A HYBRID APPROACH TO SEGMENTING HAIR IN DERMOSCOPIC IMAGES USING A UNIVERSAL KERNEL

by

Nhi Hoang Nguyen

B.Eng.(Hons), The University of Sydney, Australia, 2005

a THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
in the School
of
Computing Science

© Nhi Hoang Nguyen 2009
SIMON FRASER UNIVERSITY
Fall 2009

All rights reserved. This work may not be reproduced in whole or in part, by photocopy or other means, without the permission of the author.
APPROVAL

Name: Nhi Hoang Nguyen
Degree: Master of Science
Title of Thesis: A hybrid approach to segmenting hair in dermoscopic images using a universal kernel

Examinining Committee: Dr. Brian Funt
Professor, School of Computing Science
Chair

Dr. M. Stella Atkins
Professor, School of Computing Science
Senior Supervisor

Dr. Tim K. Lee
Adjunct Professor, School of Computing Science
Supervisor

Dr. Ghassan Hamarneh
Associate Professor, School of Computing Science
Examiner

Date Approved: Nov 2, 2009
Declaration of Partial Copyright Licence

The author, whose copyright is declared on the title page of this work, has granted to Simon Fraser University the right to lend this thesis, project or extended essay to users of the Simon Fraser University Library, and to make partial or single copies only for such users or in response to a request from the library of any other university, or other educational institution, on its own behalf or for one of its users.

The author has further granted permission to Simon Fraser University to keep or make a digital copy for use in its circulating collection (currently available to the public at the “Institutional Repository” link of the SFU Library website <www.lib.sfu.ca> at: <http://ir.lib.sfu.ca/handle/1892/112>) and, without changing the content, to translate the thesis/project or extended essays, if technically possible, to any medium or format for the purpose of preservation of the digital work.

The author has further agreed that permission for multiple copying of this work for scholarly purposes may be granted by either the author or the Dean of Graduate Studies.

It is understood that copying or publication of this work for financial gain shall not be allowed without the author’s written permission.

Permission for public performance, or limited permission for private scholarly use, of any multimedia materials forming part of this work, may have been granted by the author. This information may be found on the separately catalogued multimedia material and in the signed Partial Copyright Licence.

While licensing SFU to permit the above uses, the author retains copyright in the thesis, project or extended essays, including the right to change the work for subsequent purposes, including editing and publishing the work in whole or in part, and licensing other parties, as the author may desire.

The original Partial Copyright Licence attesting to these terms, and signed by this author, may be found in the original bound copy of this work, retained in the Simon Fraser University Archive.

Simon Fraser University Library
Burnaby, BC, Canada
Abstract

Hair occlusion often causes automated melanoma diagnostic systems to fail. We present a new method to segment hair in dermoscopic images. First, all possible dark and light hairs are amplified without prejudice with a universal matched filtering kernel. We then process the filter response with a novel tracing algorithm to get a raw hair mask. This raw mask is skeletonized to contain only the centerlines of all the possible hairs. Then the centerlines are verified by applying a model checker on the response and the original images. If a centerline indeed corresponds to a hair, the hair is reconstructed; otherwise it is rejected. The result is a clean hair mask which can be used to disocclude hair. Application on real dermoscopic images yields good results for thick hair of varying colours. The algorithm also performs well on skin images with a mixture of both dark and light hair.

**Keywords:** image segmentation; medical image segmentation; hair segmentation; hair disocclusion; matched filtering

**Subject terms:** image processing; medical imaging; dermoscopic image processing
In the memory of my brother, Nguyen Hoang Huy.
Acknowledgments

The work described in this thesis would have not been possible without the support of the people involved, directly or indirectly.

I want to express my gratitude to Dr. Stella Atkins, my senior supervisor, for her constant support and guidance during my MSc degree. I’m especially thankful for the support I received from her during my personal hard time and during the course of my thesis. I also want to sincerely thank Dr. Tim Lee, my supervisor, for his academic guidance, his patience and the inspiration he gave me. Another person I’m grateful to is Mr. Steven Bergner, whose insightful ideas and unwavering support were of great help on both academic and personal levels. I also want to thank Dr. Ghassan Hamarneh, my examiner, for his detailed and thoughtful comments on my work.

This thesis concludes my Master’s degree, which spanned over the course of two years. These two years were among the hardest ones in my life, and I’m thankful to the people who were there for me and offered their friendship and support. This applies especially to my friends in the Computer Science department at Simon Fraser University.

Finally, as always, I am very grateful to and for my family, who have always loved, cared and supported me with every step I make.
Contents

Approval ii

Abstract iii

Dedication iv

Acknowledgments v

Contents vi

List of Tables ix

List of Figures x

1 Introduction 1

2 Background 3
  2.1 Hair segmentation ............................................. 3
  2.2 Retinal image segmentation ..................................... 5
  2.3 Matched filtering .............................................. 7
  2.4 Hair in dermoscopic images .................................... 8
    2.4.1 Properties of hair ....................................... 8

3 Initial algorithm design 10
  3.1 Phantom hair builder ......................................... 10
  3.2 Method overview ............................................. 13
    3.2.1 Assumptions ........................................... 13

vi
3.2.2 Phase 1 ........................................... 14
3.2.3 Phase 2 ........................................... 15
3.3 Phase 1: Identifying the hair centerlines ......................... 15
  3.3.1 Matched filtering .................................. 15
  3.3.2 Smoothing ......................................... 19
  3.3.3 Thresholding ...................................... 21
  3.3.4 Length filtering ................................... 23
  3.3.5 Morphological thinning ............................... 24
3.4 Phase 2: Reconstructing the hair from the centerlines .......... 25
  3.4.1 Gaussian curve fitting ............................... 25
  3.4.2 Length filtering ................................... 26
  3.4.3 Disocclusion ...................................... 26
3.5 Method analysis ........................................ 28
  3.5.1 Quantitative evaluation .............................. 28
  3.5.2 Qualitative evaluation ............................... 29
  3.5.3 Experiments ........................................ 29
3.6 Discussion ............................................. 33

4 Revised algorithm design ...................................... 38
  4.1 Method overview ...................................... 38
  4.2 Phase 1: Identifying the hair centerlines ..................... 38
    4.2.1 Matched filtering .................................. 39
    4.2.2 Processing the filter response ....................... 40
    4.2.3 Morphological thinning ............................... 43
  4.3 Phase 2: Verifying and reconstructing the hairs ............... 43
    4.3.1 Three-peak response test ............................ 43
    4.3.2 Gaussian profile test ................................ 45
    4.3.3 Hair reconstruction .................................. 45
  4.4 Phase 3: Post-processing ................................ 46
    4.4.1 Region growing ..................................... 46
    4.4.2 Length filtering ..................................... 48
  4.5 Method analysis ........................................ 49
    4.5.1 Evaluation method ................................... 49
4.5.2 Experiments ................................................................. 49
4.5.3 Quantitative assessment: Application on images with phantom hairs . 50
4.5.4 Qualitative assessment: Application on real dermoscopic images . . 52
4.6 Discussion ................................................................. 52

5 Conclusion ..................................................................... 56
  5.1 Conclusion ................................................................. 56
  5.2 Future work .............................................................. 56

Bibliography ................................................................. 58
List of Tables

3.1 Diagnostic accuracy and computation time of different angular resolutions . . 32
3.2 Quantitative assessment for the pilot experiment . . . . . . . . . . . . . . . . 33
4.1 Quantitative evaluation of the phantom hair experiment result . . . . . . . 50
List of Figures

2.1 Examples of dermoscopic images .............................................. 6
2.2 Intensity profile of a typical hair .............................................. 9
3.1 An example of phantom hair ..................................................... 11
3.2 Phantom hair builder demonstration ......................................... 12
3.3 Light phantom hair ............................................................... 13
3.4 Hair disocclusion process ......................................................... 14
3.5 Illustration of the filtering process on one-dimensional signals ....... 17
3.6 A 2-D horizontal kernel ......................................................... 19
3.7 Sample filter response of an input image .................................... 19
3.8 Effect of smoothing on the filter response .................................. 20
3.9 Quadrants of the co-occurrence matrix ..................................... 22
3.10 Thresholding the filter response using local entropy ................... 23
3.11 Morphological thinning on the raw hair mask ......................... 24
3.12 Hair pixels reconstruction ...................................................... 27
3.13 Final length filtering step ...................................................... 27
3.14 Final output of the algorithm .................................................. 28
3.15 Phantom hair image used in optimal angular resolution experiment . 31
3.16 Diagnostic accuracy of different angular resolutions .................. 33
3.17 Pilot experiment ................................................................. 34
3.18 Examples of the drawback of the initial design ......................... 35
3.19 The drawback of local entropy thresholding ............................. 36
3.20 Results from Chanwimaluang and Fan’s implementation ............ 37
4.1 Revised hair segmentation process ............................................ 39
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2</td>
<td>Revised algorithm demonstration: Input and filter response</td>
<td>40</td>
</tr>
<tr>
<td>4.3</td>
<td>Computing the range of the hair response</td>
<td>41</td>
</tr>
<tr>
<td>4.4</td>
<td>Length and curvature tests on the filter response</td>
<td>42</td>
</tr>
<tr>
<td>4.5</td>
<td>Intensity profile of response from true positives and false positives</td>
<td>44</td>
</tr>
<tr>
<td>4.6</td>
<td>Three-peak response test</td>
<td>44</td>
</tr>
<tr>
<td>4.7</td>
<td>Reconstructed hair mask from skeletonized mask</td>
<td>45</td>
</tr>
<tr>
<td>4.8</td>
<td>Discontinuity at intersections</td>
<td>47</td>
</tr>
<tr>
<td>4.9</td>
<td>Direction constraint implementation</td>
<td>47</td>
</tr>
<tr>
<td>4.10</td>
<td>Region growing demonstration</td>
<td>48</td>
</tr>
<tr>
<td>4.11</td>
<td>Application of the revised algorithm on phantom hairs</td>
<td>51</td>
</tr>
<tr>
<td>4.12</td>
<td>Example of segmentation results on easy and moderately difficult cases</td>
<td>53</td>
</tr>
<tr>
<td>4.13</td>
<td>Comparison of the initial and the revised methods on difficult cases</td>
<td>54</td>
</tr>
<tr>
<td>4.14</td>
<td>Example of failed segmentation results on difficult cases</td>
<td>55</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Malignant melanoma is a serious, deadly kind of skin cancer. As with any other type of cancer, early detection is key in improving treatment efficiency and increasing the survival rate. An automated diagnostic system which uses image processing techniques to analyze skin images would be a quick, accessible diagnosis tool to aid dermatologists and even general practitioners in the early detection of melanoma. Such images are obtained with a handheld microscope-like device called a “dermoscope” and thus called “dermoscopic images”. They can reveal important diagnostic information on melanomas and skin lesions otherwise not visible to the unaided eye.

The main challenge in such system is the correct segmentation and classification of moles, often occluded by hair in images obtained with a dermoscope. Hair occlusion causes segmentation algorithms to fail to identify the correct nevus border, and can cause errors in estimating texture measures. The accuracy of hair disocclusion is important not only to the mole segmentation results, but also to the information contained in the nevi, as inaccurate disocclusion can lead to pathological information being lost, resulting in wrong diagnosis.

In this thesis, we present a new method to disocclude hair in dermoscopic images using a universal kernel, which is capable of amplifying both dark and light hair without prior knowledge of the hair type. We use a hybrid model of matched filtering, verification and reconstruction to segment hair.

The thesis is divided into five chapters. In chapter 1, we introduce the problem and the overall structure of the thesis. Chapter 2 provides the relevant background information to the problem: existing methods to segment hair in the literature, similar problems, key concepts and the main characteristics of the problem. Chapter 3 describes our initial
algorithm design and its results, as well as details on our phantom hair builder. Chapter 4 is our primary focus, the revised algorithm design, where we overcome the shortcomings of the initial design and propose a novel method to segment hair of both dark and light colours. It also contains experiment results on synthetic images with phantom hairs and on real dermoscopic images. Chapter 5 concludes the thesis and addresses potential for future work.
Chapter 2

Background

2.1 Hair segmentation

The most popular software for hair removal in dermoscopic images is DullRazor® by Lee et al. [1]. This program applies morphological closing operations\(^1\) separately on all three RGB channels of the input image, with three structure elements in horizontal, diagonal and vertical directions. The maximum result out of the three structure elements is taken as the response at each pixel. A hair mask is generated for each channel by taking the difference between the original channel value and the response at each pixel, then thresholding this value using a hard threshold. The software then removes the hair using this hair mask and repaints the image using linear interpolation with two nearby non-hair pixels. While being generally effective, this software has a limitation on its applicability: it only works on thick dark hair, as the authors clearly document in their paper [1].

Schmid-Saugeon et al. [2] use a similar approach but the morphological closing operation is applied to the three components of the \(L^*u^*v^*\) uniform colour space. Then a hair mask is created by thresholding the difference in the luminance component before and after morphological closing with a hard threshold. It should be noted that hair removal is only a step in their diagnostic system for pigmented skin lesions. This step is mentioned very briefly in the paper, without any study on the accuracy of the hair disocclusion scheme, as the authors favour physical hair removal over image processing techniques [2]. The major

\(^1\)Although morphological operations are usually applied on binary images, they can be applied on grayscale images too. The result of a grayscale morphological operation is also a grayscale image.
assumption the authors make in this paper is that hair is darker than its surrounding, which limits the algorithm’s application to dark hair only.

Fleming et al. [3] take a different approach to segmenting hair in dermoscopic images. Based on the observation that hairs are long, straight curvilinear structures with relatively constant width and curvature, they develop a tracing algorithm to follow and connect the curvilinear hair segments. First, they apply Steger’s line detection algorithm ([4]) on the input to get a list of line segments present in the image. This detection algorithm views the grayscale input image as a surface in which pixel intensity corresponds to surface height. The endpoints of each line segment are detected as the points where the first directional derivative vanishes and the second directional derivative has a large magnitude. For each line segment, a feature vector is calculated to reflect the constraints for hair segments: having length exceeding that of pigment network\(^2\) segments, relatively straight, contiguous and colinear with other line segments, and vary little in width. Then linear discriminant analysis is applied on the feature vector for each line segment in order to classify it as hair or not hair, with the most important components of the linear discriminant function being the number of points in the line segments and the mean curvature. The output of this step is a list of hair segments, with possible gaps. A graph search is done to fill the gaps and connect the hair segments. Then length filtering is implemented as the final step to reject the remaining false positives. This algorithm, despite being complicated, does not address the problem of intersecting hair, which is important, as shown in Figure 2.1(a)\(^3\). The paper also does not explicitly mention the case of light hair, and the results they present are on a small number of images with dark hair only.

A similar algorithm is implemented by Zhou et al. [6]. Their modified algorithm uses the same approach to identify the line segments (Steger’s line detection algorithm [4]), but it adds an extra step to detect intersections and analyze them to reconnect the line segments broken at intersections. Then it fits a curve to each hair segment and evaluates the fit to reject non-hair segments. The result in this paper clearly shows all the light hairs being left out in the segmentation; in other words, this algorithm is applicable to dark hair only.

Xie et al. [7] propose using the morphological closing-based top-hat operator to enhance hair, then thresholding the result using a “statistic threshold”. The “statistic threshold”

---

\(^2\)A pigment network is defined as a delicate, regular grid of brownish lines over a diffuse light-brown background [5]

\(^3\)We address the intersecting hair problem in section 4.4.1.
they use is actually a hard threshold of 5% on the ratio of the amount of hair pixels in the whole image. After this step, the false positives are eliminated using a size filtering algorithm which thresholds on the ratio between the length and the area of the detected objects. This paper focuses mainly on dark hair, with the authors making the assumption that blond or gray hair is also darker than its surrounding background. This is not a reasonable assumption; however, it is crucial that the object is darker than the background for their morphological closing-based top-hat operator to work. They argue that segmenting hair lighter than the background is a similar process: instead of the closing-based top-hat operator, the opening-based top-hat operator can be used to enhance light hair. However, this requires prior knowledge of the hair type and adds another parameter to the algorithm. The paper also does not address the case of input images with both dark and light hair, as seen in Figure 2.1(a), and the use of hard thresholding on the ratio of hair makes it rigid against variation in hair density.

Most existing methods for dermoscopic hair segmentation in the literature overlook or ignore the case of hair lighter than the background. This is partly due to the difficulty of the problem since light hair has similar characteristics to skin lines (i.e., lines that form the skin patterns or dermatoglyphics, which can be seen in Figure 2.1(b)). Furthermore, light hair is usually finer than dark hair, which makes segmentation difficult; in fact, most existing hair segmentation algorithms focus on thick dark hair only.

Figure 2.1 shows examples of hair, skin lines and pigment networks in dermoscopic images.

2.2 Retinal image segmentation

The analysis of blood vessels can provide useful insights into a patient’s condition in various pathologies. An example of such is the diagnosis of retinal blood vessels in ocular fundus images, whose abnormalities are linked to diseases and medical disorders such as hypertension, arteriosclerosis and diabetes. There are a number of algorithms for the automatic segmentation and registration of retinal blood vessels that are of interest to us. Blood vessels in retinal images and hair in dermoscopic images exhibit similar characteristics: they are thin, long structures with little variation in width and sparse occurrence. From an image processing point of view, these similarities mean algorithms developed for retinal blood vessel segmentation may be applicable or at least provide some insight to dermoscopic hair
Figure 2.1: Examples of dermoscopic images: (a) A dermoscopic image with light hair, dark hair and pigment networks, (b) A dermoscopic image with hair and skin lines segmentation.

Chaudhuri et al. [8] propose using a matched filter to amplify the retinal blood vessels. The matched filter is designed based on the observation that retinal blood vessels can be approximated by piecewise linear segments, are darker than their surrounding and have almost constant width across their length. These characteristics are modelled by a two-dimensional Gaussian matched filter with a zero mean to reduce the effect of background noise. A set of 12 kernels, created by rotating the base kernel in 15 degree successions, is used to cover all possible curvatures of the vessels. The filtering process is done by convolving the input image with all 12 kernels, taking the maximum response at each pixel as the filter response. The whole image response is then thresholded using an automatic thresholding algorithm that maximizes the inter-class intensity variance [9] to create a binary vessel map.

Various algorithms based on Chaudhuri’s matched filtering idea have been documented in retinal blood vessel segmentation literature. Hoover et al. [10] propose a different approach to process the matched filtering response. They use an iterative probing process to label pixels as vessel or background. In each iteration, a set of criteria is evaluated in order to determine the threshold for the region being probed. The threshold varies spatially across the image. A different approach to thresholding is proposed by Chanwimaluang and Fan [11]. This approach computes a single “optimal” threshold for the whole image using a modified local entropy thresholding algorithm, based on the local entropy algorithm introduced in [12]. This threshold is selected, out of all possible threshold values, so that it maximizes the
total second order entropy of the object and the background.

2.3 Matched filtering

Matched filtering is a technique usually used in telecommunications to detect a known signal in a noisy input. In principle, matched filtering is the process of using a kernel to detect signals similar to the kernel in the input. It involves convolving the noisy input with a time-reversed version of the kernel (cross-correlation). Signals similar to the kernel are amplified in the response; hence, it is possible to threshold the response to discard the noise and background, effectively detecting the desired objects in the input.

In the context of image processing, matched filtering is the cross correlation of a kernel with the input image. The kernel is designed to have characteristics resembling those of the object to be detected. Given an input image $I$ of size $(M, N)$ and a kernel $K$ of size $(P, Q)$, the output of matched filtering the image with kernel $K$ is:

$$R(i,j) = \sum_{m=0}^{(M-1)} \sum_{n=0}^{(N-1)} K(m,n) \times \bar{I}(m+i, n+j)$$ (2.1)

where $0 \leq i < M + P - 1$, $0 \leq j < N + Q - 1$ and $\bar{I}$ is the complex conjugate of $I$. Since we are dealing only with real numbers, the above equation can be simplified to:

$$R(i,j) = \sum_{m=0}^{(M-1)} \sum_{n=0}^{(N-1)} K(m,n) \times I(m+i, n+j)$$ (2.2)

Matched filtering kernels used in image segmentation are usually designed to have zero sum to reduce the effect of background noise and other artifacts. As Equation 2.2 implies, in background areas, where the intensity does not vary much, the filter response will be zero or close to zero due to this zero-sum property of the kernel.

Chaudhuri et al. [8] popularize the use of matched filtering for image segmentation in their retinal blood vessel detection paper ([8]). Their two-dimensional Gaussian filter is characterized by:

$$K(x, y) = -e^{-\frac{x^2}{2\sigma^2}}$$ (2.3)

for $|y| < L/2$, with $L$ being the length of the piecewise blood vessels.
2.4 Hair in dermoscopic images

We study skin images obtained with a dermoscope in clinical studies for characteristics of hair, in particular; the length, width, curvature and colour of hair among all the skin images of patients with different ethnic backgrounds.

Our set of data consists of 40 real dermoscopic images collected from Argenziano et al.’s Interactive atlas of dermoscopy [5] and Marghoob et al.’s Atlas of dermoscopy [13]. We choose the images which have good contrast and do not have other artifacts such as ink, oil and marker lines.

2.4.1 Properties of hair

We observe that dark hair on skin has properties similar to those of blood vessels in retinal images mentioned in [8]. The properties as applied to dark hair are:

- Hair has small curvature and can be approximated by piece-wise linear segments, even for curly hair.

- Although the hair colour varies from patient to patient, the intensity profile taken in the direction perpendicular to its length can be approximated by a Gaussian curve. Figure 2.2 illustrates these properties of both dark hair and light hair, taken on the V channel of the HSV version of the input images.

- The width of hair stays approximately unchanged along its length.

Light hair exhibits similar properties, with the only exception being the shape of the Gaussian profile: in light hair cases, the Gaussian curve is reversed.
Figure 2.2: Intensity profile of a typical hair: (a) A dermoscopic image with dark hair, (b) Intensity profile taken along the line marked in red in (a), (c) A dermoscopic image with light hair, (d) Intensity profile taken along the line marked in red in (c)
Chapter 3

Initial algorithm design

3.1 Phantom hair builder

For evaluation purposes, we need to know the ground truth of each image used in segmentation. Using images with real hair is unrealistic in this step, as it is not possible to create an absolute hair mask with pixel precision for real dermoscopic images. Hence we choose to use phantom hair implanted on top of skin images obtained with a dermoscope from the Interactive Atlas of Dermoscopy by Argenziano et al. [5]. Our phantom hair builder is a program which enables users to click arbitrarily on an image to indicate the approximate position, length and curvature of the hair. It then uses the points the user chooses as the control points of a spline to build the phantom hair. The hair can have any orientation and curvature. Users can also specify dark or light hair as a binary option. The phantom hair exhibits all three important properties of hair as mentioned earlier in this section. An example of phantom hair as produced by our program is shown in Figure 3.1.

The phantom hair builder receives input from the user’s mouse clicks. A parametric variational cubic spline curve is constructed from the click coordinates. This spline is then used as the skeleton of the phantom hair. The phantom hair is built to have perfect Gaussian intensity profiles across its width, in the direction perpendicular to its length. In order to do that, we use distance transform to compute the shortest distance from each pixel to the spline, then this distance is used to look up the corresponding value on the Gaussian curve to create a “base” Gaussian profile. Then we scale this profile based on our empirical values of hair properties in order to achieve a realistic looking hair. The scaling is done such that the shoulder of the Gaussian curve is equal to the average intensity of the surrounding area,
with the purpose of making the hair look more natural and blend well into the image. The scaled Gaussian function is computed with the following formula:

\[ G(x) = \frac{D}{g_{\text{max}} - g_{\text{min}}} \cdot G_0(x) + I - \frac{D}{g_{\text{max}} - g_{\text{min}}} \cdot g_{\text{max}} \]  

(3.1)

\[ = \frac{D}{g_{\text{max}} - g_{\text{min}}} \cdot (G_0(x) - g_{\text{max}}) + I \]  

(3.2)

where:

- \( I \) is the average intensity of the bounding box of the phantom hair,
- \( D \) is the depth of the Gaussian curve (defined based on the average intensity difference between hair and skin in our data set),
- \( g_{\text{max}} \) and \( g_{\text{min}} \) are respectively the maximum and minimum of the base Gaussian function in the interval \([-\sigma, \sigma] \):

\[ G_0(x) = -e^{-\frac{x^2}{2\sigma^2}} \]  

(3.3)

\( (g_{\text{max}} = G_0(\sigma) = -e^{-\frac{1}{2}} \) and \( g_{\text{min}} = G_0(0) = -1) \)

In effect, equation 3.2 takes the base Gaussian curve in equation 3.3, “stretches” it to a realistic height (defined by \( D \)) then “shifts” it to the intensity of the surrounding skin.

Figure 3.2 illustrates the steps in creating a phantom hair from user’s input. First, the user is presented with a background image of real skin (Figure 3.2(a)). The user then can
click arbitrarily on the image to specify the direction, length and curvature of the phantom hair (Figure 3.2(b), the points where the user clicks are indicated by the square dots). The phantom hair builder constructs a spline based on these points and calculates the distance from each pixel in the image to the spline using distance transform (Figure 3.2(c): the distance transform mask is confined to a bounding box to improve speed). It then computes the cross section profile for this hair using Equation 3.2. The distance transform mask contains the distance from each point to the hair spline, thus it is trivial to get the set of all the points within $\sigma$ pixels of the spline using this mask. The intensity of each point in this set ($I_p$) is looked up using its distance to the spline ($d_s$) and the cross section intensity curve, using $I_p = G(d_s)$. Finally, the corresponding points on the skin image are set to their corresponding $I_p$ (Figure 3.2(d)).

![Figure 3.2: Phantom hair builder demonstration: (a) The background image presented to the user, (b) The spline as built from user’s mouse clicks, (c) The distance transform mask built from the spline, (d) The final phantom hair implanted on the background image.](image)

Our phantom hair builder is also capable of building light hair. The process for light hair is similar to that for dark hair, except that the base Gaussian curve is reversed. Figure
3.3 shows a typical phantom light hair and its properties.

![Figure 3.3: Light phantom hair: (a) The hair spline, (b) The resulting phantom hair, (c) Intensity profile of the phantom hair (taken along the line segment marked in red in (a))](image)

### 3.2 Method overview

The hair segmentation process consists of two phases. The first phase is to identify the centerline of possible hair. The output of this phase is a skeletonised hair mask. The second phase takes this skeletonised mask and reconstructs the hair, as well as eliminates the false centrelines detected in phase 1. Figure 3.4 is a visual overview of the algorithm.

#### 3.2.1 Assumptions

We make the following assumptions in the design of the algorithm:
Hair has constant width across its length. As Gaussian curves are symmetric, we assume that this width is $2\sigma$ pixels, where $\sigma$ is an integer.

Hair can be approximated by line segments; in other words, hair is piecewise linear.

### 3.2.2 Phase 1

Phase 1 of the algorithm identifies the possible hair based on the observed properties of hair, most notably the Gaussian nature of the cross section intensity profile. First, the input image is converted to grayscale by taking the V channel of the HSV version of the image\(^1\). Then matched filtering is applied on this grayscale image to amplify all the possible hair. We use a special matched filtering process, which matches not only the shapes similar to the kernel, but also the shapes that are similar to the inverted version of it. This is a major difference from traditional matched filtering. Section 3.3.1 elaborates on our matched filter. The next step is to smooth the filter response to bridge the “ghost” response to the true positive response, preparing it for the morphological thinning step. This is due to the output of our modified matched filter having three peaks at each matched position. More details on the three-peak response and the smoothing process are given in section 3.3.1 and

---

\(^1\)The V channel is chosen based on the fact that this channel reflects the Gaussian intensity profile property the most, and all other assumptions (constant width and piecewise linearity of hair) are observable in this channel as well. As our algorithm considers only intensity, it is not necessary to include colour information in processing.
section 3.3.2. After smoothing, the response is thresholded to produce a raw hair mask. The method used is a modified version of the local entropy thresholding algorithm, described in [11]. The raw hair mask is then refined by a length filtering process to eliminate small blobs that are unlikely to be hair, based on size criteria. The final step in phase 1 is morphological thinning. It is applied on the refined hair mask to produce a skeletonised version containing only the centerlines of possible hair.

### 3.2.3 Phase 2

Phase 2 takes the skeletonised hair mask as input and attempts to reconstruct the hair pixels as well as eliminate the false response, based on the centerlines and the original input image. It starts with a Gaussian curve fitting procedure, which takes each pixel on the centerlines and checks whether it is indeed the centre pixel of a hair. It does so by trying to expand \( \sigma \) pixels on each side of the centerline and fit a Gaussian curve on these \( 2\sigma + 1 \) pixels. If the Gaussian curve fits within a certain error threshold, the pixel is accepted as a true hair centerline pixel, and the mask is expanded with the aforementioned \( 2\sigma + 1 \) pixels set to “hair”. If it is impossible to fit a Gaussian curve centered at the pixel, the pixel is rejected and set to background. At the end of the fitting process, the hair mask is reconstructed to contain full-size hair. Length filtering is applied once again on this new mask to refine it and produce the final hair mask. If there is a need for recovering the image, this hair mask is used to in-paint the image and recover the pixels occluded by hair, using algorithms such as linear interpolation (same as the one used in DullRazor [1]) or in-painting [14]. In this thesis we focus on hair segmentation and on comparing our segmentation result with DullRazor so we use linear interpolation for this purpose, even though it is less accurate than in-painting [14].

### 3.3 Phase 1: Identifying the hair centerlines

#### 3.3.1 Matched filtering

We use a different approach to matched filtering, differing from the traditional approach in that instead of cross correlating the template with the input, we calculate the absolute value of their normalized cross correlation. To establish the foundation of our approach, we first illustrate the process in the one-dimensional case.
CHAPTER 3. INITIAL ALGORITHM DESIGN

Given two signals characterised by the following equations\(^2\):

\[
H_1(x) = A_1 - d_1 e^{-\frac{x^2}{2\sigma^2}}
\]  

(3.4)

and

\[
H_2(x) = A_2 + d_2 e^{-\frac{x^2}{2\sigma^2}}
\]  

(3.5)

with \(A_1, A_2, d_1\) and \(d_2\) being constants, and a kernel characterised by:

\[
K_1(x) = A_3 - d_3 e^{-\frac{x^2}{2\sigma^2}}
\]  

(3.6)

with \(d_3\) being a constant, and \(A_3\) being chosen so that

\[
\sum_{x=-3\sigma}^{3\sigma} K_1(x) = 0
\]

An example of such signals and kernel are shown in Figure 3.5.

If we take the normalized cross correlation of the two signals with the kernel, the output reflects the difference in their shapes (Figure 3.5(d), 3.5(e)). However, if we take the absolute value after normalized cross correlation, the two opposite signals produce the same response (Figure 3.5(f), 3.5(g)).

This example illustrates our modified matched filtering process and how it works. With an appropriately designed filter, normalized cross correlation can amplify both light and dark hair, making them stand out from the rest of the image.

Given an input image \(I\) of size \((P,Q)\) and a kernel\(^3\) \(K\) of size \((M,N)\), our matched filtering function uses Matlab’s implementation of the fast normalized cross-correlation in [15]. The response of our modified matched filter is characterised by:

\[
R(u, v) = \max_i \left| \frac{\sum_{x,y} [I(x, y) - \bar{I}_{u,v}] [K_i(x - u, y - v) - \bar{K}_i]}{\left(\sum_{x,y} [I(x, y) - \bar{I}_{u,v}]^2 \sum_{x,y} [K_i(x - u, y - v) - \bar{K}_i]^2\right)^{0.5}} \right|
\]

(3.7)

where:

- \(K_i\) is a kernel in the series of rotated kernels \((i = 1, 2, .., 18)\)

\(^2\)\(H_1\) can be thought of as corresponding to a signal darker than the background, and \(H_2\) as a signal lighter than the background

\(^3\)The term “kernel” and “template” are used interchangably in this thesis with the same meaning
Figure 3.5: Illustration of the filtering process on one-dimensional signals: (a) $H_1$, (b) $H_2$, (c) $K_1$, (d) The normalized cross correlation of $H_1$ and $K_1$, (e) The normalized cross correlation of $H_2$ and $K_1$, (f) The absolute value of the normalized cross correlation of $H_1$ and $K_1$, (g) The absolute value of the normalized cross correlation of $H_2$ and $K_1$
CHAPTER 3. INITIAL ALGORITHM DESIGN

- $\bar{K}_i$ is the mean of $K_i$
- $\bar{I}_{u,v}$ is the mean of $I(x, y)$ in the region under the template

Based on the observed characteristics of hair (section 2.4.1) and the properties of normalized cross correlation, we design the kernel to have a Gaussian shape, similar to the intensity profile of dark hair. Such kernels have been used in retinal blood vessel segmentation. We adopt the kernel designed by Chaudhuri et al. [8]:

$$K(x, y) = -e^{-\frac{x^2}{2\sigma^2}}$$

for $|y| \leq L/2$, with $L$ being the length of the piecewise linear hair segment.

It should be noted that this two-dimensional kernel does not depend on $y$ and hence is not a conventional two-dimensional Gaussian kernel usually seen in image processing literature, but rather a stack of one-dimensional Gaussian kernels. In fact, the two-dimensional aspect is only a feature to speed up the filtering process and filter out structures similar to hair but shorter. A single one-dimensional kernel (i.e. a kernel with $L = 1$) is capable of amplifying the hair, and it has the advantage of being more sensitive when hair is curly and the piecewise linear assumption does not hold. However, we take hair as piecewise linear and so, we try to amplify it one straight segment at a time by using multiple one-dimensional kernels concatenated together. This also has the advantage of filtering out the false response produced by pigment networks, which are present in nevi. Pigment networks usually have an intensity profile of similar shape with hair, but weaker in amplitude; and their length is usually less than the piecewise linear segments in hair. The kernel is adjusted to have zero sum to reduce the effect of background noise. This zero-sum property is achieved by subtracting the kernel means from all points in the kernel. Figure 3.6 shows an example of such kernels.

Due to the fact that hair can have any orientation and curvature, and that in matched filtering, the signal is amplified only when the kernel is aligned at the same angle with the signal, we have to rotate the kernel to cover all possible orientations. At each pixel, the maximum response out of all alignments is taken as the response at that location. The angle corresponding to the maximum response is also recorded at each pixel. This information will be useful in a later step (Gaussian curve fitting, section 3.4.1). We assign one of our pilot experiments to determine the optimal rotation angle. This experiment examines and compares the accuracy of segmentation using different rotation angles ($5, 10, 15, 20, 30, 45, 60, 90$...).
CHAPTER 3. INITIAL ALGORITHM DESIGN

Figure 3.6: A 2-D horizontal kernel, with $L = 40$ and $\sigma = 2$

Figure 3.7: Sample filter response of an input image with hairs of dark and light colours

degrees) in the matched filtering process. Details on this experiment as well as its result are given in section 3.5.3.

Figure 3.7 shows an input image and its filter response. We can see that the hairs are amplified and stand out from the background.

3.3.2 Smoothing

The design of the kernel and taking the absolute value of the filter response makes it possible to amplify both dark and light hair using a universal kernel. However, it also introduces two lower peaks alongside the main peak where the signal matches. The “ghost” response can be seen in the one-dimensional illustrative example (Figure 3.5), as well as in the filter response in Figure 3.7(b). Applying thresholding right on this response would create many
false positives, resulting in an incorrect hair mask where each correctly identified hair is accompanied by two false hairs running in parallel. The smoothing step aims at solving this problem.

We use Gaussian smoothing to bridge the “ghost” responses to the peak response, creating a single peak instead of three peaks at each matched point. Gaussian smoothing is chosen for its low-pass property, especially in areas where the three-peak response occurs. In such cases, it attenuates the highest peak, bringing it closer to the lower peaks and increases the intensity levels of the “valleys” between the peaks, effectively bridging the three peaks into one. This prepares the response for the thresholding step.

Figure 3.8 shows an input image and its filter response before and after smoothing. The intensity profiles (Figures 3.8(d) and 3.8(e)) show the effect of smoothing on the shape of the response peaks.
3.3.3 Thresholding

Thresholding the matched filter response is not trivial, as dermoscopic images usually have non-uniform background, as well as other objects in the foreground that are not hair (nevi, pigment networks etc.). As we cross correlate the image with the kernel, these objects and artifacts can generate non-zero response, with a lower magnitude than response from hair. For this reason, the filter output needs to be thresholded to separate the response from hair, which has higher value, from the rest. We use the local entropy method, described in [11] based on the original idea in [12], to threshold the filter response. Local entropy is a measure of information present in a binary image. It is introduced by Pal and Pal in [12], with the intention of preserving the information present in the spatial structure in an image, which is not possible with previous histogram-based measures of entropy. It uses the co-occurrence matrix, which is a well-known measure of correlation of a particular property between pixels in an image. In the context of grayscale image segmentation, the co-occurrence matrix $T$ is an integer matrix of size $Y \times Y$, where $Y$ is the number of intensity levels in the whole image. It is calculated as follows:

$$T_{ij} = \sum_{l=1}^{P} \sum_{k=1}^{Q} \delta$$

where

$$\delta = \begin{cases} 
1 & \text{if } f(l, k) = i \text{ and } f(l, k + 1) = j \\
0 & \text{otherwise}
\end{cases}$$

and $(P, Q)$ is the size of the image.

This co-occurrence matrix can be used to compute the probability of co-occurrence of any intensity levels $i$ and $j$ at neighbouring pixels in the image:

$$p_{ij} = \frac{T_{ij}}{\sum_i \sum_j T_{ij}}$$

A threshold $s$ divides the co-occurrence matrix into four quadrants (Figure 3.9) [12]. The optimal threshold is the one that maximizes the total second order local entropy of the object and the background, which is defined as:

$$H_T^{(2)}(s) = H_A^{(2)}(s) + H_C^{(2)}(s)$$
CHAPTER 3. INITIAL ALGORITHM DESIGN

Figure 3.9: Quadrants of the co-occurrence matrix

$H_A^{(2)}(s)$ and $H_C^{(2)}(s)$ are the second order local entropy of the object and the background respectively. They are defined in [11] as:

$$H_A^{(2)}(s) = -\frac{1}{2} \sum_{i=0}^{s} \sum_{j=0}^{s} P_{ij}^A \log_2 P_{ij}^A$$

(3.12)

and

$$H_C^{(2)}(s) = -\frac{1}{2} \sum_{i=s+1}^{Y-1} \sum_{j=s+1}^{Y-1} P_{ij}^C \log_2 P_{ij}^C$$

(3.13)

where

$$P_{ij}^A = \frac{T_{ij}}{s} \sum_{i=0}^{s} \sum_{j=0}^{s} T_{ij}$$

(3.14)

and

$$P_{ij}^C = \frac{T_{ij}}{Y-1} \sum_{i=s+1}^{Y-1} \sum_{j=s+1}^{Y-1} T_{ij}$$

(3.15)

We notice from trial runs that the original local entropy thresholding in [11] is too low to segment hair. In fact, the authors of [11] announce a correction ([16]) for their retinal image segmentation papers which use local entropy thresholding. The correction addresses the change in the method used to calculate the co-occurrence matrix. They add what they call a “dithering effect” to the original co-occurrence matrix to increase the local entropy, so that Equation 3.9 becomes:

$$T_{ij} = T_{id} + 1$$

(3.16)

where $I(l, k) = i$ and $I(l, k + 1) = j$ and $I(l + 1, k + 1) = d$ [16].

We use this modified co-occurrence matrix in our local entropy calculation. The $s$ threshold selected is the one that maximizes $H_T^{(2)}(s)$ (Equation 3.11). It is used to threshold
3.3.4 Length filtering

As we can see from the sample hair mask in Figure 3.10, the raw hair mask includes many false positives. These false positives are usually resulting from pigment network segments and skin lines, whose intensity profiles resemble that of hair. However, their length is usually less than that of normal hair. Observing that hair is thin, long structures, we use length as a criterion to eliminate the objects in the hair mask whose length is less than a certain threshold.

First we identify the connected regions in the hair mask using label propagation.

Length filtering is done using label propagation with the eight-connected neighbourhood. Two pixels are considered to be in the same region if the first pixel is in one of the eight adjacent positions with the second. We use Matlab’s implementation of the label propagation algorithm outlined in [17]. The steps in the procedure are:

1. Run-length encode the input image.

2. Scan the runs, assigning preliminary labels and recording label equivalences in a local equivalence table.

Figure 3.10: Thresholding the filter response using local entropy: (a) The (smoothed) filter response, (b) The hair mask obtained by thresholding the response using the local entropy thresholding algorithm.
3. Resolve the equivalence classes.

4. Relabel the runs based on the resolved equivalence classes.

Label propagation identifies connected regions in the refined hair mask. We use the labels to compute the length of each region then rejects the ones under our length threshold. The rejected pixels are re-labelled as background, resulting in a new hair mask which is passed onto the next step. The effect of this step is demonstrated in Figure 3.11(a) which is computed from Figure 3.10(b). Details on setting the parameters in length filtering are given in section 3.5.3.

3.3.5 Morphological thinning

The final step in identifying the hair centerlines is morphologically thinning the raw hair mask to get the centerlines. As the “ghost” responses are symmetric and the smoothing step connects them to the real peak, morphological thinning returns the centerline of the hair, given that the response from hair is strong enough. Figure 3.11(b) shows an example of the effect of morphological thinning on the hair mask after length filtering.

The centerline mask is the final output of phase 1. We have identified all possible centerlines of hair, with emphasis on the number of true positives. This leads to false positives being accepted as well, as illustrated in Figure 3.11(b), and phase 2 aims at verifying if a centerline indeed corresponds to a real hair and reconstructing the hair in such cases.
3.4 Phase 2: Reconstructing the hair from the centerlines

Phase 1 returns a skeleton hair mask with possible hair centerlines. Phase 2 takes this mask as input, reconstructs the hair and produces the final, complete hair mask. It does so by using the knowledge that hair has a Gaussian intensity profile and is a thin, long structure. In essence, this phase is the verification step: if a centerline indeed corresponds to a hair, it is expanded to the full hair width; otherwise it is rejected. Thus the hair mask is expanded at the true positive points and cleaned up at the false positive points.

3.4.1 Gaussian curve fitting

The skeleton obtained in phase 1 provides a guideline as to where the centre of the hair is. Using it as input, we follow each pixel on the skeleton, take $2\sigma + 1$ pixels in the input image, in the perpendicular direction of the hair centering at this pixel and check if the intensity of the $2\sigma + 1$ pixels conforms to a Gaussian curve. If they do, we expand the skeleton at that point to cover the whole hair width; otherwise we reject the point (i.e. label it as background). This process rejects most of the false positives and reconstructs the true positives (hair) to their full width.

We know the direction of the hair at each pixel since the angle corresponding to the maximum response is recorded in the matched filtering process (section 3.3.1). We know hair has an intensity profile that can be modelled by the following equation:

$$I(x) = A + de^{-\frac{x^2}{2\sigma^2}}$$  \hspace{1cm} (3.17)

It should be noted that the above equation can model both dark hair and light hair profiles: $d < 0$ for dark hair and $d > 0$ for light hair. Using this information, we fit the model in equation 3.17 to the $2\sigma + 1$ pixels centered at the centerline and evaluate the goodness of fit. We use Matlab’s Curve Fitting Toolbox’s implementation of fitting a nonlinear model (equation 3.17) to data (the $2\sigma + 1$ pixels). This implementation is an iterative process which starts with an initial estimate for each coefficient, provides the fitted curve for the current set of coefficients, then adjusts the coefficients and determines whether the fit improves [18]. In our model, $d$ and $\sigma$ are the coefficients; and their starting values are set as follows:

- $\sigma$ is set to our assumed half-width of hair (see the assumptions in section 3.2.1)
- $d$ is set to $I_c - A$, where $I_c$ is the intensity of the centerline pixel. This value is set
based on the observation that if the pixels conform to a Gaussian curve, then the centerline pixel has the intensity of $I_c = I(0) = A + d$.

We also estimate $A$ to be the average intensity of the surrounding skin pixels.

When the curve fitting process stops\(^4\), we are returned with the “best” fit for $\sigma$ and $d$, and also with the goodness-of-fit statistics of the model to the data. We use the root mean square error in the goodness-of-fit measures as the deciding factor in accepting the pixels. The accepted pixels are set to hair in the hair mask, otherwise they are rejected and set as background.

The expanding process may leave single-pixel gaps within the reconstructed hair pixels, as only the pixels in the perpendicular direction to the hair at each centerline pixel are processed. An example of such gaps is shown in Figure 3.12(c). To overcome this problem, we implement a post-processing step after reconstruction, checking for gaps and fill them. A background pixel is classified as a gap when it is surrounded by two hair pixels in the horizontal, vertical or diagonal direction. When we find a gap pixel, we relabel it as a hair pixel.

Figure 3.12 shows the reconstruction process and the effect of the gap-filling process.

### 3.4.2 Length filtering

After reconstructing the hair, length filtering is applied once more on the hair mask to eliminate short segments. This step uses the same algorithm as the length filtering step in phase one (section 3.3.4)

Figure 3.13 shows an example of the effect of the final length filtering step.

### 3.4.3 Disocclusion

Disocclusion is the process of recovering the pixels originally occluded by hair. As this thesis focuses on segmentation rather than disocclusion, we use a simple algorithm to recover the pixels. We adopt the method used in [1], which is linear interpolation of the neighbouring non-hair pixels to repaint the hair pixels.

Figure 3.14 shows the final output of the algorithm. We superimpose the hair mask on top of the original image for easy visualization (Figure 3.14(a)). The recovered image after

\[^4\]The process stops when the convergence criteria are reached, or the number of iterations exceeds a set limit [18]
Figure 3.12: Hair pixels reconstruction: (a) The skeleton hair mask, (b) The reconstructed hair mask before post-processing, (c) Example of a region with gaps, (d) The region in (c) after gap filling.

Figure 3.13: Final length filtering step: (a) The hair mask before length filtering, (b) The hair mask after length filtering.
disocclusion is also shown (Figure 3.14(b)). It should be noted that the area at the edge of the input image is not used in assessing the result due to border effect. The computable area is outlined by the inner rectangle in Figure 3.14.

3.5 Method analysis

We evaluate the algorithm by doing quantitative analysis first on images with phantom hair. Then we use qualitative evaluation on real dermoscopic image results.

3.5.1 Quantitative evaluation

The use of phantom hair enables us to have a pixel-precise hair mask to use as ground truth in quantitative evaluation. To evaluate the effectiveness of an algorithm and find the optimal parameters, we carry out initial experiments on synthetic images created by implanting phantom hair on top of dermoscopic images with no hair. The segmentation result is compared with the ground truth to calculate statistics such as true positive, false positive, true negative and false negative. From these values we compute statistical measures of the performance of the algorithm: specificity, sensitivity, accuracy and diagnostic accuracy.

Specificity, sensitivity and accuracy are well known statistical measures of the performance of a binary classification system. They are defined using the following formulas:

\[
\text{Specificity(\%) = } \frac{TN}{TN + FP} \times 100
\]
\[ \text{Sensitivity(\%)} = \frac{TP}{TP + FN} \times 100 \]  
\[ \text{Accuracy(\%)} = \frac{TP + TN}{TP + FP + TN + FN} \times 100 \]

where TP, TN, FP, FN stand for the number of True Positives, True Negatives, False Positives and False Negatives respectively.

We choose diagnostic accuracy as the deciding factor for efficiency. Diagnostic accuracy is often used in evaluating the accuracy in the clinical diagnosis of malignant melanoma [19]. It is calculated as follows:

\[ \text{Diagnostic Accuracy(\%)} = \frac{TP}{TP + FP + FN} \times 100 \]  

The advantage of diagnostic accuracy is that it is not affected by true negatives. The majority of pixels in a dermoscopic images are skin pixels; hair is often only a small part. Hence, a large number of true negatives (skin classified as skin) often dominate the segmentation result. Any metric that uses true negative in its computation is biased toward this true negative rate. Diagnostic accuracy thus suits our purpose of evaluating dermoscopic image segmentation at the pixel level.

### 3.5.2 Qualitative evaluation

It is not feasible to have pixel-precise ground truth for real dermoscopic images. We use visual inspection to evaluate the algorithm on dermoscopic images. A segmentation result is classified as “good” if it contains most of all the hairs in the computable area.

### 3.5.3 Experiments

#### Optimal angular resolution experiment

In section 3.3.1, we explained why a series of kernels aligned at different angles, rather than a single kernel, is needed in matched filtering. The original paper by Chaudhuri [8] uses an angular resolution of 15°; that is, they rotate the kernel 12 times at 15° intervals to cover all possible curvature of blood vessels in retinal fundus images. In order to determine a suitable rotation angle for matched filtering in dermoscopic images, we designed an experiment where we tried different angular resolutions for the kernel and examined the extent of the
improvement resulted from higher resolutions. It is expected that the smaller the rotation angle is, the better the result will be. But smaller angles also mean slower computation time, as the base kernel is rotated more times and the computing extensive process of normalized cross-correlation is done for a larger number of kernels in the series. We evaluated the diagnostic accuracy of results from different rotation angles to determine a sufficient angular resolution. A “sufficient” rotation angle should provide a noticeable improvement in detection rate compared to greater angles and not introduce too much computation overhead.

Angular resolution was not the only parameter in this algorithm. Threshold value was the other controllable factor which affected the segmentation result. For the purpose of determining the optimal rotation angle, we used hard thresholding with a set of values covering the whole spectrum of gray levels in the filter response. The idea was to make the result independent of the thresholding method, so we used hard thresholding with 255 threshold values covering the whole response spectrum.

Our optimal rotation angle experiment was a program that executed the following steps:

1. Create phantom hair on a uniform background, as described in section 3.1.

   For this experiment we used two circular hairs of different diameters, implanted on a background image of $728 \times 480$ pixels. The background has a uniform intensity of 153, and the intensity of hair varies from 113 to 153 (i.e. the depth of the Gaussian cross section profile is $40^5$). Our goal was to cover all directions and different piecewise linear length of hair. While circular hair is not commonly seen in dermoscopic images, it covers all curvature and hence suits our purpose. The difference in size of the two circular hairs aimed at testing how strict the kernel was with respects to the length of the piecewise linear segments in hair. Figure 3.15 shows the phantom image used in this experiment. A mask of the phantom hair was also produced in this step. It would be used as the ground truth to evaluate the segmentation.

2. Run matched filtering on the phantom image.

   In each iteration, we rotated the kernel by $\alpha$ degrees until we covered the whole 180 degrees. We tried different values for $\alpha$: 5, 10, 15, 20, 30, 45, 60, 90 degrees.

3. Threshold the filter response for each angular resolution using hard thresholding.

---

5This number is based on the average intensity depth of hair in our set of real dermoscopic images
There were 255 threshold values, as stated earlier; hence for each resolution, the response was thresholded 255 times. A binary hair mask was produced for each rotation angle and each threshold value in this step. We also computed the local entropy for each angular resolution for comparison purpose.

4. Evaluate the hair mask against the ground truth.

For each mask of the total 2040 segmentation results (8 angular resolutions and 255 threshold values), we computed the accuracy, sensitivity, specificity and diagnostic accuracy of the segmentation. The computation time was also recorded and used as a factor in selecting the optimal rotation angle.

![Phantom hair image used in optimal angular resolution experiment](image)

Figure 3.15: Phantom hair image used in optimal angular resolution experiment

Diagnostic accuracy was used as the main factor in judging the results. In order for this experiment to be independent of thresholding methods, for each angular resolution, we used hard thresholding with threshold values in the [1,255] range\(^6\), and we only kept the result from the best hard threshold. The procedure was to go through the results of each angular resolution, discarding all of them but the one with the best diagnostic accuracy. This means that the experiment possibly used different thresholds for different rotation angle, but it ensured that the best thresholding value was used for each one.

\(^6\)The filter response is in the range [0,1], but in this experiment, we scale it to [0,255] for easy comparison between the optimal threshold and the local entropy threshold.
Table 3.1 shows the diagnostic accuracy and computation time of each angular resolution, after selecting the one with the best thresholding value. Figure 3.16 shows a graph of diagnostic accuracy as a function of angular resolution. We can see from the table and the graph that rotation angles of 5 and 10 degrees produce results with the highest diagnostic accuracy (within a small margin of each other). However, segmentation with a 10° rotation angle takes much less time than that with a 5° rotation angle. Using the result diagnostic accuracy as the main deciding factor and computation time as the secondary one, we come to the conclusion that rotating kernels at 10° is best for matched filtering dermoscopic images with a two-dimensional Gaussian filter.

<table>
<thead>
<tr>
<th>Angular resolution (degrees)</th>
<th>Threshold at best accuracy ([1..255])</th>
<th>Local entropy threshold</th>
<th>Diagnostic accuracy (%)</th>
<th>Computation time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>133</td>
<td>6</td>
<td>84.0</td>
<td>12.16</td>
</tr>
<tr>
<td>10</td>
<td>131</td>
<td>29</td>
<td>82.9</td>
<td>6.17</td>
</tr>
<tr>
<td>15</td>
<td>126</td>
<td>4</td>
<td>79.3</td>
<td>4.31</td>
</tr>
<tr>
<td>20</td>
<td>122</td>
<td>12</td>
<td>74.6</td>
<td>3.33</td>
</tr>
<tr>
<td>30</td>
<td>112</td>
<td>14</td>
<td>66.7</td>
<td>2.50</td>
</tr>
<tr>
<td>45</td>
<td>99</td>
<td>12</td>
<td>50.6</td>
<td>1.93</td>
</tr>
<tr>
<td>60</td>
<td>97</td>
<td>6</td>
<td>36.7</td>
<td>1.65</td>
</tr>
<tr>
<td>90</td>
<td>65</td>
<td>7</td>
<td>30.2</td>
<td>1.33</td>
</tr>
</tbody>
</table>

Table 3.1: Diagnostic accuracy and computation time of different angular resolutions

**Application on synthetic images with phantom hair**

We designed a pilot experiment with a synthetic input image with phantom hair to quantitatively evaluate the algorithm. The input image was built by superimposing phantom hairs on a real dermoscopic image with a melanoma and no hair (size: 728 × 480). We used two circular phantom hairs, one darker and one lighter than the background. The hairs were 4 pixels wide and had perfect Gaussian profiles of different depths. The skin background was non-uniform with the presence of a nevus. We used the following set of parameters:

- Kernel size: \( \sigma = 2 \) (the assumed half-width of hair), \( L = 20 \) (the length of the piecewise linear segment)

- Angular resolution: \( \alpha = 10 \), based on the result of our optimal rotation angle experiment
CHAPTER 3. INITIAL ALGORITHM DESIGN

3.6 Discussion

The initial algorithm design produces satisfying results on the simple cases of thick, constant width hair in both dark and light colours. However, it fails to handle fine hairs and the complicated case of hairs with many intersections. Example of such cases are shown in Figure 3.18. It also misses some hair segments and wrongly classifies some pigmented skin lesions as hair in regions with noisy backgrounds (Figure 3.17(d)). Efficiency is also a problem, due to the computing extensive process of Gaussian curve fitting being carried out on a large number of pixels. This is partly caused by the local entropy thresholding step,
Figure 3.17: Pilot experiment: (a) The input image, (b) The ground truth, (c) The filter response, (d) The output hair mask superimposed on the input image
which returns too many false positives in the raw hair mask for verification.

Figure 3.18: Example of the drawbacks of the initial design

As observed in the optimal angular resolution experiment (Table 3.1) and the demonstrative example of a response from a real dermoscopic image (Figure 3.10(b)), local entropy thresholding does not return the optimal threshold for the response from our matched filter. Using it to threshold the filter response results in a very noisy mask and effectively feeds a great number of false positives as hair centerlines to the verification step. The principle of local entropy thresholding is to select the threshold that maximizes the total second order local entropy of the object and the background [11]. It does not have knowledge about the properties of our universal filter: it works on the premise that strong responses in the response image correspond to the objects in the original image pixel-to-pixel. This might work in the general case of matched filtering; however, with our modified filter, the non-zero response does not correspond to the object pixel-to-pixel: each matching object usually generates a response wider than its width. The presence of the two “ghost” peaks alongside every peak, as illustrated in Figure 3.5(f) and 3.5(g), complicates the thresholding process. Furthermore, the normalization in the cross correlating computation amplifies noise in the input image; hence strong responses in the response image are not always from the objects. This drawback is very pronounced in images with skin backgrounds. Example of such images, their filter response and the resulting hair mask computed by local entropy thresholding algorithm are shown in Figure 3.19.

To further demonstrate the unsuitability of local entropy thresholding for dermoscopic images, we downloaded the code from [11] (made available to the public on their website [20]) and ran it on our set of data. The results are shown in Figure 3.20. It should be noted
that this code uses the classical concept of matched filtering, hence the response is different from our method; and it is not expected to be able to pick up light hair, as the filter was designed for dark, thin structures (retinal blood vessels). Nonetheless, the local entropy thresholding is expected to be able to segment dark hair to a certain extent of accuracy. As seen on Figure 3.20(c), the resulting hair mask (focusing on the dark hair result only) is very noisy, due to noise in the skin background. We investigated the problem further by inspecting the result of their code on their data set (20 retinal images). On images with noise, their local entropy thresholding did not return a clean hair mask. Their algorithm relied heavily on length filtering in such cases, which worked in their favour as the object in
CHAPTER 3. INITIAL ALGORITHM DESIGN

consideration (the vascular tree) is one connected segment. It did not work well in our case, as we were only interested in segmenting hair on dermoscopic images with the presence of nevi and thus, pigment networks (ie. noise).

Figure 3.20: Results from Chanwimaluang and Fan’s [11] implementation: (a) Our phantom hair input, (b) The response from classical matched filtering, (c) Local entropy thresholding result, before length filtering, (d) A retinal fundus image, (e) The response from classical matched filtering, (e) Local entropy thresholding result, before length filtering

Another weak point of the initial design is that Gaussian curve fitting is done for each point on the hair centerlines. Curve fitting with a non-linear model such as Gaussian is an expensive operation; thus applying it on all possible hair pixels is not desirable, especially when the previous steps produce a large number of false positives to be verified. This, coupled with the ineffectiveness of local entropy thresholding for our filter response, affects the speed and accuracy of the method.
Chapter 4

Revised algorithm design

We revise our algorithm design to improve its efficiency and accuracy. The weakest points in the initial design are addressed and redesigned. In particular, the thresholding step is reviewed and changed to a novel tracing method. The verification phase is also re-implemented to use simpler tests in conjunction with Gaussian curve fitting. We also address the intersection problem in a post-processing step.

4.1 Method overview

The principle of the algorithm remains the same: we use a universal kernel to segment both dark hair and light hair without prejudice or prior knowledge of the hair type. The algorithm uses a hybrid model of filtering, verification and reconstruction. We divide the process into three phases: The first phase locates the centerlines of all the possible hairs, the second phase verifies the centerlines and reconstructs the hair, and the third phase is added as the post-processing step to handle intersections and clean up the hair mask. Figure 4.1 shows a diagram of all the steps and phases in the algorithm.

4.2 Phase 1: Identifying the hair centerlines

The idea of Phase 1 is to locate the centerlines of all the possible hairs. We do so by using our modified matched filtering method and processing the filter response to get a raw hair mask. This raw hair mask then undergoes morphological thinning to produce a skeletonized mask containing all the possible centerlines.
4.2.1 Matched filtering

We use the same matched filtering method as detailed in section 3.3.1. The main details are repeated in this section for easy reference.

We use a series of 18 kernels, each at an 10-degree offset of each other for filtering. The kernel is characterized by:

\[ K(x, y) = -e^{-\frac{x^2}{2\sigma^2}} \]  \hspace{1cm} (4.1)

for \(|y| < L/2\), with \(L\) being the length of the piecewise linear hair segment.

The filtering process computes the normalized cross correlation of the kernel and the input image at each angle and takes the maximum out of all angles as the filter response. The filter response of an input image \(I\) of size \((P, Q)\) with a kernel \(K_i\) of size \((M, N)\) is:

\[ R(u, v) = \max_i \left| \frac{\sum_{x,y} [I(x, y) - \bar{I}_{u,v}] [K_i(x - u, y - v) - \bar{K}_i]}{\left( \sum_{x,y} [I(x, y) - \bar{I}_{u,v}]^2 \sum_{x,y} [K_i(x - u, y - v) - \bar{K}_i]^2 \right)^{0.5}} \right| \]  \hspace{1cm} (4.2)

where:

- \(K_i\) is a kernel in the series of rotated kernels \((i = 1, 2, ..., 18)\)
- \(\bar{K}_i\) is the mean of \(K_i\)
- \(\bar{I}_{u,v}\) is the mean of \(I(x, y)\) in the region under the template

Figure 4.2 shows an input image (with phantom hair on real skin background) and its filter response. We will use this phantom image to demonstrate all the steps in the algorithm. Information on this synthetic input image and its results can be found in section 4.5.3.

We also save the angle at which the kernel produces the highest response at each pixel for future use.
4.2.2 Processing the filter response

The revised algorithm differs majorly from the initial design in the way it processes the filter response to get the raw hair mask. Observing that local entropy thresholding is not suitable for our filter response (section 3.6), and that our filter response is very noisy due to normalization in the cross correlation process, we decide a different approach to process the response. We exploit the facts that hair usually has stronger response than noise, and being piece-wise linear, it varies little in curvature compared to artifacts such as pigment networks. As a result, response from hair is distinguishable from response from artifacts by the fact that the directional variation within the hair’s response is lower than in the artifact’s response. Also, we have information about the direction of each hair at each pixel from the angle of the maximum response we saved in the filtering step. These properties form the basic of our algorithm to process the filter response, detailed below.

**Algorithm 1** Response processing algorithm

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:</td>
<td>repeat</td>
</tr>
<tr>
<td>2:</td>
<td>$R_t \leftarrow$ intensity of the brightest point in the response image</td>
</tr>
<tr>
<td>3:</td>
<td>$R_b \leftarrow R_t \times e^{-\frac{1}{2}}$</td>
</tr>
<tr>
<td>4:</td>
<td>$H \leftarrow$ all pixels whose response $r$ satisfies $R_b \leq r &lt; R_t$ and connected to $R_t$</td>
</tr>
<tr>
<td>5:</td>
<td>if ($H$ satisfies curvature and length conditions) then</td>
</tr>
<tr>
<td>6:</td>
<td>add $H$ to the hair mask</td>
</tr>
<tr>
<td>7:</td>
<td>reset all pixels in $H$ to zero in the response image</td>
</tr>
<tr>
<td>8:</td>
<td>end if</td>
</tr>
<tr>
<td>9:</td>
<td>until $R_t &lt; 0.5$</td>
</tr>
</tbody>
</table>

The main idea of the process is to pick up the brightest point in the response image
(R_t), then find all pixels that are connected to it and have response fall within the range [R_t, R_b]. R_t and R_b are calculated based on the fact that response from hair, which has an approximately Gaussian intensity profile, is also has a Gaussian shape. Figure 4.3 visually explains the computation on line 3 of the algorithm: The brightest point (R_t) is assumed to be the top of a Gaussian curve, which is modelled by the following equation:

\[ R = d e^{-\frac{x^2}{2\sigma^2}} \]  

(4.3)

This curve reaches its global maximum at \( x = 0 \), so:

\[ R_t = R_{x=0} = d e^{-\frac{0}{2\sigma}} = d \]  

(4.4)

We want to take all the response values that fall within \( \sigma \) distance from the peak (R_t), so R_b is calculated as the response at \( x = \sigma \) on the Gaussian curve in 4.3.

\[ R_t = R_{x=\sigma} = d e^{-\frac{\sigma^2}{2\sigma^2}} = R_t e^{-\frac{\sigma}{2}} \]  

(4.5)

We find connected pixels using the same label propagation algorithm outlined in [17], which has been explained in section 3.3.4.

Each connected region is tested against two criteria:

- Length: Each piece of “hair” has to have length greater than a minimum length threshold.

- Curvature: We calculate the directional variance within each piece of “hair” using the angle information we saved in the filtering step. This information contains the direction of the maximum response at each pixel. For each pixel \((u, v)\) in the connected
4.4. Length and curvature tests on the filter response: (a) All connected pieces subject to testing, colour coded, (b) The raw hair mask containing all the pieces that pass the two tests.

Given that normalized cross correlation gives us the correlation coefficient at each pixel in the range $[0, 1]$, all points whose response is above 0.5 are the “better” matches to the template. This hard threshold is low enough to result in a raw hair mask which favours false positives over false negatives.

The tracing process stops when the brightest point in the response image drops to 0.5.

The curvature test computes $C_H$ for each connected region and compares it against a curvature threshold to determine whether the connected region is a hair segment. The curvature threshold should be higher than the angular resolution, since the angular resolution (ie. rotation angle) is the minimum directional difference between any two pixels.

The output of this step is a raw hair mask containing all the possible hairs.
4.2.3 Morphological thinning

We use the same method to skeletonize the hair mask as in the initial design (section 3.3.5). The output of this step is a mask of the centerlines of all the possible hair.

4.3 Phase 2: Verifying and reconstructing the hairs

Our filtering and response processing steps produce a raw hair mask with all the possible hair. The false positives in the raw hair mask either correspond to the two “ghost” peaks alongside a hair, or response from noise (variation in skin colour, pigment networks). Phase 2 checks each centerline resulted from skeletonizing the raw hair mask to see if it is indeed a hair centerline, and reconstructs the hair accordingly. For each pixel on the centerline, we use two tests for verification: first we check the cross section of the response to see if it has the expected property (one true peak in between two “ghost” peaks), then we check on the input image to see if the cross section of the hair at this pixel has a Gaussian shape. The three-peak response test is sufficient to determine whether a centerline pixel is a hair pixel, so we only carry out the Gaussian profile test when the three-peak response test fails.

4.3.1 Three-peak response test

Figure 4.5 shows the intensity profiles across the response of a hair and the response of a non-hair artifact. We observe that hair corresponds to bright, thin structures in the response image that are usually accompanied by two bright structures in parallel. This property can be observed by the “halo” alongside each hair in the response image. Response from noise does not have this property. This difference is due to the fact that our filtering process only retains the maximum response at each pixel, and the design of our filter results in three peaks at each matched position. The “ghost” peaks from hair response are much stronger than the response from noise in its surrounding area, even after normalization; thus they remain in the response image. Noise does not create the same effect for two reasons: (i) its cross section profile is usually not an exact match in shape with the filter, and (ii) even if its cross section profile matches the filter and thus creates the “ghost” peaks, these peaks are too low to overpower the response from other noise.

Based on this property, we design a 3-peak response test to verify if a point on the skeletonized hair mask is indeed a point on the centerline of a hair. For each point on
Figure 4.5: Intensity profile of response from true positives and false positives: (a) Intensity profile of a hair (true positive) response, (b) Intensity profile of a noise (false positive) response

The skeleton, we take $2\sigma$ pixels on each side of it, in the direction perpendicular to the hair (recall that we know the direction of the hair at each pixel based on the angle of the maximum response). We then check these $4\sigma + 1$ pixels to see whether their responses have three peaks in the order and magnitude similar to the shape in Figure 4.6. This test is a simple pixel comparison so it is not expensive; nonetheless, it suits our purpose.

The test is implemented as follows:

- $P_0$ is taken as the response of the middle point of the $4\sigma + 1$ pixels (i.e., at position $2\sigma + 1$ in the pixel sequence)
**4.3.2 Gaussian profile test**

If a centerline pixel fails the three-peak response test, we carry out the Gaussian curve fitting test as an additional verification step. We keep the implementation details of this step the same as in the initial design (section 3.4.1).

**4.3.3 Hair reconstruction**

If a pixel on the skeletonized hair mask passes one of the two verification tests, we expand it to cover $2\sigma$ pixels (the assumed hair width) in the perpendicular direction to the hair length. This is the reconstruction step, which gives us a full-width hair mask. Figure 4.7(b) shows a reconstructed hair mask from a skeletonized mask.
4.4 Phase 3: Post-processing

We add a post-processing phase to handle the hair pixels missed in the previous phases due to rounding and hair intersection. The pixels added in this step are based on the raw segmentation result obtained in Phase 2. We also filter out the false positives by length filtering.

4.4.1 Region growing

A drawback of our filtering method is the inaccurate results at hair intersections. This is due to two reasons:

- At intersections, the stronger response of one hair overwrites that of the other. The “halo” effect of the hair response also introduces a discontinuity at intersections: the response from the more dominant hair has two “ghost” peaks, which makes it wider than the actual hair width and hence creates two gaps on both side of it for the other hair.

- In the response processing step (section 4.2.2), each connected region is checked for directional variance. At intersections of hairs with approximately the same width and intensity, the intersection is returned as one connected region by the tracing step. This connected region contains two or more linear segments in different directions, hence its calculated curvature is higher than the curvature threshold.

Figure 4.8 illustrates the discontinuity as intersections when the filter response is processed.

We solve this problem by a region growing step to bridge the gaps at intersections. We “grow” each hair from its end points, using the 8-connected neighbourhood with additional constraints on the direction and intensity of the neighbouring pixels. The direction constraint ensures that each hair only grow in two directions, that is the direction of the hair taken at the two end points. The intensity constraint restricts the process to growing toward pixels whose intensity is in the intensity range of the current hair.

The hair end points are found by taking each pixels on the skeletonized hair mask and considering the number of foreground-to-background transitions. End points are the points which traversing its eight neighbouring pixels in the clockwise direction yields only one foreground-to-background transition.
The direction constraint is implemented by creating a mask of two “tubes” at each end point (direction mask), in the direction of the hair at the end points, and with width being $2\sigma$ to allow for slight change in curvature. The “tube” is implemented as follows (see Figure 4.9 for visualization): For each end point $(x_0, y_0)$, $\vec{u}$ is the vector in the direction of the hair at that end point (characterized by the angle $\theta$, which is the angle of the maximum response), $\vec{n}$ is the normal vector of $\vec{u}$. For every point $(x, y)$ in the image, let $\vec{v}$ be the vector from the end point to that point: $\vec{v} = (x - x_0, y - y_0)$. The distance from $(x, y)$ to the hair direction $\vec{u}$ is $d = |\vec{v}| \cos \phi = \vec{v} \cdot \vec{n}$. The tube, which has width of $2\sigma$, is characterized by all points $(x, y)$ in the image that satisfy $\vec{v} \cdot \vec{n} \leq \sigma$. 

Figure 4.8: Discontinuity at intersections: (a) The response image, (b) The connected regions returned by the response processing step of the area in the red rectangle in (a), with different regions coded by different colours
CHAPTER 4. REVISED ALGORITHM DESIGN

Figure 4.10: Region growing demonstration: (a,b,c) The input image and the hair mask before and after region growing of a phantom image, (d,e,f) The input image and the hair mask before and after region growing of a real dermoscopic image

This direction mask is applied on the image before the region growing process starts to indicate which direction to grow.

The intensity constraint ensures that the process only grows to pixels whose intensity $I_x$ is in the range $[I_{min}, I_{max}]$, where $I_{min}$ and $I_{max}$ are the minimum and maximum intensity within the current hair, respectively.

Figure 4.10 shows two examples of region growing to bridge intersections: one with our demonstrative example (phantom hair on skin background) and one with a real dermoscopic image which contains many intersecting hairs.

It should be noted that each hair is grown at two ends in the direction given at the corresponding end points only. Hence, it is not possible for the region growing algorithm to follow a hair that changes direction outside its current end points.

4.4.2 Length filtering

We apply another round of length filtering on the final hair mask to eliminate the “hairs” whose length is lower than our length threshold. This cleans up the hair mask from most false positives remaining in the mask after the verification step, as hairs are usually thin,
long structures whereas noise is short, small structures. Details on our length filtering implementation can be found in section 3.3.4.

We also use the same disocclusion method as in the previous design (section 3.4.3).

4.5 Method analysis

4.5.1 Evaluation method

We used the same approach to evaluating our revised algorithm as in the previous chapter (section 3.5): we used a phantom hair image whose ground truth is known to quantitatively assess the accuracy of the method, then we applied the algorithm on real dermoscopic images for qualitative evaluation. Results from DullRazor were also used as a comparison tool.

4.5.2 Experiments

Algorithm parameters

We used the following set of parameters in the application of our algorithm on the phantom images as well as the images from Argenziano et al.’s *Interactive atlas of dermoscopy* CD [5] and Marghoob et al.’s *Atlas of dermoscopy* [13].

- **Kernel size**: $\sigma = 2$ (the assumed half-width of hair), $L = 20$ (the length of the piecewise linear segment)

  Out of the two parameters, the accuracy of the results is more sensitive to $\sigma$ than to $L$. The algorithm assumes that hair has fixed width of $2\sigma$, hence the result depends heavily on the value of $\sigma$. The value set here is based on the average width of hair in our data set. The value of $L$ does not greatly affect the accuracy of the algorithm (see section 3.3.1 for explanation).

- **Angular resolution**: $\alpha = 10$, based on the result of our optimal rotation angle experiment (section 3.5.3).

- **Minimum length**: The minimum length of hair used in length filtering is 50 pixels.

  This is set based on the average minimum length of hair in our data set. This parameter moderately affects the segmentation result, as it is used to quickly discard false
positives. Setting it too high will falsely eliminate some short hairs, whereas setting it too low will let noise passes as hair.

- Minimum angle variation within one hair segment (used in curvature test): 15 degrees.

We use 10 degrees as the angular resolution in the filtering step, so the directional variance within each piece of hair cannot be lower than this angular resolution. 15 degrees variance is a reasonable threshold for smoothly curved hairs. Higher directional variance thresholds are not desirable as they allow for sharp “turns” within a hair segment, which makes the algorithm wrongly detect noise and pigment networks as hair and thus, affects the accuracy of the algorithm.

It should be noted that the values of these parameters are dependent on the resolution of the input images. However, it is expected that each set of input images has a fixed resolution, as dermoscope settings are usually consistent.

### 4.5.3 Quantitative assessment: Application on images with phantom hairs

We designed a pilot experiment to evaluate the efficiency and accuracy of our algorithm. We used a phantom image in this experiment in order to have a pixel-precise ground truth. The input image was created by planting phantom hairs of different colours, width and curvature on a background taken from a real dermoscopic image with a nevus and no hair. The phantom hairs were of dark and light colours, with width varying from 3 to 6 pixels. They were placed at various positions on the background, including inside the mole area. There were hairs that were close and parallel to each other, as well as intersecting hair (dark crossing dark, light crossing light and dark crossing light). There was a long, circular hair to cover all possible curvatures. The phantom hairs were designed to cover common as well as challenging cases. The image and its segmentation result are shown in Figure 4.11.

We calculated statistics such as true positive, false positive, true negative, false negative and used them to compute accuracy measures. We also computed those for the result returned by DullRazor for comparison purpose.

<table>
<thead>
<tr>
<th></th>
<th>Sensitivity</th>
<th>Specificity</th>
<th>Accuracy</th>
<th>Diagnostic accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our result</td>
<td>80.1%</td>
<td>98.8%</td>
<td>97.9%</td>
<td>64.9%</td>
</tr>
<tr>
<td>DullRazor’s result</td>
<td>45.7%</td>
<td>99.6%</td>
<td>97.0%</td>
<td>42.6%</td>
</tr>
</tbody>
</table>

Table 4.1: Quantitative evaluation of the phantom hair experiment result
Figure 4.11: Application on an input images phantom hairs: (a) The input image, (b) The ground truth, (c) The filter response, (d) The computed hair mask, (e) Our hair mask superimposed on the input image, (f) DullRazor’s hairmask superimposed on the input image
4.5.4 Qualitative assessment: Application on real dermoscopic images

We selected 40 images from [5] and [13] to test our algorithm. We chose the images with hair (both dark and light colours) and without other artificial artifacts such as ink and oil. Only images with good contrast are selected. These are images where hair stands out from the background and distinguishable from other artifacts.

The algorithm performed well on images with sparse hair, hair of both light and dark colours, and hairs that vary little in width. In such cases, our method was able to produce very clean hair masks with small numbers of false negatives (see Figure 4.12). In the easy cases of thick, constant width and sparse hair, its results were comparable to DullRazor’s. It outperformed DullRazor in the presence of light hairs in the image, or with cases with fine hairs. Examples of such segmentation results (easy and moderately difficult cases) are shown in Figure 4.12. Note that the hairs close to the image margin were not used in assessing the accuracy of segmentation due to the border effect.

The revised method also overcomes the problems of fine hairs and complicated intersection encountered in the initial design. Figure 4.13 shows the improvement of the new method on difficult cases.

Our method was disadvantaged against the case of heavy occlusion, or with hairs that vary greatly in width in the same image. It also had difficulties segmenting hairs that merge into each other or run closely in parallel. Figure 4.14 shows examples of these difficult cases.

4.6 Discussion

The strength and main contribution of our method is its ability to detect both dark hair and light hair at the same time, without prior knowledge of the hair type. This has not been done for hair segmentation in the literature. Our algorithm also produces clean hair mask with few false negatives. We favour fewer false negatives over more true positives, since false positives usually lie in the nevus areas and may contain useful pathological information, thus disoccluding them is not beneficial in later diagnostic steps. While our method was designed based on the assumption that hair has fix width, it is fairly tolerant toward width variation (in the range of $[\sigma - 2, \sigma + 2]$). Experiments show that it is usually able to pick up more fine hairs than DullRazor.

Another advantage of our method is the information about the hair we discover in the filtering process. For each hair successfully segmented, we know its internal curvature as well
Figure 4.12: Example of segmentation results on easy and moderately difficult cases: first column: the input images, second column: our segmentation results, third column: DullRazor’s results.
as its direction at each pixel. Using this information, it is possible to distinguish different hairs, even in the case of intersecting hair, which cannot be done by a simple connected region count. A connected region approach would return two intersecting hairs as one, but we can process the intersection to see which part of it belongs to which hair, as we know the direction of the two hairs at each pixel and thus, count them as two separate hairs. This feature was not implemented in this thesis as it is not our focus; however, implementation is fairly simple. Such feature can be of interest for hair research.

In the case of dermoscopic images with a high density of hair, our algorithm performs poorly. However, we favour physical hair removal in such cases, since too much information is lost in such heavy disocclusion.
Figure 4.14: Example of failed segmentation results on difficult cases (first column: the input images, second column: the disoccluded images using our hair mask, third column: the disoccluded images using DullRazor’s hairmask)
Chapter 5

Conclusion

5.1 Conclusion

We have successfully developed a novel method to segment hair in dermoscopic images. Our method uses a hybrid model of matched filtering and other image processing techniques to amplify the hairs and distinguish them from other artifacts. Our main contribution is the modified matched filtering process, which is capable of amplifying both dark and light hairs in the images, and our response tracing algorithm which follows the hair response and verifies its curvature. Our response processing method differs from most existing methods in the literature, as it does not use a threshold approach to get a hair mask from the filter response. Experiments on real dermoscopic images show that our algorithm is capable of segmenting hairs of dark and light colour, varying little in width and with medium density.

5.2 Future work

There are a number of remaining problems that are outside the scope of this thesis:

- Large margin of width variation: at the moment our algorithm has a margin of $[\sigma - 2, \sigma + 2]$ in width tolerance. Real dermoscopic images sometimes have hairs that vary greatly in width (eg. Figure 4.14(g)), which are missed in our segmentation. A possible approach to solving this problem is to apply more than one kernel at each pixel, with the kernels varying in width, then take the maximum response out of all the kernels.

- Use in-painting in the disocclusion process. We did not use this approach in our
implementation as we wanted to focus on segmentation and to be able to compare our segmentation result to DullRazor, which uses linear interpolation for disocclusion. In-painting is known to be more accurate in dermoscopic image disocclusion ([14]), with a tradeoff in computation time. For general purpose of hair removal, in-painting is favoured over linear interpolation.
Bibliography


