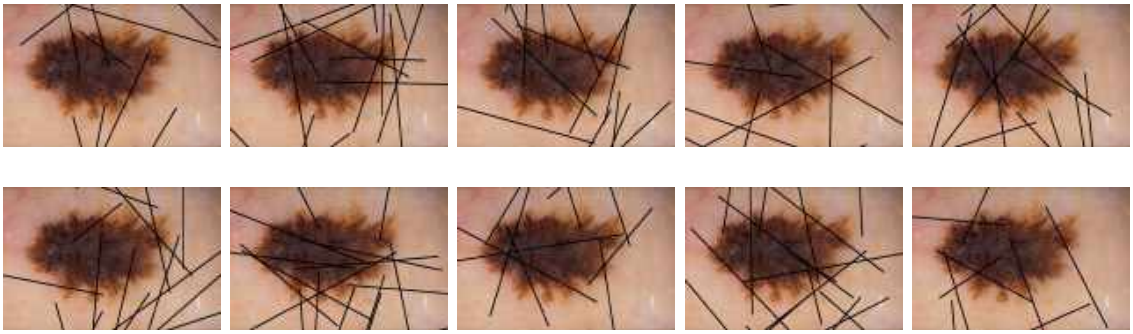


Figure 4.16: Reference image 4 with synthetic hair

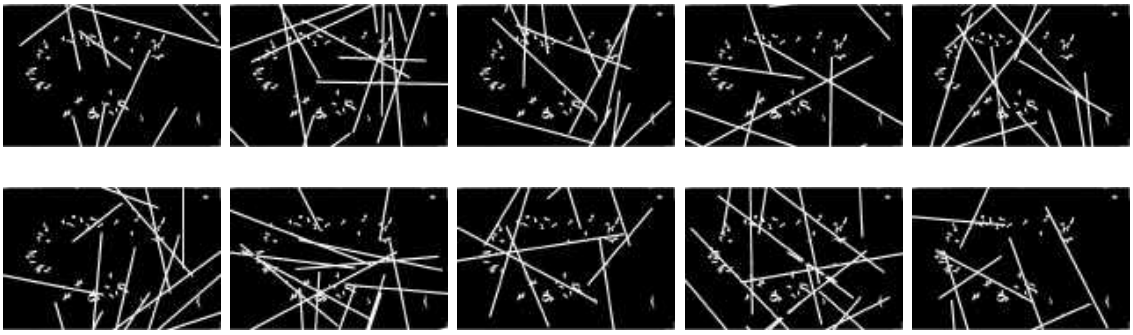


Figure 4.17: Reference image 4 hair masks

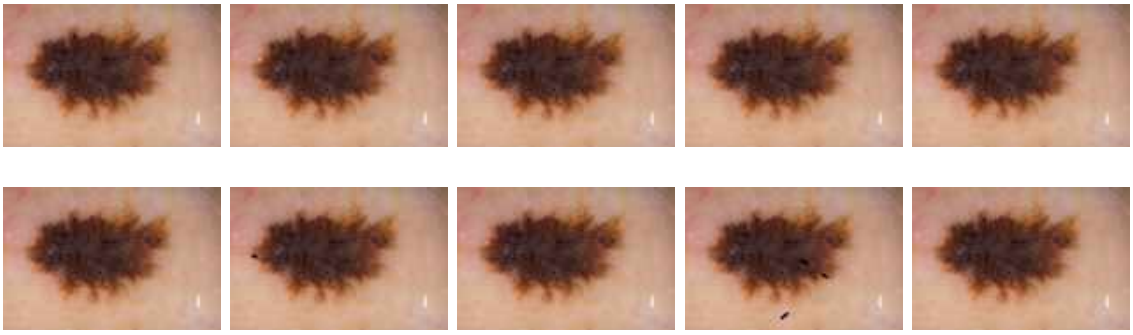


Figure 4.18: Reference image 4 without hairs

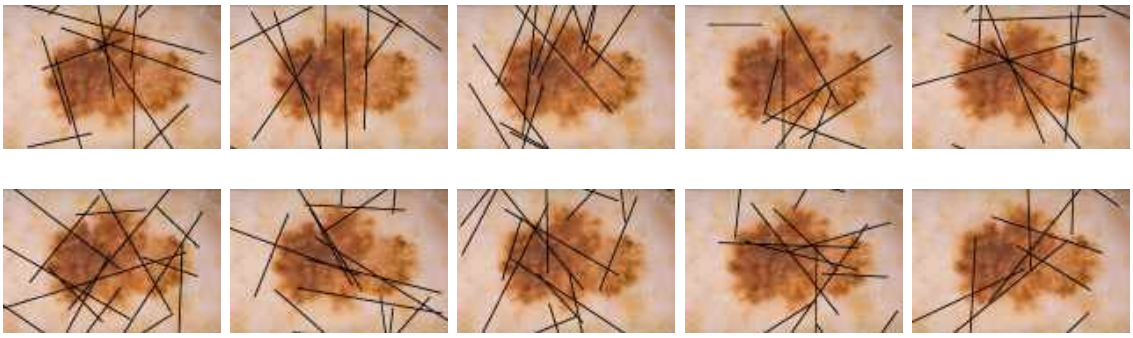


Figure 4.19: Reference image 5 with synthetic hair

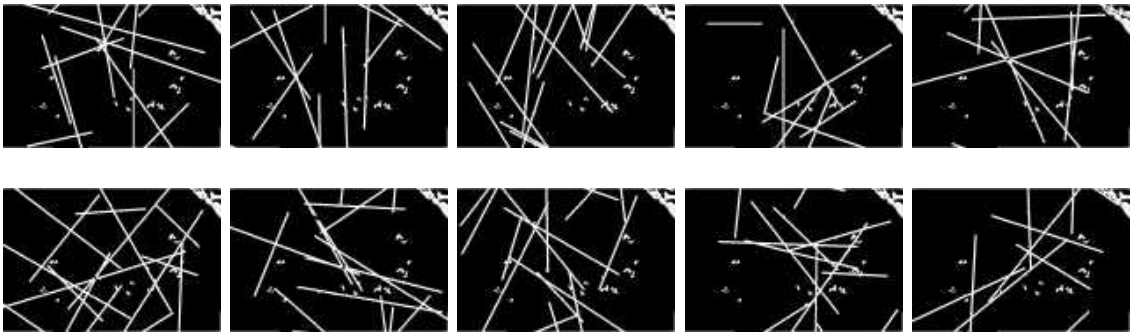


Figure 4.20: Reference image 5 hair masks

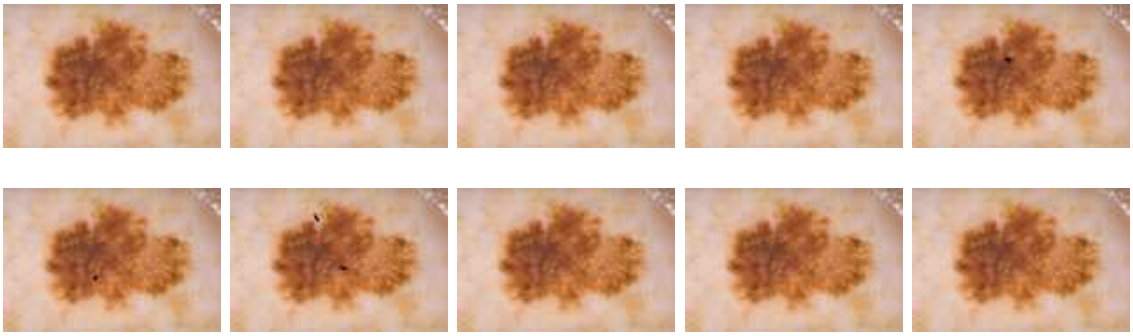


Figure 4.21: Reference image 5 without hairs

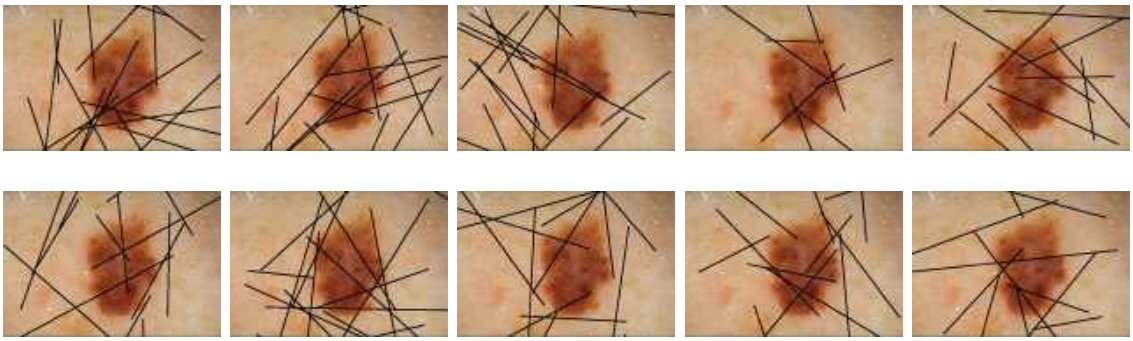


Figure 4.22: Reference image 6 with synthetic hair

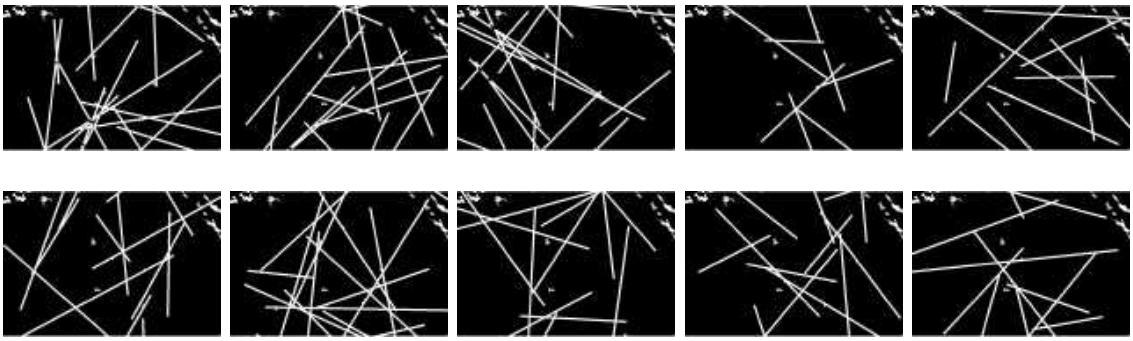


Figure 4.23: Reference image 6 hair masks

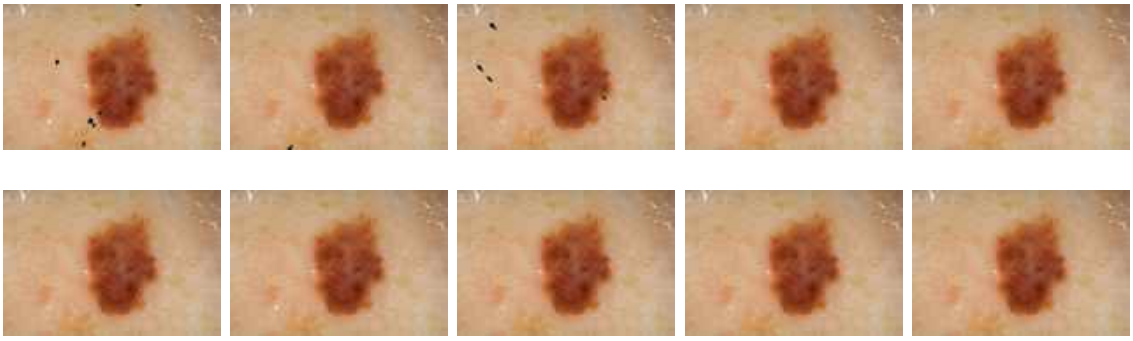


Figure 4.24: Reference image 6 without hairs

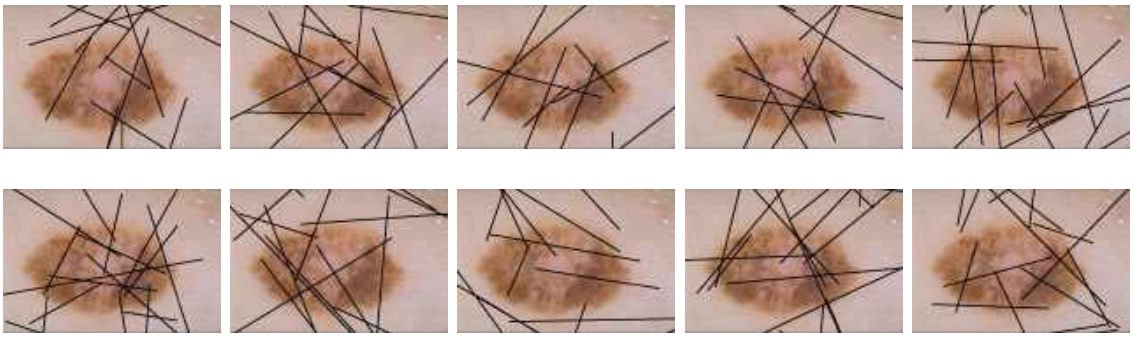


Figure 4.25: Reference image 7 with synthetic hair

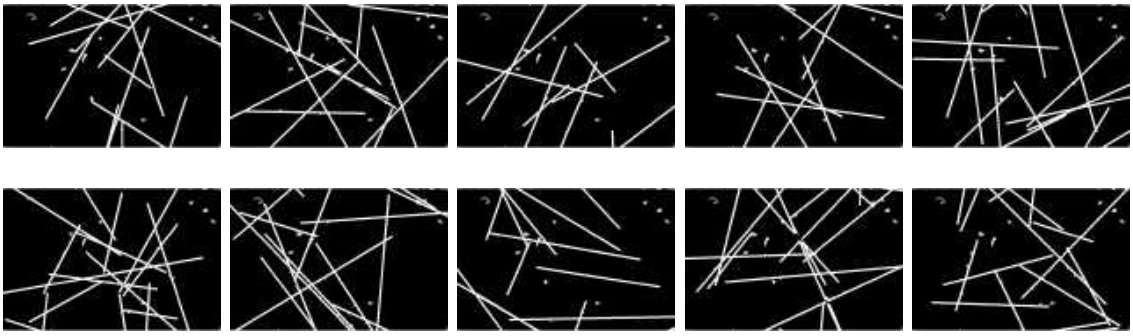


Figure 4.26: Reference image 7 hair masks

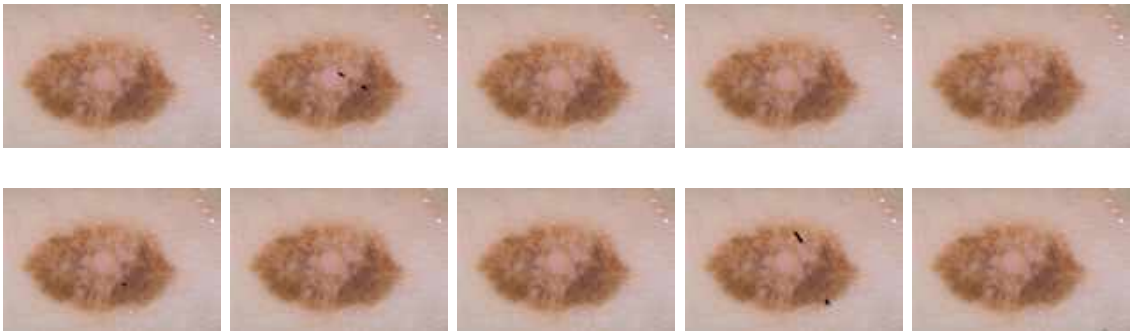


Figure 4.27: Reference image 7 without hairs

Here, several interesting things can be seen. The first one is that in the majority of the images, a great amount of hair is removed. The problem is that in some cases, more specifically where several hairs are together, the algorithms does not recognize them anymore. As emphasized previously, the algorithm looks for straight lines, not groups. When several hairs are placed together, it can be seen that the algorithm is not able to recognize the points were all of them converge, leaving some black dots on the image.

In order to compare the images, the RMSE was applied between the images with hair and the originals and the images after the algorithms is used and the originals. The results can be found in Tables 4.2,4.3,4.4,4.5,4.6,4.7,4.8.

33.7624	25.4198	23.4250	29.7195	24.0949
5.7472	5.7150	5.6974	5.8004	5.6963
35.6652	25.1871	23.9227	32.4035	27.3707
5.7771	6.0043	5.7119	6.8139	5.8434

Table 4.2: RMSE image 1 before (up inside cell) and after (down inside cell) the algorithm, for image 1 on the top right corner, image 2 the next one on the right,...

41.8191	33.8946	33.1547	37.0633	40.0585
6.3608	6.0330	5.8849	5.7906	6.0101
31.9551	34.9454	37.6895	32.0395	35.2255
5.8931	5.8795	5.8393	5.8718	6.3754

Table 4.3: RMSE image 2 before (up inside cell) and after (down inside cell) the algorithm, for image 1 on the top right corner, image 2 the next one on the right,...

As it can clearly be seen, the images looks more similar after the algorithm has been used over the image than before. Obviously, it is impossible to be entirely correct in the prediction, because the information of the skin behind the hair is lost, but there could be a quite accurate estimation.

36.2678	31.6917	36.7476	34.6212	37.9362
3.7792	4.1673	3.8304	3.7893	4.3556
31.6208	31.7653	37.0689	29.8855	33.0201
3.7408	3.8539	5.7588	4.4325	3.7858

Table 4.4: RMSE image 3 before (up inside cell) and after (down inside cell) the algorithm, for image 1 on the top right corner, image 2 the next one on the right,...

25.9677	36.8262	33.1023	29.4753	33.8074
4.5681	4.7513	4.6501	4.6348	4.6513
36.8868	37.1832	27.2733	38.0916	26.5185
4.6116	5.3635	4.7086	7.2265	4.6006

Table 4.5: RMSE image 4 before (up inside cell) and after (down inside cell) the algorithm, for image 1 on the top right corner, image 2 the next one on the right,...

37.4745	34.7565	37.9652	30.3714	35.2813
4.6120	4.5501	4.6027	4.5172	4.8562
44.3522	39.4985	38.8024	33.8238	31.0889
5.1550	6.8499	4.5920	4.5977	4.5555

Table 4.6: RMSE image 5 before (up inside cell) and after (down inside cell) the algorithm, for image 1 on the top right corner, image 2 the next one on the right,...

42.8964	40.5955	44.3718	23.6042	38.0108
9.1067	7.0122	9.4923	6.2391	6.3031
37.2091	44.6369	38.4690	34.5453	35.7413
6.2724	6.3776	6.2855	6.2735	6.2853

Table 4.7: RMSE image 6 before (up inside cell) and after (down inside cell) the algorithm, for image 1 on the top right corner, image 2 the next one on the right,...

37.0988	41.9952	33.2236	33.8180	43.2664
6.25642	7.19570	6.24984	6.25834	6.33033
41.0766	43.0413	38.7811	42.2015	39.3079
6.44751	6.32163	6.25489	8.56239	6.27014

Table 4.8: RMSE image 7 before (up inside cell) and after (down inside cell) the algorithm, for image 1 on the top right corner, image 2 the next one on the right,...

Chapter 5

Conclusion

This chapter includes a summary of the contributions of this research, as well as future work and problems found.

5.1 Summary of Contributions and Problems

5.1.1 Superresolution

The first task completed related to superresolution was to test several existing algorithms. It was found that the combination of the Keren estimation algorithm and Interpolation was the best combination.

However, other approaches were examined, such as an iterative approach.

The problem with the iterative approach was, most likely, that the performance

of the registration was not good enough and made the iterations divergent, causing a huge loss in the quality of the images.

5.1.2 Non-homogeneous illumination removal

A non-homogeneous illumination removal was designed, implemented and tested, showing good results for low frequency illumination.

Nevertheless, it must be taken into account that this algorithm was designed for low frequency nonhomogeneous illumination only. In some of the tests, when the illumination had nonlinear patterns and/or the illumination difference was very strong, the algorithm did not behave as well as expected.

5.1.3 Hair removal

A hair removal algorithm, similar to DullRazor but with a different gap-filling approach was designed, implemented and tested.

There are two main problems with this algorithm: The first one is that the algorithm works very well with thick, black hair but its performance with thin, light hairs has not been tested. Presumably, due to the thresholding step, the performance is not going to be as good as with thick, black hair. The second issue is that in the cases where several hairs were close together, the hair is no longer a line and is more similar to a disk. This is why the morphological closing does not work and creates some artifacts in some of the images.

5.2 Future Work

5.2.1 Subpixel registration and memory usage

Regarding the superresolution algorithms, one of the main constrains was that the algorithms require lots of memory. This is why, in the future, memory saving techniques should be used, to optimize the usage of the computer's RAM.

Also, it was found that some approaches are very sensitive to errors in the registration process. Due to the fact that doing subpixel level registration is not a trivial task, those algorithms show a poor performance. In the future, more work should be done in improving the performance of those algorithms

5.2.2 High frequency nonhomogeneous illumination removal and surface correction

Nonhomogeneous illumination removal algorithms work best for low frequency illuminations. It would be very interesting to see how it behaves for high frequency illumination components. One of the possible ways to solve this may be to just increase the polynomial's degree, but it has not been tested.

5.2.3 Thin bright hair removal and detection of groups of hairs

The hair removal algorithm has been prepared for black, thick hairs. The algorithm has not been tested for bright, thin hairs, but it will probably not work. Some modifications in the thresholding step may be required to make it work.

Finally, one of the biggest drawbacks of this algorithm is that it does not recognize areas were several hairs converge, creating some artifacts. The next step would be to remove those artifacts.

Chapter 6

Budget

6.1 Equipment

The equipment used in this project includes:

- A desktop MAC.
- A desktop PC.
- A laptop computer.

6.2 Expenses

The expenses included:

- $\frac{engineer\_salary}{hours} \times number\_hours$

- $\frac{MAC\_prize}{useful\_life} \times number\_hours$
- $\frac{PC\_prize}{useful\_life} \times number\_hours$
- $\frac{laptop\_prize}{useful\_life} \times number\_hours$
- Matlab license

The prices are:

- The average starting salary for a Master's in Telecommunication Engineering is $59,240 per year. This project has lasted 8 months at 3 hours per day.
- The MAC used in this project was an iMAC 12,1. It costs $1,045. The life expectancy of the iMAC is 5 years. It was used 4 hours per day during 5 months.
- The PC used in this project was an AMD A10-5800K. It costs $500. The life expectancy of the PC is 5 years. It was used 4 hours per day during 4 1/2 months.
- The laptop used on this project was an Aspire E1-571G. Its price was $388. The life expectancy of the laptop is 5 years. It was used 4 hours per day during 9 1/2 months.
- The new Matlab license is $5000 as per year per user.

So the expenses of this project have been:

- $Salary : \frac{59{,}240}{2{,}086} \times 521 = 14,795.8\$$
- $MAC\_Amortization : \frac{1{,}045}{14{,}600} \times 326 = 23.33\$$
- $PC\_Amortization : \frac{500}{14{,}600} \times 293 = 10.03\$$
- $Laptop\_Amortization : \frac{388}{14{,}600} \times 521 = 13.84$

This project has involved a price of $19,843.

Bibliography

[1] Q. Abbas, M. Celebi, and I. F. García. Hair removal methods: A comparative study for dermoscopy images. *Biomedical Signal Processing and Control*, 6(4):395–404, 2011.

[2] Q. Abbas, I. F. Garcia, M. E. Celebi, and W. Ahmad. A feature-preserving hair removal algorithm for dermoscopy images. *Skin Research and Technology Skin Res Technol*, 19(1), 2011.

[3] J. X. Ahmedin Jemal, Rebecca Siegel and E. Ward. Cancer statistics, 2010. *CA CANCER*, 60:277–300, 2010.

[4] A.-R. A. Ali and T. M. Deserno. A systematic review of automated melanoma detection in dermatoscopic images and its ground truth data. *Medical Imaging 2012: Image Perception, Observer Performance, and Technology Assessment*, 2012.

[5] AUDIOVISUAL COMMUNICATIONS LCAV. Super resolution, 2016. [Online; accessed April June, 2016].

[6] K. J. Barnard. High-resolution image reconstruction from a sequence of rotated

and translated frames and its application to an infrared imaging system. *Optical Engineering Opt. Eng*, 37(1):247, Jan 1998.

[7] N. Barzigar, A. Roozgard, P. Verma, and S. Cheng. A video super-resolution framework using scobep. *IEEE Trans. Circuits Syst. Video Technol. IEEE Transactions on Circuits and Systems for Video Technology*, 26(2):264–277, 2016.

[8] M. Binder. Epiluminescence microscopy. a useful tool for the diagnosis of pigmented skin lesions for formally trained dermatologists. *Archives of Dermatology*, 131(3):286–291, Jan 1995.

[9] W. Bishara, T.-W. Su, A. F. Coskun, and A. Ozcan. Lensfree on-chip microscopy over a wide field-of-view using pixel super-resolution. *Opt. Express Optics Express*, 18(11):11181, Dec 2010.

[10] L. Brochez, E. Verhaeghe, L. Bleyen, and J.-M. Naeyaert. Diagnostic ability of general practitioners and dermatologists in discriminating pigmented skin lesions. *Journal of the American Academy of Dermatology*, 44(6):979–986, 2001.

[11] B. Buttkus. Homomorphic filtering - theory and practice*. *Geophysical Prospecting Geophys Prospect*, 23(4):712–748, 1975.

[12] B. Buttkus. Homomorphic filtering - theory and practice*. *Geophysical Prospecting Geophys Prospect*, 23(4):712–748, 1975.

[13] P. G. Cavalcanti, J. Scharcanski, and C. B. O. Lopes. Shading attenuation

in human skin color images. *Advances in Visual Computing Lecture Notes in Computer Science*, page 190–198, 2010.

[14] G. R. Day and R. H. Barbour. Automated melanoma diagnosis: where are we at? *Skin Research and Technology Skin Res Technol*, 6(1):1–5, 2000.

[15] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence IEEE Trans. Pattern Anal. Mach. Intell.*, 38(2):295–307, Jan 2016.

[16] G. Z. W. T. S. et al. Number 54. 2015.

[17] L. From, M. Mckenzie, and N. Iscoe. Photographic follow-up of dysplastic nevi; incidence of melanoma. *Melanoma Research*, 3(1):48, 1993.

[18] D. Garcia. Robust smoothing of gridded data in one and higher dimensions with missing values. *Computational Statistics and Data Analysis*, 54(4):1167–1178, 2010.

[19] R. Hu, C. M. Queen, and G. Zouridakis. Detection of buruli ulcer disease: Preliminary results with dermoscopic images on smart handheld devices. *2013 IEEE Point-of-Care Healthcare Technologies (PHT)*, 2013.

[20] M. Irani and S. Peleg. Improving resolution by image registration. *CVGIP: Graphical Models and Image Processing*, 53(3):231–239, 1991.

[21] D. Keren, S. Peleg, and R. Brada. Image sequence enhancement using sub-pixel displacements. *Proceedings CVPR '88: The Computer Society Conference on Computer Vision and Pattern Recognition*.

[22] T. Lee, V. Ng, R. Gallagher, A. Coldman, and D. Mclean. Dullrazor®: A software approach to hair removal from images. *Computers in Biology and Medicine*, 27(6):533–543, 1997.

[23] Lee, T. and Ng, V. and Gallagher, R. and Coldman, A. and McLean. Dullrazor, 2016. [Online; accessed April June, 2016].

[24] D. C. Lozano, P. M. Kulkarni, G. Zouridakis, and M. D. Twa. A statistical model of retinal sd-oct data: Preliminary simulation and evaluation of denoising performance. *Investigative Ophthalmology and Visual Science*, 52(4):1318–1318, 2011.

[25] L. Lucchese and G. Cortelazzo. A noise-robust frequency domain technique for estimating planar roto-translations. *IEEE Transactions on Signal Processing IEEE Trans. Signal Process.*, 48(6):1769–1786, 2000.

[26] G. Lukinavičius, K. Umezawa, N. Olivier, A. Honigmann, G. Yang, T. Plass, V. Mueller, L. Reymond, I. R. C. Jr, Z.-G. Luo, and et al. A near-infrared fluorophore for live-cell super-resolution microscopy of cellular proteins. *Nature Chemistry Nature Chem*, 5(2):132–139, Jun 2013.

[27] I. Maglogiannis and K. Delibasis. Hair removal on dermoscopy images. *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2015.

[28] B. Marcel, M. Briot, and R. Murrieta. Calcul de translation et rotation par la transformation de fourier. *TS. Traitement du signal*, 14(2):135–149, 1997.

[29] Mathworks. Homomorphic filtergin post 1, 2016. [Online; accessed April June, 2016].

[30] Mathworks. Registering aerial photos using point mapping, 2016. [Online; accessed April June, 2016].

[31] S. C. Park, M. K. Park, and M. G. Kang. Super-resolution image reconstruction: a technical overview. *IEEE Signal Process. Mag. IEEE Signal Processing Magazine*, 20(3):21–36, 2003.

[32] T. Q. Pham, L. J. V. Vliet, and K. Schutte. Robust fusion of irregularly sampled data using adaptive normalized convolution. *EURASIP Journal on Advances in Signal Processing EURASIP J. Adv. Signal Process.*, 2006:1–13, 2006.

[33] M. R. Project. Melanoma research project. `https://http://melresproj.com/`, 2016. [Online; accessed June-2016].

[34] E. Proksch, J. M. Brandner, and J.-M. Jensen. The skin: an indispensable barrier. *Experimental Dermatology*, 17(12):1063–1072, 2008.

[35] D. N. Rebecca Siegel and A. Jemal. Cancer statistics, 2013. *CA CANCER*, 63:11–30, 2010.

[36] T. Schindewolf, W. Stolz, R. Albert, W. Abmayr, and H. Harmst. Efficiency of the abcd-rule for the differentiation between malignant melanoma and melanocytic nevi using digital image analysis. *Melanoma Research*, 3:41, 1993.

[37] Y. Shi, M. Xiao, and J. Yang. Pixel-based skin color detection considering

overlap region. *2007 International Symposium on Intelligent Signal Processing and Communication Systems*, 2007.

[38] N. Situ, T. Wadhawan, R. Hu, K. Lancaster, X. Yuan, and G. Zouridakis. Evaluating sampling strategies of dermoscopic interest points. *2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2011.

[39] H. Stark. Theory of convex projection and its application to image restoration. *1988., IEEE International Symposium on Circuits and Systems.*

[40] M. T. B. Toossi, H. R. Pourreza, H. Zare, M.-H. Sigari, P. Layegh, and A. Azimi. An effective hair removal algorithm for dermoscopy images. *Skin Research and Technology Skin Res Technol*, 19(3):230–235, Jul 2013.

[41] P. Vandewalle, S. Süsstrunk, and M. Vetterli. A frequency domain approach to registration of aliased images with application to super-resolution. *EURASIP Journal on Advances in Signal Processing EURASIP J. Adv. Signal Process.*, 2006:1–15, 2006.

[42] T. Wadhawan, N. Situ, K. Lancaster, X. Yuan, and G. Zouridakis. Skinscan©: A portable library for melanoma detection on handheld devices. *2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2011.

[43] T. Wadhawan, N. Situ, H. Rui, K. Lancaster, X. Yuan, and G. Zouridakis. Implementation of the 7-point checklist for melanoma detection on smart handheld devices. *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2011.

[44] Wikipedia, the free encyclopedia. Bicubic interpolation, 2016. [Online; accessed April June, 2016].

[45] Wikipedia, the free encyclopedia. Bilinear interpolation, 2016. [Online; accessed April June, 2016].

[46] Wikipedia, the free encyclopedia. Nearest neighbor interpolation, 2016. [Online; accessed April June, 2016].

[47] A. Zomet and S. Peleg. Efficient super-resolution and applications to mosaics. *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000.*

[48] G. Zouridakis, T. Wadhawan, N. Situ, R. Hu, X. Yuan, K. Lancaster, and C. M. Queen. Melanoma and other skin lesion detection using smart handheld devices. *Methods in Molecular Biology Mobile Health Technologies*, page 459–496, May 2014.

[49] G. Zouridakis, T. Wadhawan, N. Situ, R. Hu, X. Yuan, K. Lancaster, and C. M. Queen. Melanoma and other skin lesion detection using smart handheld devices. *Methods in Molecular Biology Mobile Health Technologies*, page 459–496, May 2014.

Appendix A

Algorithms

In this chapter, the pseudocode used for the algorithms is shown. It must be taken into account that the code has been written in Matlab 2015B, so it may happen that some of the functions are not explained, because the function is exclusive to a Matlab 2015B toolbox. The toolbox that was used the most is the Image Processing Toolbox.

Although in the previous chapter all the methods were explained, here not every method is going to be shown, but only the final ones, the ones that have been proved to work best.

A.1 Superresolution

Data: uint8 RGB melanoma image (x4 or more), factor

Result: uint8 RGB superresolution melanoma image

initialization;

deltaest,phiest←keren(melanomasRGB);

melanomaYUV←RGB2YUV(melanomasRGB.1);

srU←bicubicinterpolation(melanomaYUV.U,factor);

srV←bicubicinterpolation(melanomaYUV.V,factor);

srY←srinterpolation(melanomasRGB,factor);

srRGB←YUV2RGB(srYUV);

return srRGB;

Algorithm 1: Superresolution algorithm

Data: uint8 RGB melanoma image (x4 or more), factor

Result: deltaest, phiest

initialization;

reference←image.1;

for *every image starting from the second* **do**

    pyramid.3←image;

    pyramid.2←resize(pyramid.3,0.5);

    pyramid.1←resize(pyramid.2,0.5);

    for *every pyramid level* **do**

        d/dy←computeddy(level);

        d/dx←computeddx(level);

        df1/dx←computedf1dx(level);

        df1/dy←computedf1dy(level);

        df1/dy*xdf1/dx*y←computedf1dyxdf1dxy(level);

        while *parameter difference not small* **do**

            a←computea(parameters);

            b←computeb(parameters);

            θ ←computeθ();

        end

        if *not last level* **then**

            sendabθtonextpyrlevel();

        end

    end

end

return a,b,θ;

Algorithm 2: Keren's registration estimation algorithm

Data: uint8 RGB melanoma image (x4 or more), factor, deltaest, phiest

Result: uint8 RGB superresolution melanoma image

initialization;

for *every image* **do**

melanomaY.image←RGB2Y(melanomasRGB.image);

for *every pixel* **do**

col.pixel←(rotate(melanomaY.image,phiest.image).pixel.col-
deltaest.image.col);

row.pixel←(rotate(melanomaY.image,phiest.image).pixel.row-
deltaest.image.col);

intensity.pixel=melanomaY.image.pixel.intensity;

end

end

finalsr = griddata(col,row,intensity,desiredpoints(melanomaY,factor));

return finalsr;

Algorithm 3: Superresolution image generation algorithm

A.2 Non-homogeneous illumination removal

Data: uint8 RGB melanoma image

Result: uint8 RGB Non-homogeneous illumination corrected melanoma image

initialization;

YUV←RGB2YUV(melanomaRGB);

for *every YUV layer* **do**

  melmask.layer←segmentation(YUV.layer);

end

melbinarymask←melmask.Y **or** melmask.U **or** melamask.V;

illuminationmask←nonhomoilluminationmask(Y,melbinarymask);

desiredillumination←max(illuminationmask);

for *every pixel in illuminationmask* **do**

  corretionmask.pixel← desiredillumination/illuminationmask.pixel;

end

for *every pixel in Y* **do**

  Y.pixel← Y.pixel*corretionmask.pixel;

end

finalRGB←YUV2RGB(YUV);

return finalRGB;

Algorithm 4: Illumination correction algorithm

Data: double RGB melanoma image

Result: double YUV melanoma image

Y←0.299R+0.587G+0.114B;

U←-0.147R-0.298G+0.436B;

V←0.615R-0.515G-0.1B;

return YUV;

Algorithm 5: Illumination color space change from RGB to YUV

Data: double YUV melanoma image

Result: double RGB melanoma image

R←Y+0U+1.14V;

G←Y-0.396U-0.581V;

B←Y+2.029U+0V;

return RGB;

Algorithm 6: Illumination color space change from YUV to RGB

Data: double Y intensity melanoma image, segmented melanoma

Result: non-homogeneous illumination mask

```
/* First each one of the coordinates of the grayscale image is
   taken into a separate array                                  */
```

for *every pixel that belong to healthy skin* **do**

    x.pixel←Y.pixel.x;

    y.pixel←Y.pixel.y;

    value.pixel←Y.pixel.value;

end

```
/* Then the fit function is called, which does a linear
   regression using the specified model, in this case a 2nd order
   2D polynomial                                                */
```

finalmask=fit(x,y,z,'poly22');

return finalmask;

Algorithm 7: Illumination color space change from YUV to RGB

A.3 Hair removal

Data: uint8 RGB melanoma image

Result: uint8 RGB melanoma image without hairs

initialization;

binaryhairmask←hair mask(melanomaRGB);

melanomawithnohairs←hole filling(melanomaRGB,binaryhairmask);

return melanomawithnohairs;

Algorithm 8: hair removal algorithm

Data: uint8 RGB melanoma image

Result: binary hair mask

melanomaGRAY← grayscale(melanomaRGB);

create structuring elements for closing operation;

closed ← melanomaGRAY;

for *every structuring element* **do**

  closed ← imclose(closed,closing element);

end

abssolutedifference ← abs(closed - melanomaGRAY);

level←20;

binarymask ← zeros;

for *every pixel in absolutedifference* **do**

  if *absolutedifference.pixel>level* **then**

    binarymask.pixel←1;

  end

end

```
/* To clean the mask from small groups of pixels                */
```

binarymaskcleaned←bwareaopen(binarymask,50);

```
/* 50 is the minimum size to be considered hair                  */
/* To make the mask thicker                                      */
```

finalmask←imdilate(binarymaskcleaned,1 pixel estructuring element);

```
/* 1 pixel estructuring element is obtained by calling
   strel('disk',1), which creates a disk of size 1              */
```

return finalmask;

Algorithm 9: hair mask

Data: uint8 RGB melanoma image, binary hair mask

Result: uint8 RGB melanoma image without hairs

for *every layer of the RGB image* **do**

```
/* The first function filles the holes using interpolation.
   It is used as a reference for inpaintn                    */
```

filled0←regionfill(melanomaRGB.layer, mask);

```
/* inpaintn algorithm fills the pixels where the value of the
   pixes is NaN                                              */
```

melanomaRGBNaN←melanomaRGB.layer;

for *every pixel detected as hair* **do**

melanomaRGBNaN←NaN;

end

melanomanohairs.layer←inpaintn(melanomaRGBNaN,100,filled0);

end

return melanomanohairs;

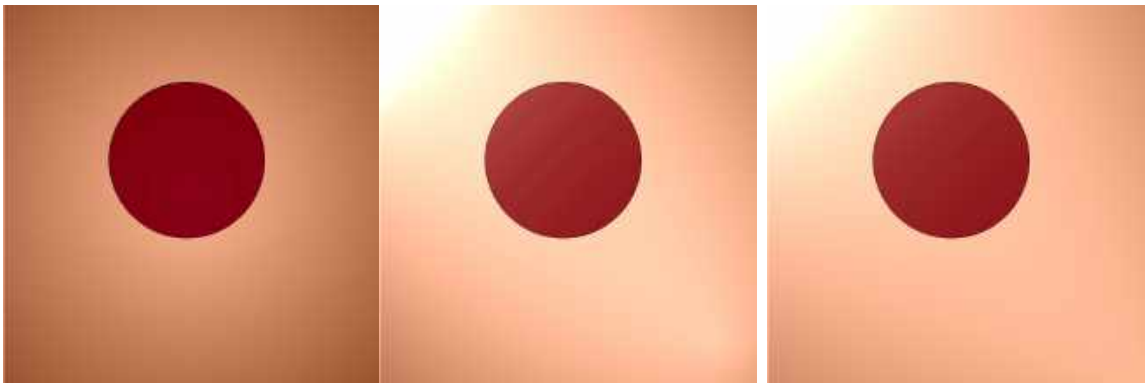
Algorithm 10: hole filling algorithm

Appendix B

Illumination correction: regression

Here the results of the illumination correction using linear regression are shown.

B.1 Images



(a) First order polynomial circular mask

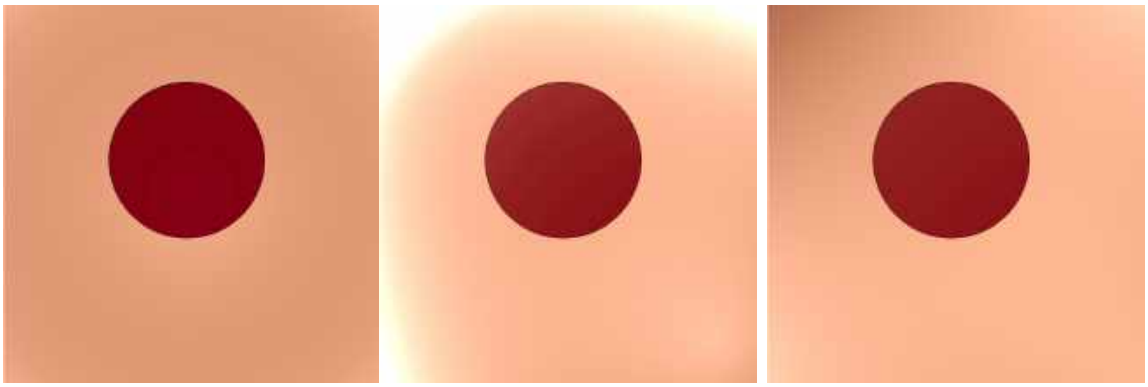
(b) First order polynomial circular not centered mask

(c) First order polynomial diagonal mask

(d) First order polynomial horizontal mask

(e) First order polynomial vertical mask

Figure B.1: Synthetic melanomas with removed non-homogeneous illumination, first order polynomial



(a) Second order polynomial circular mask

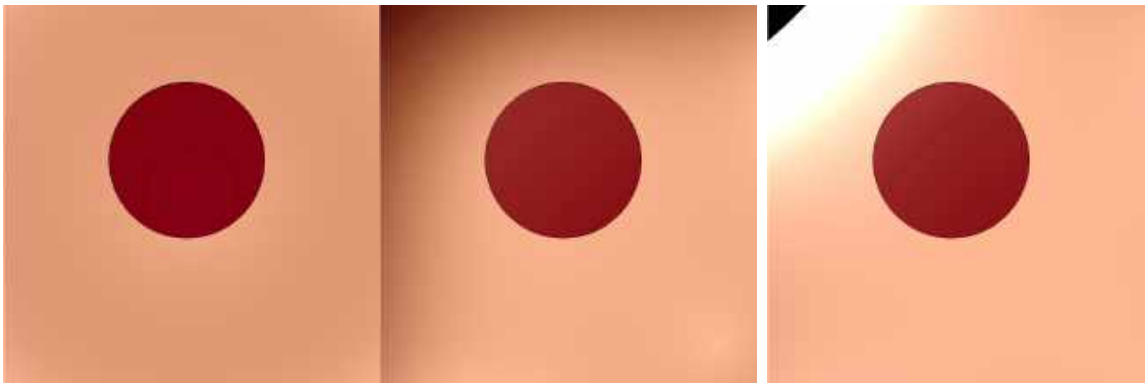
(b) Second order polynomial circular not centered mask

(c) Second order polynomial diagonal mask

(d) Second order polynomial horizontal mask

(e) Second order polynomial vertical mask

Figure B.2: Synthetic melanomas with removed non-homogeneous illumination, second order polynomial



(a) Third order polynomial circular mask

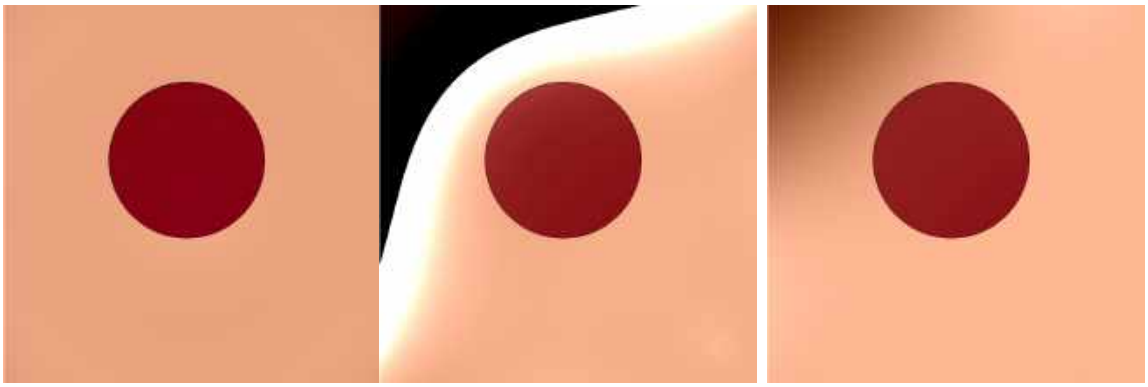
(b) Third order polynomial circular not centered mask

(c) Third order polynomial diagonal mask

(d) Third order polynomial horizontal mask

(e) Third order polynomial vertical mask

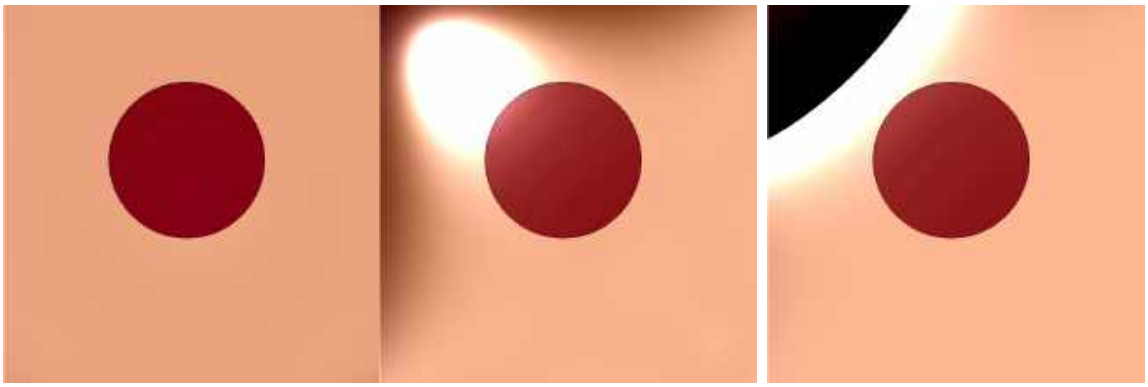
Figure B.3: Synthetic melanomas with removed non-homogeneous illumination, Third order polynomial



(a) Fourth order polynomial circular mask (b) Fourth order polynomial circular not centered mask (c) Fourth order polynomial diagonal mask

(d) Fourth order polynomial horizontal mask (e) Fourth order polynomial vertical mask

Figure B.4: Synthetic melanomas with removed non-homogeneous illumination, Fourth order polynomial



(a) Fifth order polynomial circular mask

(b) Fifth order polynomial circular not centered mask

(c) Fifth order polynomial diagonal mask

(d) Fifth order polynomial horizontal mask

(e) Fifth order polynomial vertical mask

Figure B.5: Synthetic melanomas with removed non-homogeneous illumination, Fifth order polynomial

B.2 Numerical results

54.3731	56.8492	28.5885	92.1611	25.1300	86.2729
44.5430	46.5642	28.5123	89.9647	24.9917	83.2669
	5.7776	77.0055	6.9397	75.5467	
	5.7776	75.3963	6.9397	73.5753	

Table B.1: RMSE between non-homogeneos illuminated synthetic melanoma and original before (right) and after (left) the algorithm, for just the melanoma(down) and for the whole image(up): Polynomial 1st order

26.1739	56.8492	25.0065	92.1611	16.5265	86.2729
21.5826	46.5642	24.9918	89.9647	16.5141	83.2669
	5.7775	77.0055	6.9402	75.5467	
	5.7775	75.3963	6.9402	73.5753	

Table B.2: RMSE between non-homogeneos illuminated synthetic melanoma and original before (right) and after (left) the algorithm, for just the melanoma(down) and for the whole image(up): Polynomial 2nd order

26.3464	56.8492	42.1201	92.1611	29.2802	86.2729
21.7175	46.5642	42.0017	89.9647	29.2788	83.2669
	5.7763	77.0055	6.9362	75.5467	
	5.7763	75.3963	6.9362	73.5753	

Table B.3: RMSE between non-homogeneos illuminated synthetic melanoma and original before (right) and after (left) the algorithm, for just the melanoma(down) and for the whole image(up): Polynomial 3rd order

18.7020	56.8492	73.2195	92.1611	32.8651	86.2729
15.4914	46.5642	73.1753	89.9647	32.8647	83.2669
		5.9117	77.0055	6.9369	75.5467
		5.9117	75.3963	6.9369	73.5753

Table B.4: RMSE between non-homogeneos illuminated synthetic melanoma and original before (right) and after (left) the algorithm, for just the melanoma(down) and for the whole image(up): Polynomial 4th order

19.0116	56.8492	41.9326	92.1611	60.5686	86.2729
15.7404	46.5642	41.8893	89.9647	60.5685	83.2669
		5.8725	77.0055	9.9140	75.5467
		5.8725	75.3963	9.9140	73.5753

Table B.5: RMSE between non-homogeneos illuminated synthetic melanoma and original before (right) and after (left) the algorithm, for just the melanoma(down) and for the whole image(up): Polynomial 5th order

B.3 Subjective results

(a) Mole 1 before non-homogeneous illumination correction

(b) Mole 1 after non-homogeneous illumination correction

Figure B.6: Mole 1 before and after non-homogeneous illumination correction

(a) Mole 1 before non-homogeneous illumination correction

(b) Mole 1 after non-homogeneous illumination correction

Figure B.7: Mole 1 before and after non-homogeneous illumination correction

(a) Mole 1 before non-homogeneous illumination correction

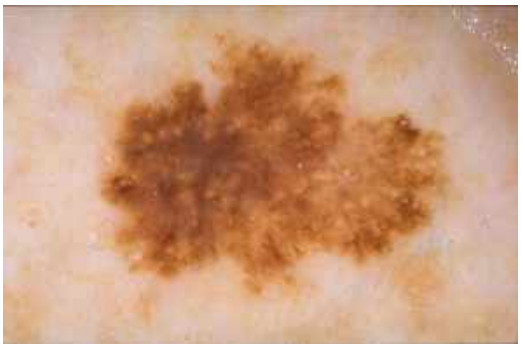
(b) Mole 1 after non-homogeneous illumination correction

Figure B.8: Mole 1 before and after non-homogeneous illumination correction

(a) Mole 1 before non-homogeneous illumination correction

(b) Mole 1 after non-homogeneous illumination correction

Figure B.9: Mole 1 before and after non-homogeneous illumination correction



(a) Mole 1 before non-homogeneous illumination correction

(b) Mole 1 after non-homogeneous illumination correction

Figure B.10: Mole 1 before and after non-homogeneous illumination correction

(a) Mole 1 before non-homogeneous illumination correction

(b) Mole 1 after non-homogeneous illumination correction

Figure B.11: Mole 1 before and after non-homogeneous illumination correction

(a) Mole 1 before non-homogeneous illumination correction

(b) Mole 1 after non-homogeneous illumination correction

Figure B.12: Mole 1 before and after non-homogeneous illumination correction