AUTOMATED CURVED HAIR DETECTION AND REMOVAL IN SKIN IMAGES TO SUPPORT AUTOMATED MELANOMA DETECTION

by

Madison Kretzler

Submitted in partial fulfillment of the requirements

For the degree of Master of Science

Thesis Adviser: Dr. Marc Buchner

Department of Electrical Engineering

CASE WESTERN RESERVE UNIVERSITY

May 2013
CASE WESTERN RESERVE UNIVERSITY

SCHOOL OF GRADUATE STUDIES

We hereby approve the thesis/dissertation of

__________________________
Madison Kretzler

candidate for the ______ Master of Science ______ degree *

__________________________
Dr Marc Buchner

(chair of the committee)

__________________________
Dr Kenneth Loparo

__________________________
Dr Vira Chankong

__________________________

(date) ____March 27, 2013____

*We also certify that written approval has been obtained for any

proprietary material contained therein.
# Contents

List of Figures ...................................................................................................................... 6

List of Tables ..................................................................................................................... 10

Abstract ............................................................................................................................. 11

1 Introduction ................................................................................................................... 12

1.1 Thesis Structure ...................................................................................................... 14

2 Background and Literature Review ................................................................................ 16

2.1 Skin Cancer: ............................................................................................................. 16

2.1.1 Cancer Statistics: .............................................................................................. 16

2.1.2 Types of Skin Cancer: ....................................................................................... 18

2.2 Computer-Aided Methods of Diagnosing Melanomas: .......................................... 28

2.3 Hair Detection and Removal in Lesion Images ....................................................... 33

3 Radon Transform ........................................................................................................... 39

3.1 Slant Stacking of the Radon Transform .................................................................. 40

3.1.1 Defining the \((p, \tau)\) Radon Transform............................................................... 40

3.1.2 Discrete Slant Stacking....................................................................................... 46

3.2 Discrete Radon Transform of a Discrete Line ........................................................ 49
3.2.1 Nearest Neighbor Interpolation Radon Transform of Discrete Line .......... 49
3.2.2 Comparison of Nearest Neighbor and Linear Interpolation ..................... 53
3.3 The Normal Radon Transform ........................................................................ 57
  3.3.1 Defining the $(p, \theta)$ Radon Transform .................................................. 57
3.4 Radon Transform for Curve Detection ............................................................ 59
  3.4.1 The Generalized Radon Transform ............................................................. 60
  3.4.2 Image Point Mapping ................................................................................. 64
4 Generalized Quadratic Radon Transform ............................................................ 65
  4.1 Initial Hair Masking: The DullRazor® [28] Method ......................................... 65
  4.2 Generalized Quadratic Radon Transform ....................................................... 70
    4.2.1 Generalized Centered Quadratic ................................................................. 71
    4.2.2 Quadratic Radon Transform Implementation .............................................. 73
  4.3 Translation and Windowing ........................................................................... 77
  4.4 Multiple Peak Detection ................................................................................. 77
    4.4.1 Peak Detection Algorithm ......................................................................... 78
    4.4.2 Manual Threshold Selection for Peak Detection ........................................ 81
  4.5 Curve Removal ............................................................................................... 82
5 Results .............................................................................................................. 87
  5.1 Validation ...................................................................................................... 87
List of Figures

FIGURE 1: CELLULAR SKIN CANCER IMAGE. [21] This image shows the development and position of the three main occurring skin cancers: Basal Cell Carcinoma, Squamous Cell Carcinoma, and the Melanoma. ..........19

FIGURE 2: IMAGES OF BASAL CELL CARCINOMAS ON THE NOSE [7]. .............................................................................. 20

FIGURE 3: SQUAMOUS CELL CARCINOMA (SCC): elderly man with badly sun damaged skin has SCC on his face...........21

FIGURE 4: MALIGNANT MELANOMA - first signs of melanomas is often the change in size, shape, and/or color. This melanoma is showing both the color variation and irregular border warning signs. [21] ..................... 23

FIGURE 5: ABCDEs of detecting and diagnosing MALIGNANT MELANOMAS [7] ...................................................... 24

FIGURE 6: LEFT: A two dimensional function that only is non-zero in the point \((x, y) = (x^*, y^*)\). RIGHT: The corresponding Radon transform (slant stacking result). Only when the Radon domain parameters match the parameters of the line a non-zero result is found. [28] ......................................................... 44

FIGURE 7: LEFT: A two dimensional function that is only non-zero when on the line. RIGHT: The corresponding Radon transform (slant stacking result). When the Radon domain parameters match the parameters of the line, a peak is found positioned at the parameters of the line in the image. The finite terms in the parameter domain are here ignored for sake of clarity. [28]........................................................................ 46

FIGURE 8: IMAGE DOMAIN CONTAINING TWO LINES \((p_1 = 1, \tau_1 = 0)\) AND \((p_2 = 0.5, \tau_2 = -50)\) .....................50

FIGURE 9: CORRESPONDING DISCRETE RADON TRANSFORM DONE BY NEAREST NEIGHBOR SLANT STACKING....................... 50

FIGURE 10: DENSE SAMPLING, ZOOMED IN, NEAREST NEIGHBOR DISCRETE RADON TRANSFORM OF IMAGE IN FIGURE 8 ......... 51

FIGURE 11: SPARSE SAMPLING, ZOOMED OUT, NEAREST NEIGHBOR DISCRETE RADON TRANSFORM OF IMAGE IN FIGURE 8...... 51

FIGURE 12: IMAGE DOMAIN CONTAINING TWO LINES \((p_1 = 1, \tau_1 = 0)\) AND \((p_2 = 0.5, \tau_2 = -50)\) ..................... 54

FIGURE 13: CORRESPONDING DISCRETE RADON TRANSFORM DONE BY LINEAR INTERPOLATION SLANT STACKING ................. 54

FIGURE 14: DENSE SAMPLING, ZOOMED IN, LINEAR INTERPOLATION DISCRETE RADON TRANSFORM OF IMAGE IN FIGURE 8.....55

FIGURE 15: DENSE SAMPLING, ZOOMED IN, NEAREST NEIGHBOR DISCRETE RADON TRANSFORM OF IMAGE IN FIGURE 8 .......... 55

FIGURE 16: THE TWO PARAMETERS \(p \) AND \( \theta \) USED TO SPECIFY THE POSITION OF THE LINE ............................................ 58
FIGURE 17: STRUCTURE ELEMENTS FROM DULLRAZOR® FOR THE GENERALIZED CLOSING OPERATION. (A) HORIZONTAL STRUCTURE ELEMENT, (B) DIAGONAL STRUCTURE ELEMENT, AND (C) VERTICAL STRUCTURE ELEMENT [8] ...................66

FIGURE 18: GRAPH OF QUADRATIC WITH VARYING $\alpha > 1$ ...........................................................................................................72

FIGURE 19: GRAPH OF QUADRATIC WITH VARYING $\alpha < 1$ ...........................................................................................................72

FIGURE 20: QUADRATIC CURVE WITH $\alpha=1$ AND $\theta = 0$. ............................................................................................................90

FIGURE 21: ZOOMED OUT QUADRATIC RADON TRANSFORM OF IMAGE IN FIGURE 20 ..................................................................90

FIGURE 22: QUADRATIC CURVE WITH $\alpha=1$ AND $\theta = 0$. ............................................................................................................91

FIGURE 23: ZOOMED IN QUADRATIC RADON TRANSFORM OF IMAGE IN FIGURE 22 ..................................................................91

FIGURE 24: QUADRATIC CURVE WITH $\alpha=0.5$ AND $\theta = \pi/2$ .......................................................................................................92

FIGURE 25: ZOOMED IN QUADRATIC RADON TRANSFORM OF IMAGE IN ERROR! REFERENCE SOURCE NOT FOUND. ................92

FIGURE 26: QUADRATIC CURVE WITH $\alpha=0.1$ AND $\theta = \pi/4$ .......................................................................................................93

FIGURE 27: QUADRATIC RADON TRANSFORM OF IMAGE IN FIGURE 26 .............................................................................................93

FIGURE 28: ZOOMED IN QUADRATIC RADON TRANSFORM OF IMAGE IN FIGURE 26 ..................................................................94

FIGURE 29: MULTI-QUADRATIC CURVE IMAGE WITH $\alpha=0.1, 0.5$ AND $\theta = \pi/4, \pi/2$ .................................................................96

FIGURE 30: MULTI-QUADRATIC RADON TRANSFORM OF IMAGE IN FIGURE 29 ..................................................................................96

FIGURE 31: MULTI-QUADRATIC CURVE IMAGE WITH $\alpha=0.1, 1, 0.5$ AND $\theta = \pi/4, \pi/2$ .................................................................97

FIGURE 32: MULTI-QUADRATIC RADON TRANSFORM OF IMAGE IN FIGURE 31 ..................................................................................97

FIGURE 33: MULTI-QUADRATIC CURVE IMAGE WITH $\alpha=0.1, 0.5, 0.1, 1$ AND $\theta = 3*\pi/4, \pi/4, 5*\pi/4, 7*\pi/4$ ...............98

FIGURE 34: MULTI-QUADRATIC RADON TRANSFORM OF IMAGE IN FIGURE 33 ..................................................................................98

FIGURE 35: MULTI-QUADRATIC CURVE IMAGE WITH $\alpha=0.001, 0.025, 0.05, 0.075$ AND $\theta = 0$ ...........................................99

FIGURE 36: MULTI-QUADRATIC RADON TRANSFORM OF IMAGE IN FIGURE 35 ..................................................................................99

FIGURE 37: ORIGINAL SINGLE HAIR SKIN IMAGE, .GIF IMAGE .................................................................................................101

FIGURE 38: HAIR SELECTION IMAGE FOR DESIGNATED RADON TRANSFORM IMAGE IN FIGURE 39 AND FIGURE 40 .........101

FIGURE 39: QUADRATIC RADON TRANSFORM OF FIGURE 38 ...........................................................................................................102

FIGURE 40: PEAK DETECTION, THRESHOLD OF 600, QUADRATIC RADON TRANSFORM OF IMAGE IN FIGURE 39 ..................102

FIGURE 41: ORIGINAL IMAGE AS DISPLAYED IN MATLAB, IN BLACK AND WHITE BECAUSE IT IS A Gif IMAGE (SINGLE LAYER) ..102
FIGURE 42: HAIR REMOVAL RESULTS OF ORIGINAL IMAGE AFTER FULL THESIS CODE IS PERFORMED ON IMAGE IN FIGURE 41, USES INTERPOLATION VERSION 2 ................................................. 102

FIGURE 43: SYNTHETIC HAIR IMAGE WITH TWO HAIRS ................................................................. 105

FIGURE 44: INITIAL HAIR MASKING OF IMAGE IN FIGURE 43 ........................................................ 105

FIGURE 45: WINDOWED IMAGE CONTAINING HAIR TO BE QUADRATIC RADON TRANSFORMED .......................................................... 105

FIGURE 46: QUADRATIC RADON TRANSFORM OF IMAGE IN FIGURE 43 ........................................ 105

FIGURE 47: PEAK DETECTION, THRESHOLD OF 600, QUADRATIC RADON TRANSFORM OF IMAGE IN FIGURE 45 ......................... 105

FIGURE 48: HAIR REMOVAL RESULTS OF ORIGINAL IMAGE AFTER FULL THESIS CODE IS PERFORMED ON IMAGE IN FIGURE 43, USES INTERPOLATION VERSION 2 ................................................. 105

FIGURE 49: SYNTHETIC HAIR IMAGE WITH THREE HAIRS, ONE ROTATED ........................................ 107

FIGURE 50: INITIAL HAIR MASKING OF IMAGE IN FIGURE 49 ........................................................ 107

FIGURE 51: WINDOWED IMAGE CONTAINING HAIR TO BE QUADRATIC RADON TRANSFORMED ........................................................................................................................................... 107

FIGURE 52: QUADRATIC RADON TRANSFORM OF IMAGE IN FIGURE 49 ........................................ 107

FIGURE 53: PEAK DETECTION, THRESHOLD OF 600, QUADRATIC RADON TRANSFORM OF IMAGE IN FIGURE 51 .......................... 107

FIGURE 54: HAIR REMOVAL RESULTS OF ORIGINAL IMAGE AFTER FULL THESIS CODE IS PERFORMED ON IMAGE IN FIGURE 43, USES INTERPOLATION VERSION 2 ................................................. 107

FIGURE 55: SYNTHETIC HAIR IMAGE WITH MULTIPLE ROTATED HAIRS ........................................ 109

FIGURE 56: INITIAL HAIR MASKING OF IMAGE IN FIGURE 55 ........................................................ 109

FIGURE 57: WINDOWED IMAGE CONTAINING HAIR TO BE QUADRATIC RADON TRANSFORMED ........................................................................................................................................... 109

FIGURE 58: QUADRATIC RADON TRANSFORM OF IMAGE IN FIGURE 55 ........................................ 109

FIGURE 59: PEAK DETECTION, THRESHOLD OF 1000, QUADRATIC RADON TRANSFORM OF IMAGE IN FIGURE 55 .......................... 109

FIGURE 60: HAIR REMOVAL RESULTS OF ORIGINAL IMAGE AFTER FULL THESIS CODE IS PERFORMED ON IMAGE IN FIGURE 55, USES INTERPOLATION VERSION 2 ................................................. 109

FIGURE 61: NON-LESION HAIR IMAGE ................................................................................................. 111

FIGURE 62: NON-LESION HAIR IMAGE HAIR MASK ............................................................................. 111

FIGURE 63: WINDOWED IMAGE CONTAINING HAIR TO BE QUADRATIC RADON TRANSFORMED ........................................................................................................................................... 111

FIGURE 64: QUADRATIC RADON TRANSFORM OF WINDOWED IMAGE ............................................. 111
FIGURE 65: PEAK DETECTED QUADRATIC RADON TRANSFORM OF WINDOWED IMAGE, THRESHOLD OF 800

FIGURE 66: FINAL INTERPOLATED IMAGE OF FIGURE 61

FIGURE 67: SECTION 1 OF NON-LESION HAIR IMAGE

FIGURE 68: INITIAL HAIR MASK OF SECTION 1 OF NON-LESION HAIR IMAGE

FIGURE 69: WINDOWED IMAGE CONTAINING HAIR TO BE QUADRATIC RADON TRANSFORMED

FIGURE 70: QUADRATIC RADON TRANSFORM OF WINDOWED IMAGE

FIGURE 71: PEAK DETECTED QUADRATIC RADON TRANSFORM OF WINDOWED IMAGE, THRESHOLD OF 800

FIGURE 72: FINAL INTERPOLATED IMAGE OF FIGURE 67

FIGURE 73: SECTION 2 OF NON-LESION HAIR IMAGE

FIGURE 74: INITIAL HAIR MASK OF SECTION 2 OF NON-LESION HAIR IMAGE

FIGURE 75: WINDOWED IMAGE CONTAINING HAIR TO BE QUADRATIC RADON TRANSFORMED

FIGURE 76: QUADRATIC RADON TRANSFORM OF WINDOWED IMAGE

FIGURE 77: PEAK DETECTED QUADRATIC RADON TRANSFORM OF WINDOWED IMAGE, THRESHOLD OF 800

FIGURE 78: FINAL INTERPOLATED IMAGE OF FIGURE 73

FIGURE 79: SECTION 1 OF NON-LESION HAIR IMAGE

FIGURE 80: INITIAL HAIR MASK OF SECTION 1 OF NON-LESION HAIR IMAGE

FIGURE 81: IMAGE DILATED HAIR MASK OF FIGURE 80

FIGURE 82: FINAL IMAGE PRODUCED THROUGH IMAGE DILATION AND INTERPOLATION VERSION 2

FIGURE 83: SKIN LESION HAIR IMAGE

FIGURE 84: SKIN LESION HAIR IMAGE INITIAL HAIR MASK

FIGURE 85: WINDOWED IMAGE CONTAINING HAIR TO BE QUADRATIC RADON TRANSFORMED

FIGURE 86: QUADRATIC RADON TRANSFORM OF WINDOWED IMAGE

FIGURE 87: PEAK DETECTED QUADRATIC RADON TRANSFORM OF WINDOWED IMAGE, THRESHOLD OF 1500

FIGURE 88: FINAL INTERPOLATED IMAGE OF FIGURE 83
List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE 1</td>
<td>Set sampling parameters of image and Radon domains for Figure 8 and Figure 9</td>
<td>50</td>
</tr>
<tr>
<td>TABLE 2</td>
<td>Settings of sampling parameters for zoomed in/out nearest neighbor Radon transforms</td>
<td>51</td>
</tr>
<tr>
<td>TABLE 3</td>
<td>Discrete Slant Stacking Nearest Neighbor Radon Transform Code</td>
<td>53</td>
</tr>
<tr>
<td>TABLE 4</td>
<td>Set sampling parameters of image and Radon domains for Figure 12 and Figure 13</td>
<td>54</td>
</tr>
<tr>
<td>TABLE 5</td>
<td>Settings of sampling parameters for zoomed in nearest neighbor and linear interpolation Radon transforms</td>
<td>55</td>
</tr>
<tr>
<td>TABLE 6</td>
<td>Discrete Slant Stacking Linear Interpolation Radon Transform Code</td>
<td>57</td>
</tr>
<tr>
<td>TABLE 7</td>
<td>Reproduction code of DullRazor®’s description of hair masking code [28]</td>
<td>70</td>
</tr>
<tr>
<td>TABLE 8</td>
<td>Generalized Quadratic Radon Transform Code</td>
<td>77</td>
</tr>
<tr>
<td>TABLE 9</td>
<td>FastPeakFind [29] code used for peak detection of quadratic Radon transform</td>
<td>81</td>
</tr>
<tr>
<td>TABLE 10</td>
<td>Adapted interpolation method code from DullRazor® description</td>
<td>84</td>
</tr>
<tr>
<td>TABLE 11</td>
<td>Interpolation Code Version 2 for replacement of image pixels</td>
<td>86</td>
</tr>
<tr>
<td>TABLE 12</td>
<td>Code for generating rotated quadratic images</td>
<td>88</td>
</tr>
<tr>
<td>TABLE 13</td>
<td>Parameters for Figure 20 and Figure 21</td>
<td>90</td>
</tr>
<tr>
<td>TABLE 14</td>
<td>Parameters for Figure 22 and Figure 23</td>
<td>91</td>
</tr>
<tr>
<td>TABLE 15</td>
<td>Parameters for images in Figure 24 and Figure 25</td>
<td>92</td>
</tr>
<tr>
<td>TABLE 16</td>
<td>Parameters of images in Figure 26 and Figure 27</td>
<td>93</td>
</tr>
<tr>
<td>TABLE 17</td>
<td>Parameters of images in Figure 26 and Figure 28</td>
<td>94</td>
</tr>
<tr>
<td>TABLE 18</td>
<td>Parameters for Figure 29 and Figure 30</td>
<td>96</td>
</tr>
<tr>
<td>TABLE 19</td>
<td>Parameters for images in Figure 31 and Figure 32</td>
<td>97</td>
</tr>
<tr>
<td>TABLE 20</td>
<td>Parameters for images in Figure 33 and Figure 34</td>
<td>98</td>
</tr>
<tr>
<td>TABLE 21</td>
<td>Parameters for images in Figure 35 and Figure 36</td>
<td>99</td>
</tr>
<tr>
<td>TABLE 22</td>
<td>Parameters used for image transforms based in actual images</td>
<td>101</td>
</tr>
</tbody>
</table>
Abstract

Automated Curved Hair Detection and Removal in Skin Images to Support Automated Melanoma Detection

MADISON KRETZLER

If detected early, skin cancer has a 95-100% successful treatment rate; therefore early detection is crucial and several computer-aided methods have been developed to assist dermatologists. In skin images removing hairs without altering the lesion is important to effectively apply detection algorithms. This thesis focuses on the use of image processing techniques to remove hairs by identifying hair pixels contained within a binary image mask using the Generalized Radon Transform. The Radon Transform was adapted to find quadratic curves characterized by rotational angle and scaling. The method detects curved hairs in the image mask for removal and replacement through pixel interpolation. Implementing this technique in MATLAB gives the ability to perform tests rapidly on both simulated and actual images. The quadratic Radon transform performs well in curve detection; however, the research points out the need for better algorithms to improve hair masking, peak detection, and interpolation replacement.
1 Introduction

Skin cancer is the most common form of cancer diagnosed in the United States each year, overshadowing those of breast, prostate, lung, and colon cancer combined [1]. There are currently more than 3.5 million recognized types of skin cancers [1], but three types account for close to 100 percent of all diagnosed skin cancer cases [2] [3]: basal cell carcinoma (BCC), squamous cell carcinoma (SCC), and malignant melanoma [4]. The most serious form of skin cancer is malignant melanoma. Despite the fact that this accounts for less than 5 percent of all diagnosed cases of skin cancer, it is the main cause of skin cancer related deaths [5] [1] [2].

Despite its high occurrence, if detected early skin cancer is also the most treatable form of cancer. Due to the importance of early detection and diagnosis many different visual as well as computational algorithms have been developed to aid dermatologists in early diagnosis of skin lesions. The most common of these algorithms include the Menzies method, the 7-point checklist, and the CASH algorithm; however, the most popular algorithm for diagnosing skin lesions is the ABCD method [4] [6]. The ABCD method describes the pointers of malignancy through Asymmetry, Border, Color, and diameter/differential structure. Not included in the initial acronym but added in in 2003 [7] is an E term referring to the Evolution of the lesion over time.

The combination of developments in these algorithms and developments in the field of image processing the popularity of using these algorithms as guidelines for image processing programs to aid in the diagnosis of malignancy in skin lesions has
grown. Since early detection of malignancy is so important, many software approaches focus on aiding dermatologists in early recognition. With the assistance of these programs dermatologists can increase their effectiveness at early diagnosis. Because of the usefulness of varying imaging techniques to clearly show pigmentation patterns in skin lesions, dermoscopy images have become an indispensable tool for such imaging programs [4].

Despite the growing amount of research in this area and concomitant advances in technology there remain considerable work to be done to create effective and reliable automated diagnostic methods. In many automation programs there is a lot of discussion and adaptation referring to the mapping, segmentation, and classification/diagnosis of skin lesions; however, not much research has focused has been done on the subject of hair removal. Human hair covers the entire body and has a range of different colors, textures, and orientations. As a result images and lesions can become occluded by these hairs and disrupt algorithms being used for lesion detection, therefore hair can cause major informational corruptions when working with a skin lesion image.

Two main papers expanded on in this thesis address this issue directly but approach the problem differently. The DullRazor® [8] and E-Shaver [9] papers elaborate on their methods and results and are both effective for the removal and replacement of hairs within images. Despite this success both programs have their limitations as well. In this thesis hair detection and removal is addressed through the use of the Radon
transform. A generalized quadratic Radon transform is developed for the purpose of
detecting a broader class of hairs for removal.

Though the Radon Transform is most commonly recognized for its use in
Computer Axial Tomography (CAT) scans [10], in this thesis an extended form of the
Radon transform is an essential part of curved hair detection. Because the Radon
transform works by taking line integrals through an object in various directions [11] the
transformation of the line into a quadratic was essential for the detection.

The Radon transform has the capability to map two-dimensional images
containing lines, or in this case curves, into a Radon transform domain of line/curve
parameters. An image containing straight lines can be mapped to a Radon domain of
line parameters where the lines in the image produce peaks in the Radon domain
positioned at the corresponding line parameters. In this thesis a quadratic Radon
transform was developed to detect and locate curvatures in an image. Once an initial
hair masking of the image was done to differentiate the hairs from lesion and skin the
quadratic Radon transform was used to classify whether a pixel is part of a hair or not.
Using a peak detector and an interpolation method the peaks in the Radon domain,
relating to the detection of a curve in the image domain, were used to attempt
“shaving” the hair from the image.

1.1 Thesis Structure

The remainder of the thesis is structured as follows:
Chapter 2 – “Background and Literature Review” provides background information on skin cancer, skin cancer statistics, and skin cancer varieties. It describes various algorithms used for diagnosing melamomas and different indicators and patterns of skin cancer. Different methods of automation of melanoma detection and diagnosis are reviewed as well as current hair detection and removal papers and processes.

Chapter 3 – “Radon Transform” provides background information on the continuous and discrete generalized linear Radon transform, defining properties and equations, as well as examples of its use and adaptations.

Chapter 4 – “Generalized Quadratic Radon Transform” details the actual transform used in the thesis research. Descriptions of hair image masking techniques, the development of the quadratic Radon transform, peak detection methods, and pixel interpolation techniques are included here.

Chapter 5 – “Results” includes all testing and validation processes

Chapter 6 – “Conclusions and Future Work” reviews the effectiveness and contributions of the work as well as provides a discussion of future work and expansions to be made in the process of hair removal and replacement.
2 Background and Literature Review

2.1 Skin Cancer:

2.1.1 Cancer Statistics:

Cancer is responsible for the top two leading causes of death in the developing world [12] [13]. In 2008, approximately 12.7 million cancer cases were recorded (excluding those of non-melanoma skin cancers), of which 7.6 million were fatal [12] [13]. According to the World Cancer Research Fund International website, this number is projected to increase to 21 million by 2030 [12] [4]. Despite these statistics there is a general consensus among physicians and scientists alike that a large portion of the world’s cancer deaths could be prevented through application of current cancer control knowledge as well as early detection [13]; however, this projection continues to steadily get worse.

Although no formal study has been conducted to investigate this steady increase in cancer, some general assumptions can be made. First, fashion over the past few decades has changed to favor greater levels of skin exposure. Combined with a trend of increased tanning outdoors and indoors (in tanning beds), leaves the skin vulnerable and more susceptible to ultraviolet (UV) rays, and as a result skin cancer. Coupling this increased skin exposure, mostly in women, with the depletion of the ozone layer [14] [15] since 1980, allows for more UV radiation reaching the human skin [14]. Because the majority of skin cancers are caused by extensive exposure to UV rays [5] there is
probably a correlation between the depletion of ozone and the increase in cases of skin cancer. Also people who first use tanning beds before the age of 35 have an increased risk for melanoma of 75 percent [1].

Skin Cancer is the most common form of cancer in the United States, accounting for nearly half of all cancer cases reported [16] [5] [17] [18] [19]. Each year new cases of skin cancer are greater than those of breast, prostate, lung, and colon cancer combined [1]. In 2008, 200 thousand new cases of Melanoma of the skin were diagnosed worldwide and accounted for 1.6% of all cancer cases [12]. Melanoma of the skin is the most serious form of skin cancer; even accounting for less than 5 percent of cases, it causes a vast majority of skin cancer related deaths [5] [1] [2]. At Least 75,000 cases of melanoma were reported in 2012. In addition to melanoma, 2 million cases of the less fatal basal and squamous cell skin cancers are diagnosed each year in the United States alone. The other accounts of skin cancer in the United States included the basal and squamous cell skin cancers. Although they are not as fatal, over 2 million cases of the basal and squamous cell skin cancers are diagnosed each year in the United States alone [5].

According to the American Academy of Dermatology, 1 in 5 Americans will develop melanoma skin cancer if current trends continue. Skin cancers are divided into two classes: non-melanoma and melanoma. [2] Melanoma skin cancer is the most fatal type of skin cancer and its occurrences are rising exponentially. Approximately 1 person dies of melanoma every hour in the United States [5] [1] [2] and according to
statistics by the Skin Cancer Foundation incidences increased by 800 percent in women and 400 percent in men between 1970 and 2009. [1] To give some perspective, in 1930 this ratio was 1 in 5,000 Americans. In 2004 the ratio was 1 in 65 [2]. Despite the common knowledge that sun protection can decrease the development of skin cancer, [7] today melanoma is the second most common cancer in women ages 20 to 29 [2].

Sun-exposure is the leading cause of skin cancer, due to the damaging effects of Ultra Violet rays, and therefore people with a long history are most susceptible. However, a family history of skin cancer accounts for between 5-10 percent of diagnosed melanomas [2]. Due to the fact that pigmentation helps protect the lower layers of the epidermis, skin cancer is more likely to develop in people of light skin, light-color eyes, fair hair, and skin with a tendency to burn or freckle. But this pigmentation only protects the skin from low levels of sun exposure. As a result a darker skinned person with low levels of sun exposure can still develop skin cancer; however, it is more likely to occur in non-sun-exposed areas [2].

2.1.2 Types of Skin Cancer:

Skin cancer is a tumor or abnormal growth of cells in the skin [3] that develops when DNA becomes damaged [2]. Since DNA is the molecule in a cell that encodes its genetic information, the body is unable to repair this damage. Due to the unrepaired damage this cell will continue to grow and divide uncontrollably. This uncontrollable multiplication of the damaged cells forms a tumor, which, due to its location, is usually
visible and easily detected. Because this tumor is forming in the skin, its visibility allows for early detection [2] [3].

Currently there more than 3.5 million recognized types of skin cancers [1], with three types accounting for close to 100 percent of all diagnosed cases [2] [3]. These three are: basal cell carcinoma (BCC), squamous cell carcinoma (SCC), and melanoma [4] [16] [5] [17] [18] [1] [2] [7] [3] [20] [6]. All other types of skin cancer account for less than 1 percent of the diagnosed cases and are classified as non-melanoma skin cancers. These types include Merkel cell carcinoma, dermatofibromasarcoma skin protuberans, Paget’s disease, and cutaneous T-cell lymphoma [5] [2]. The following image shows the three main types of skin cancers and their development within the epidermis:

![Cellular Skin Cancer Image](image_url)
2.1.2.1 Basal Cell Carcinoma (BCC):

The most common of these three main skin cancers is the basal cell carcinoma (BCC), [5] [2] [7] which accounts for about 80% of all skin cancer diagnoses [2]. The BCC can be seen in the center of Figure 1 where it occurs in basal cells. Basal cells are located at the base of the epidermis where they continuously replace the older skin cells above, displacing them towards the surface [5] [21]. BCC develops in over 1 million people every year in the United States alone. This type of cancer takes several forms although it is classified by the cell in which it developed [2] [21]. The BCC can appear as a flat, scaly red patch; small, smooth, shiny, or waxy bump (which may bleed or crust); large blood vessel patch or seeming birthmark; and/or a brown or black raised bump [5] [3]. Often this kind of tumor can become a sore that continuously heals and re-opens [2]. Example images of a BCC can be seen in Figure 2:

![Image of Basal Cell Carcinomas on the Nose](Image)

**Figure 2: Images of Basal Cell Carcinomas on the Nose [7]**

BBC’s normally appear on skin which is regularly exposed to the sun and UV rays, commonly the face (mainly nose), ears, scalp, neck, and back of hands [5] [7] [2]. They usually grow slowly and can possibly take years to reach a half an inch in diameter.
While it is unlikely for these tumors to spread to the rest of the body, early detection and treatment are still urged in order to prevent possible damage to surrounding tissue [2] [7].

2.1.2.2 Squamous Cell Carcinoma (SCC):

Developed in the upper layer of the epidermis, squamous cell carcinomas (SCC) [7] [21] account for nearly 16 percent diagnosed skin cancers [2] [21]. Every year approximately 700,000 Americans are diagnosed with SCC [7]. As the name suggests, this cancer develops in squamous cells that are located in the upper layer of the epidermis as can be seen on the left in Figure 1 [5] [21]. When new cells are moved upward through the epidermis they become flattened and are classified as squamous cells, and this is where SCC occurs. [21] SCC does not have a specific place of development and can occur on any part of the body including inside the mouth, although skin normally exposed to sun is the most common [5] [7] [2]. These carcinomas tend to develop in the elderly who have sustained long-term sun exposure and skin damage. Figure 3, gives an example image of a SCC.

Figure 3: Squamous Cell Carcinoma (SCC): elderly man with badly sun damaged skin has SCC on his face
As with the tumor in Figure 3, a SCC will often appear as a scaly or crusted layer of skin with an inflamed red base. The appearance may vary, and manifestations include: flat red scaly patches similar to a skin rash; small and smooth shiny/waxy bump that can bleed or develop a crust; and a red or brown colored scaly patch [2] [3]. SCC’s spread quickly and damage surrounding areas if not detected early. People who use tanning beds have a significantly increased chance of developing SCC early in life. Early detection of an SCC renders it highly curable [7].

2.1.2.3 Malignant Melanoma:

The most deadly type of skin cancer, melanoma, develops in the melanocytes. These cells are responsible for producing the pigment melanin that is responsible for coloration of the skin [5] [2] [21]. Melanocytes are located in the base of the epidermis [21]. The melanin produced in these cells protects the deeper layers of the skin from sun and UV exposure [5]. When detected early, melanomas have a 95-100 percent successful treatment [5] [2]; however, due to the rapid rate of spreading and its ability to invade the lymph system and internal organs they can be extremely deadly if not treated quickly. Once melanomas spread, their prognosis for successful treatment drops precipitously [2].

Melanomas can take many forms, but one image of a melanoma is presented in Figure 4:
Malignant melanomas are hard to recognize because they form in pre-existing moles or develop into impersonation moles. Many people do not keep track of these changes in their skin and scientists believe this is the reason for the high mortality rate associated with melanomas. Dermatologists insist it is important for people to know their skin and moles in order to detect changes and development that could become melanomas [2]. Melanomas can appear in the forms of a new mole, a mole that is getting bigger, a mole that changes color or shape, a mole that bleeds, a mole that itches or causes pain, or a mole with an uneven border or shape [2] [3].

2.1.2.3.1 ABCD Algorithm for Diagnosis Malignant Melanomas:

There are many different algorithms that are used for detecting and diagnosing skin cancer, the most common of which are the Menzies method, the 7-point checklist, and the CASH algorithm. The most commonly used algorithm for diagnosing malignant melanomas is the ABCD method. [4] [6]
Figure 5: ABCDEs of detecting and diagnosing Malignant Melanomas [7]

The ABCD’s method of detection was developed in 1985 to aid in self-diagnosis and monitoring due to the dangerous nature of malignant melanoma. [7] [21]. The ABCD, and recently added E in 2003 [7], method is described as follows:

A) **Asymmetry**: This is the first step in the ABCD method to determining a malignant melanoma. The asymmetry of the lesion is determined by the color, shape, and internal structure across the middle axis of the lesion, therefore if the lesion were to be folded in half it would not match itself.
Both halves could differ in color, texture, size, etc. all of which would classify it as being asymmetrical. Varied levels of asymmetry can be detected in melanomas but the greater the variation the more suspicious the lesion becomes. [7] [21]

B) **Border:** Irregular borders cast suspicion on a lesion. Sudden or abrupt breaks, drop offs, and notches in the margin of the lesion are strong indications of malignancy. Scalloped, uneven, or blurred borders show that there is a disruption in pigmentation and is a warning that the mole should be examined further. [7] [21]

C) **Color:** Strong variation in color across the lesion stands out as a strong red flag. Normally a total of 6 different colors are considered in a lesion: white, red, tan, brown, blue-gray, and black. The majority of lesions will be uniformly either tan or dark brown; therefore the most frequent indication of malignancy is the presence of 3 or more colors or shades of the same color in 85% of cases and 5 or 6 colors in 40% of all cases [6]. These color changes will often contribute to border discrepancies, another indicator for malignancy, and can be produced through multiple colors or an uneven distribution of the colors. [7] [21]
D) **Diameter and Differential Structures:** Two descriptors are often used for D:

i. **Diameter:** Cancerous melanomas generally have a diameter which is of or greater than 6 millimeters. This is approximately the size of a pencil eraser, as shown in Figure 5. Though this is a general rule of thumb as melanomas have been known to be smaller and size should not be taken as a requirement for malignancy. Commonly a mole or lesion that stands out from others but is smaller can still be a matter of concern. [7] [21]

ii. **Differential Structure:** Referring to the different types of structures within the lesion, differential structures can entail pigmented networks, structureless or homogeneous areas, streaks, dots and globules. The more structures that are present the increase in probability that the lesion is malignant [4] [6]. In more than 70% of malignant melanomas 4 or more differential structures were found [6]. [4] [7] [6] [21]

E) **Evolution:** Added to the list in 2004, evidence of evolution of a melanoma is a major indication of malignancy. This refers to a change over time in any one of the ABCD classifications of the lesion. Normally referring to an increase size over time, malignant melanomas can also develop from previous moles or lesions meaning that the change or evolution of a preexisting lesion can be cause for concern. Differences in color over time or
even the production of protruding skin can indicate that the melanoma is evolving into malignancy.

Although the ABCD diagnosis method does not cover all melanoma symptoms and appearances it is the most commonly used algorithm. Some melanomas can appear as color streaks under finger nails or toe nails and some appear as bruises which will not heal [7]. Despite these outliers, experts agree that this is a good approach to self-diagnosis and when used properly could help reduce the mortality rate of malignant melanomas. As shown in the next section, the ABCD method when used in conjunction with computer algorithms can increase the early detection of malignant melanomas and proving its usefulness as a whole.

2.1.2.3.2 Menzies Method of Detection:

In the Menzies method, two separate groups are defined – one consisting of negative features and one consisting of positive features. Negative features include are those with any presence of a singular color along with axial symmetry of pigmentation. Positive features include blue-white veil, multiple brown dots, pseudopods, radial streaming, scar-like depigmentation, peripheral black dots, multiple colors (5 to 6), multiple blue/grey dots, and a broadened network. When diagnosing melanomas negative features must be missing and at least one positive feature must be found. [6]

2.1.2.3.3 7-Point Checklist:

In the 7-point checklist three major criteria - each having a maximum score of 2 - are used. These criteria include atypical pigment network, blue-white veil, and atypical
vascular pattern. Along with these major criteria there are also four minor criteria each having a maximum score of 1. The minor criteria include irregular streaks, irregular pigmentation, irregular dots or globules, and regression structures. A minimum total score of 3 is required for the lesion to be diagnosed as melanoma. [6]

2.1.2.3.4 CASH Algorithm:

The CASH algorithm is a simplified version of pattern analysis aimed to be suitable for less experienced examiners. It incorporates a scoring system similar to that of the ABCD algorithm, which examines color (C), architecture (A), symmetry (S), and homogeneity (H) of the skin lesion. The accuracy of this algorithm is considered to be comparable to other melanoma distinguishing algorithms. [6]

2.2 Computer-Aided Methods of Diagnosing Melanomas:

Using the guidelines of the ABCD method in conjunction with properties of the Menzies method, 7-point checklist, and the CASH algorithm, some programs have been developed to help dermatologists diagnose malignancy in skin lesions. Because of their usefulness for clearly showing pigmentation patterns in skin lesions, dermoscopy images are an indispensable tool for imaging programs [4]. Frequently the use of different optical methods, multi spectral imaging [22], segmentation, classification, and pattern analysis are exploited with variables specified by the ABCD algorithm [6]. Computer aided methods of diagnosing melanomas have been produced to both work with or independently of clinicians, dermatologists, or dermoscopy experienced dermatologists. The breakdown of most of these programs and automated diagnosis
methods can be divided into three steps: (1) Determination of the tumor area from dermoscopy image; (2) extraction of image features; (3) Building of the classification model and evaluation [4].

According to *Computer-aided Dermoscopy for Diagnosis of Melanoma* [19] in *BioMed Central's BMC Dermatology*, the diagnostic accuracy of most computer-aided dermoscopy programs using neural networks in diagnosing pigmented skin lesions have been shown to perform at the level of a dermatologist trained in the use of dermoscopy, while out performing an inexperienced clinician. It was shown that when used in comparison with a simple clinical observation that these programs improved the diagnostic accuracy by 20-30% [19]. This increase appears to be the result of the program’s ability to evaluate noninvasively other morphological features that are not visible to the naked eye. Despite being the main method for diagnosing skin lesions the ABCD method’s accuracy for diagnosis is not very high. In this research 122 pigmented skin lesions were used to determine the sensitivity and specificity of computer-aided dermoscopy system for diagnosis of melanoma in Iranian patients. They compared the effectiveness of the microDERM® dermoscopy unit on each lesion with the results from examinations of two clinicians. After examinations were done with both methods, all lesions were histopathologically examined, meaning they were removed and studied though biopsy by a pathologist. Six (6) of the lesions were confirmed melanomas. When compared to the clinicians’ diagnoses melanomas were considered for 19 of the lesions, with 9 being labels as most likely melanomas. Doing comparisons considering only the most likely clinical diagnosis of melanomas with the dermoscopy score of the program
they reported no considerable difference in the sensitivity and specificity and therefore could not reduce the unnecessary removal or improve early detection. However, it was concluded that diagnostic accuracy could improve when being used with an inexperienced clinician. [19]

Research has also been aimed at automating melanoma detection through a multispectral imaging system. Automated Melanoma Detection with Novel Multispectral Imaging System [22], published in the Physics in Medicine and Biology, presented research relating to the performance of a spectroscopic system to perform automated melanoma detection. The system operates at wavelengths between 483 and 950nm and takes 15 images equally spaced between the wavelengths. The system was run using a neural network classifier for automated diagnosis on 1278 patients with 1391 cutaneous pigmented lesions, 184 of which were melanomas. The neural network was trained using 696 lesions of which 90 included melanomas and verified on 348 lesions containing 53 melanomas. The independent test set included 347 lesions with 41 melanomas. In doing this the neural network could distinguish between melanomas and non-melanomas with a sensitivity of 80.4% and specificity of 75.6%. The article concludes that their analysis shows that the computer automated diagnosis of pigmented skin lesions using multi spectral imaging has accuracy comparable to that of an expert clinician. [22]

Since early detection of melanomas is so important, many software approaches focus on aiding dermatologists in the early recognition of malignant melanomas. Just as
in [22], automated melanoma detection through multispectral imaging methods. Epiluminescence microscopy (ELM) is used as another way to perform in vivo (occurring within a living organism) examination and analysis [23]. This non-invasive technique works by using halogen light to take advantage of phenomenon that occurs in oil immersion. By exploiting this phenomenon structures in the subsurface of the skin are made visible for in vivo examinations, providing additional criteria for clinical diagnosis of the lesion [18] [23]. Having more information about the lesion undoubtedly could increase the accuracy of diagnosing a skin lesion. It has been concluded that the ELM techniques increase the sensitivity of formally trained dermatologists, therefore increasing their diagnostic ability. However, the researchers also concluded that if this method is used by a dermatologist who is not formally trained in the ELM techniques it can actually decrease their diagnosis capability. [23]

In the article in *IEEE Transactions on Medical Imaging* titled *Automated Melanoma Recognition* [18] this ELM technique was used in conjunction with automated code for segmentation, feature calculation, and feature selection and classification to perform automated recognition and diagnosis of malignant melanomas. Application of ELM imagery, along with previously described programing methods, correctly segmented 96% of the 4000 skin lesion images attempted. This process presented an extremely reliable segmentation for the skin lesion by using integrated ABCD and ELM criteria to develop features for their system. Currently this system is being used in large scale screening applications and comprehensive tests at the
Department of Dermatology at the Vienna General Hospital showing that systems like this one can aid and increase melanoma detection. [18]

Probably the most in-depth analysis of computer aided diagnosis of melanomas was done in Rahil Garnavi’s PhD thesis, Computer-aided Diagnosis of Melanoma [6]. The focus of the thesis was to develop a fast and accurate border detection method. Due to variations of the lesion’s color and the diversity of human skin colors, borders of a lesion can be extremely challenging for a program to distinguish in an image. In his dissertation Garnavi summarizes methods presented over the past two decades dealing with improving clinical diagnosis of melanoma and discussed the use of popular diagnostic algorithms which use detection methods like pattern analysis, ABCD rule of dermoscopy, Menzies method, 7-point checklist, and the CASH algorithms. Reportedly, using these methods along with dermoscopy can increase the accuracy of diagnosing melanomas by 5-30% as compared to the naked eye. This thesis also proposes a comprehensive and highly effective method for feature extraction using methods from various other sources. In order to accomplish this, current methods were examined for the computer-based diagnosis of melanoma. Some of the methods studied include: SolarScan, developed by Polartechnics Ltd., Sydney, Australia; DermoGenius-Ultra, developed by LINOS Photonics Inc.; and DBDermo-MIPS system, developed by University of Siena, Italy. [6] In conclusion of his work Rahil Garnavi provided main contributions to color-channel optimization for border-detection in dermoscopy images, automatic border detection in dermoscopy images (global and hybrid methods), objective evaluation of border-detection method in dermoscopy images, feature
extraction in dermoscopy images using wavelet, geometry and time-series analysis, and selection and optimal integration of features and classification of dermoscopy images.

[6]

Many different papers have been written on dermoscopic image processing and the field is constantly growing and improving. Increasing accuracy of diagnosing skin cancer in its earliest stages is the ultimate goal, whether these efforts combine with those of dermatologists or automatically generate diagnoses. These papers are only a representative sample of the work being done in the field of automated and computer aided recognition and diagnosis of malignant skin lesions, there are hundreds of papers relating to the subject. Despite the growing field and advances in technology there remains considerable work to be done to create effective and reliable automated diagnostic methods. An example of a problem that needs to be further addressed and improved is the process of hair removal in lesion images which occlude portions of images. This is problem and current papers addressing it are explored and discussed in the next section

2.3 Hair Detection and Removal in Lesion Images

Because dermoscopic images can show pigmentation patterns of lesions they are indispensable tools for image programs to diagnose skin cancer. In the previous papers reviewed, all refer to the mapping, segmentation, and classification/diagnosis of skin lesions as malignant melanomas; however, none address an important and common problem that can disrupt the algorithms and produce major errors in results. Human
hair covers the entire body and has a range of different colors, textures, and orientations, images of skin lesions always have the possibility of being obscured by hair [8]. Due to this, hair can cause major information corruption when working with a skin lesion image. Despite rapid growth of image processing for dermatological applications, this particular issue had never been fully addressed until the publication of DullRazor® in 1997 [8].

Before DullRazor® [8] a few solution existed that could be used to remedy this problem. The simplest of these solutions would be to remove all images obscured by or containing hair from the data set. This solution obviously would reduce the universal usage of the algorithm by eliminating a majority of usable images. Another commonly mentioned solution is to simply shave the hair physically off the skin. As mentioned before shaving can be a dangerous and painful process that can cause bleeding, skin or tissue damage, and even compromise the features of the lesion causing errors in detection and diagnosis. The slowest solution, which is also the most time consuming and costly, would be to manually go through each image and detect each border (hair and lesion). Using a low-pass filter to average the visual scene is also an option. This would remove thin narrow hairs from the image; however, it will leave thick hairs behind. This method could also result in over averaging, which would result in the loss of critical information regarding the borders and structure of the lesion. However, too little averaging would leave thick, obscuring hairs in place, preventing the image from being interpreted accurately. A different approach to using a low-pass filter would be to
first identify hair-like structures and then use the filter on the local areas where these structures occur. [8] [9]

All of these methods have validity and use in their own right. It was not until DullRazor® was published in 1997 [8] that an automated algorithm was widely used for hair detection and removal. Since 1994 this group of authors had been conducting clinical studies to analyze skin lesions using computer image processing techniques. After producing results for the first step of their analysis of the images, automatic segmentation to separate the lesion from the surrounding skin, they noticed that the images with hairs in them, specifically darker thick hairs, were producing inadequate results due to the hairs occluding parts of the lesion and confusing the segmentation program. Due to this problematic behavior of the program, they developed the DullRazor® code specifically as a preprocessing tool for the removal of dark hairs from the lesion image. [8]

Because DullRazor® is focused on removing only thick dark hairs it consists of three basic parts: (1) identify the dark hair locations; (2) replace the hair pixels by the nearby non-hair pixels; and (3) smooth the final result [8]. Using grayscale morphological closing operations in each of the three color band images (Red, Green, and Blue) allows for smoothing of low intensity values, (binary masks 0° (horizontal), 45°(diagonal), and 90° (vertical) worked well and were used here). (2) Once the binary hair mask of the image is obtained for each color band interpolation is done across the hairs to replace the hair-pixel with an interpolation for the surrounding non-hair pixel
values. Because of the penumbra effect (darker pixels around the hair caused by the shadow of the hair on the skin) exact border location of the hairs are difficult to determine. To avoid this issue surrounding pixels used for interpolation are chosen 11 pixels away from the hair borders. (3) An adaptive median operator is also used to smooth out thin lines left from interpolation and hair removal. The resulting image has dull traces of the faded removed hairs, but overall the process works effectively enough to produce satisfactory results when doing segmentation of the lesion. Though the traces of these hairs could possibly be removed, this would occur at the excessive cost of fine detail in the image. [8]

The DullRazor® code [8] satisfactorily executes the task of removing obscuring hairs from the images. There are, however, two major shortcomings with the process. The first disadvantage is the continued obstruction of the image caused by the presence of thin and light colored hairs, not targeted by the DullRazor®. Since it only targets dark thick hairs, that the program ignores the removal of thin and lighter hairs leaving them behind to occlude the image and corrupt its features. The second disadvantage is that hair-like structures which possess similar coloring as the lesion can be left behind and negatively affect the results of the lesion diagnosis. [8] [9]

The authors of *E-Shaver: An improved DullRazor® for digitally removing dark and light-colored hairs in dermoscopic images* [9] noticed these shortcomings as well as the speed of the program and presented an algorithm designed to remove both dark and light hairs as well as reduce white spots produced by reflections of light due to
perspiration or oil on the skin. Similar to DullRazor®, the E-Shaver method has the following three steps: (1) detecting hair pixels; (2) replacing the detected hair pixels with non-hair pixels; and (3) smoothing. The program uses different methods for detecting hair pixels, allowing for light-colored hair detection as well, and for replacing the detected hair pixels, claiming their method needs fewer computations. [9]

To detect hairs the E-Shaver program first identifies thin structures using edge detection. After a performance comparison among the use of the Prewitt filter, Sobel filter, Laplacian operator, and the Laplacian of Gaussian (LoG) operator, the Prewitt filter was selected for use due to its higher value Signal to Noise Ratio (SNR) indicating a higher performance for hair detection. Using 3x3 and 5x3 Prewitt filters as horizontal edge detectors, and their transposes for vertical edge detectors, these filters can be used to detect hairs in the images. In order to decrease computation, dominant orientations of hairs in the images were determined using the Radon transform. Because the Radon transform has the capability to detect linear trends in images this program incorporated it to determine if a vertical, horizontal, or combination filters should be used based on the predominate trend of the hairs. Each hair pixel is then averaged with itself as well as its immediate neighbors. Masking is run three times to pull out all hairs and then the hairs replaced by averaging. [9]

Both programs, [8] [9], show success with their methods of hair removal. DullRazor® [8] works with thick dark hairs and can effectively remove and replace these hairs with interpolated values. While commendable for being a leader in this field,
DullRazor® [8] has shortcomings. While it is effective at removing hairs, the program can still leave behind artifacts. Due to the similarities in color between the hair and lesion, DullRazor® [8] can also miss hairs as well as create false indicators on the lesion skewing diagnosis. E-shaver approaches hair removal by targeting both light and dark hairs [9]. Though E-Shaver claimed their code was faster and more effective than DullRazor® this code can also leave behind hairs and artifacts [9]. In this thesis hair detection and removal is done through the use of the generalized Radon transform to allow for a broader class of hairs that can be detected and removed.
3 Radon Transform

In this thesis the Radon transform is an essential part of the hair detection process and therefore it is described in detail in this chapter. The Radon Transform was developed by Johann Karl August Radon, an Australian Mathematician [24] [10] [25]. In 1917, Radon published the Radon transform formula for reconstructing a function by taking line integrals through an object in various directions [11]. Since its development in 1917, the Radon transform has become a popular and indispensable tool in seismic data processing, image processing, tomography, etc. [24]. The Radon Transform is most commonly recognized for its use in the inverse Radon transform [25] [26] in Computer Axial Tomography (CAT) scans, also known as Computed Tomography scans (CT scan) [10]. A CAT scan is a medical imaging method which generates a two-dimensional image of the inside of a sliced section of an object - most often used in imaging portions of a human body. Performing multiple consecutive scans while moving down the scanned object produces a detailed three-dimensional image of that object and what is inside it [27]. Because the Radon transform is so versatile, it can be defined in many different ways. However, it is most commonly used for line detection applications in image processing [28]. The Radon transform has the capability to map two-dimensional images containing lines into a transform domain of possible line parameters. An image containing straight lines maps each line in the image to a peak positioned at the corresponding line parameters in the transform domain. This can be an extremely
useful tool in image processing because it is helpful for the detection and removal of lines in an image such as when hairs might need to be removed in a skin image.

3.1 Slant Stacking of the Radon Transform

3.1.1 Defining the \((p, \tau)\) Radon Transform

A simple and general definition of the Radon transform is found in the seismic data processing literature. The Radon transform, \(\tilde{g}(p, \tau)\), where \(p\) is the slope of the line and \(\tau\) is the line offset, of the continuous two-dimensional function \(g(x, y)\) can be found by slant stacking the values of \(g\) along various slanted lines in a plane [28]:

Equation 1 [28]

\[
\tilde{g}(p, \tau) = \int_{-\infty}^{\infty} g(x, px + \tau) \, dx
\]

This linear form of the Radon transform is known as slant stacking or the \(\tau-p\) transform. Using the Dirac delta function, \(\delta(.)\), the slant stacking function can then be written as:

Equation 2 [28]

\[
\tilde{g}(p, \tau) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y) \delta(y - px - \tau) \, dx \, dy
\]

This means that \(\tilde{g}(p, \tau)\) maps the \((x, y)\) image space to the two-dimensional \((p, \tau)\)-space (i.e., the Radon space or parameter domain) [28]. With this definition, and using Equation 1 and Equation 2, the \((p, \tau)\) Radon transform can easily be shown to exhibit a set of basic properties including linearity, shifting, scaling, point source, and line properties.
3.1.1.1 Basic Properties of the Radon Transform

3.1.1.1.1 Linearity

A linear function is a function which satisfies two properties: additivity, meaning that \( f(x + y) = f(x) + f(y) \); and homogeneity, meaning that \( f(\alpha x) = \alpha f(x) \) for all values \( \alpha \). From Equation 1, it is observed that the Radon transform follows the linearity property [28].

Equation 3 [28]

\[
\hat{h}(x, y) = \sum_i w_i g_i(x, y) \Rightarrow \\
\hat{h}(p, \tau) = \sum_i w_i \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g_i(x, y) \delta(y - px - \tau) \, dx \, dy \\
= \sum_i w_i \tilde{g}_i(p, \tau)
\]

In other words the Radon transform of the weighted sum of functions is equal to the weighted sum of the individual Radon transforms.

3.1.1.1.2 Shifting

Shifting of a function refers to the translation of the function’s location on the graph without altering the size and shape of the graphed function. Assuming function shifting of \( x^* \) and \( y^* \) in the image space and taking the Radon transform of a shifted function results in:
Equation 4 [28]

\[ h(x, y) = g(x - x^*, y - y^*) \Rightarrow \]
\[ \hat{h}(p, \tau) = \int_{-\infty}^{\infty} g(x - x^*, px + \tau - y^*) \, dx \]
\[ = \int_{-\infty}^{\infty} g(\tilde{x}, p(\tilde{x} + x^*) + \tau - y^*) \, d\tilde{x}, \text{ where } \tilde{x} = x - x^* \]
\[ = \hat{g}(p, \tau - y^* + px^*) \]

Equation 4 shows that the offset parameter of the Radon transform changes by the shifting of the function. This reinforces the fact that the slope of a line, \( p \), cannot be changed by a translation; however, the offset of the line, \( \tau \), more commonly known as the \( y \)-intercept will be shifted because of the movement of the function with respect to the \( x \) and \( y \) axes [28]. This can be verified similarly by shifting a line by \( x^* \) and \( y^* \):

\[ y = px + \tau \rightarrow y - y^* = p \cdot (x - x^*) + \tau \]

Based on this one can see that, similar to equation 4, the slope is unaffected by the image shifting, yet the \( y \)-intercept has shown a change with respect to \( y \).

3.1.1.3 Scaling

Scaling a function corresponds to a linear transformation of the independent variable axes. The equations below demonstrate a scaled function for the Radon transform:
This process shows that a compression in the y-direction presents a compression in the offset, $\tau$. The compression amount is the same factor of $b$ for which $y$ is scaled. This process also shows that the slope is scaled by a ratio of the scaling parameters $a$ and $b$ coupled with scaling the entire Radon transform by the factor $a$. [28]

### 3.1.1.4 Point Source Example

A point source can be defined as the product of two delta function. Without shifting, the point source will appear at the origin of the assumed coordinate system, demonstrated below [28].

\begin{equation}
\begin{align*}
    h(x, y) &= g\left(\frac{x}{a}, \frac{y}{b}\right) \text{ where } a > 0 \text{ and } b > 0 \\
    \tilde{h}(p, \tau) &= \int_{-\infty}^{\infty} g\left(\frac{x}{a}, \frac{px + \tau}{b}\right) dx \\
    &= a \int_{-\infty}^{\infty} g\left(\frac{\tilde{x}}{b}, \frac{pa\tilde{x} + \tau}{b}\right) d\tilde{x} \text{ where } \tilde{x} = \frac{x}{a} \\
    &= a \tilde{g}\left(\frac{pa}{b}, \frac{\tau}{b}\right)
\end{align*}
\end{equation}

Shifting the point source can place it at any position in the proposed coordinate plane [28]:

\begin{equation}
\begin{align*}
    g(x, y) &= \delta(x - x^*) \delta(y - y^*) \Rightarrow \tilde{g}(p, \tau) = \delta(\tau - y^* + px^*)
\end{align*}
\end{equation}
Expanding this, one can see that any function can be written as a weighted integral of point sources [28].

Equation 7 [28]

\[
g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x^*, y^*) \delta(x - x^*) \delta(y - y^*) \, dx^* \, dy^* \implies \\
g(p, \tau) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x^*, y^*) \delta(x - x^*) \delta(\tau + px - y^*) \, dx^* \, dy^* \, dx \\
= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x^*, y^*) \delta(y^* - \tau - px^*) \, dx^* \, dy^*
\]

This property shows that any point of a function can contribute along an infinitely long line in the slope-offset parameter domain. When a point source is detected by the Radon transform the resulting graph in the Radon dimension, \( \tilde{g}(p, \tau) \), is presented as a line. Graphically this can be seen in Figure 6:

![Figure 6: Left: A two dimensional function that only is non-zero in the point \((x, y) = (x^*, y^*)\). Right: The corresponding Radon transform (slant stacking result). Only when the Radon domain parameters match the parameters of the line a non-zero result is found. [28]](image)

### 3.1.1.1.5 Line
The Line property is based on the assumption that a line can be modeled through a delta function in the following way:

**Equation 8** [28]

\[ g(x, y) = \delta(y - p^* x - \tau^*) \]

This function only has non-zero values if \((x, y)\) lie on the line with the same certain fixed parameters \((p^*, \tau^*)\). In this case the Radon transform is given by the equation below:

**Equation 9** [28]

\[
\tilde{g}(p, \tau) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \delta(y - p^* x - \tau^*) \delta(y - px - \tau) \, dx \, dy \\
= \int_{-\infty}^{\infty} \delta((p - p^*)x + \tau - \tau^*) \, dx \\
= \left\{ \begin{array}{ll}
\frac{1}{|p - p^*|} & \text{for } p \neq p^* \\
0 & \text{for } p = p^* \text{ and } \tau \neq \tau^* \\
\int_{-\infty}^{\infty} \delta(0) \, dx & \text{for } p = p^* \text{ and } \tau = \tau^*
\end{array} \right.
\]

It is worthwhile to note that for \(p = p^*\) and \(\tau = \tau^*\) the value of the Radon transform is infinite due to the fact that integrating an infinite function over an infinite interval is, of course, infinite [28]. While neglecting the finite terms, the result of the Radon transform of a line is the production of a peak (with an infinite value) in the line parameter domain at the position of the matching line parameters [28].

The line property is probably the most important property of the Radon transform because of its implications in image processing and computer vision. This
property allows the Radon transform to detect lines in an image and display their parameters in the Radon domain, in this case through the line’s slope and offset. Naturally, this property has been a basis of numerous curve parameter detection algorithms [28] allowing for the determination of location and line information in images.

![Figure 7: Left: A two dimensional function that is only non-zero when on the line. Right: The corresponding Radon transform (slant stacking result). When the Radon domain parameters match the parameters of the line, a peak is found positioned at the parameters of the line in the image. The finite terms in the parameter domain are here ignored for sake of clarity. [28]

The result of the Radon transform applied to a line is shown symbolically in Figure 7. As noted, when the Radon transform is taken of a line it results in a point source in the Radon/line parameter domain. From the symbolic representation of both the line and point source properties it is easy to see that there exists a duality between the two domains. A point represented in the image domain is transformed into a line in the Radon domain. Similarly, if finite terms are ignored, a line in the image domain is transformed into a point in the Radon domain [28].

### 3.1.2 Discrete Slant Stacking
It is important to define an appropriate discrete Radon transform for the transformation of digital images. The use of discrete Radon transforms is so universal for image processing, that depending on speed, artifacts, or simplicity requirements, a new version of it is often developed for nearly every application. In [28] a simple representation for sampling four variables to work with the discrete Radon transform is given by:

**Equation 10 [28]**

\[
x = x_m = x_{\text{min}} + m \Delta x, \quad m = 0, 1, \ldots, M - 1
\]
\[
y = y_n = y_{\text{min}} + n \Delta y, \quad n = 0, 1, \ldots, N - 1
\]
\[
p = p_k = p_{\text{min}} + k \Delta p, \quad k = 0, 1, \ldots, K - 1
\]
\[
\tau = \tau_h = \tau_{\text{min}} + h \Delta \tau, \quad h = 0, 1, \ldots, H - 1
\]

Thus \(x_{\text{min}}\) is the position of the first sample, \(\Delta x\) is the sampling distance in the \(x\) direction, and \(m\) is the discrete index of the \(M\) samples of \(x\).

Using these variables it is then possible to represent the discrete Radon transform by an approximating sum of the continuous linear Radon transform:

**Equation 11 [28]**

\[
\tilde{g}(p_k, \tau_h) = \int_{-\infty}^{\infty} g(x, p_k x + \tau_h) \, dx \approx \Delta x \sum_{m=0}^{M-1} g(x_m, p_k x_m + \tau_h)
\]

Sampling this function \(g(x, y)\) gives the digital image \(g(m, n) = g(x_m, y_n)\) [28].

Due to the nature of the sampling in Equation 11, a fundamental problem can occur when the equation creates samples that are not found in the digital image. Because of this linear sampling of all variables in the equation \(p_k x_m + \tau_h\) will never
overlap with \( y_n \). In order to fix this sampling error nearest neighbor interpolation, linear interpolation, or sinc interpolation can be employed. [28]

Sampling in the parameter domain must be done with sufficient density to avoid aliasing which can occur in discrete approximations of the Radon transform. Other frequent problems occur when one or more lines in the image have nearly vertical slopes. When this happens the line may be missed completely due to the magnitude of its slope in the truncated parameter domain. Similarly, only lines which lie inside the limits of the parameter domain will be detected. Because of this it is necessary that a parameter domain with numerous samples is created when there is no prior information about the line(s) in the image. [28]

3.1.2.1 Nearest Neighbor Interpolation

One of the most effective and probably fastest ways of correcting the sampling problem is to use nearest neighbor interpolation in the \( y \)-direction. Doing this will change the Radon transform equation to be represented as follows:

Equation 12 [28]

\[
\tilde{g}(k, h) = \Delta x \sum_{m=0}^{M-1} g(m, n(m; k, h)) \text{ where } n(m; k, h) = \left[ \frac{p_k x_m + \tau_h - y_{min}}{\Delta y} \right]
\]

where the \([.]\) notation in Equation 12 represents rounding of the internal value to the nearest integer. The \( \Delta x \) factor can generally be neglected; however, it is needed when the discrete Radon transform is used to quantitatively approximate the continuous ones. Another problem is that the point \((m, n(m; k, h))\) will not necessarily lie within
the finite image. When this point is outside of the image its value, which is needed for the transform, its value can be set to zero and therefore provide no contribution to the transform [28]. The discrete Radon transform, as with the continuous version, is also a linear function.

3.2 Discrete Radon Transform of a Discrete Line

Because of the linearity of the Radon transform, a peak should be present in the Radon parameter domain for each line expressed in an image. The ability to handle and identify multiple lines is a major strength of the Radon transform in image processing. Following are examples of code used to produce Radon transforms of an image through two different methods of interpolation: nearest neighbor and linear interpolation. In these sections the code and its products will be shown as examples of the working Radon transform on an image with two lines. In these images the white value represents a maximum, whereas the black value is set to a minimum value of zero. It should also be noted that the y-axis goes from negative to positive in the opposite direction (going from top to bottom of the image).

3.2.1 Nearest Neighbor Interpolation Radon Transform of Discrete Line

To test the capabilities of the linear Radon transform a synthetic line image of 200x200 pixels was generated. The nearest neighbor code (included at the end of the section in Table 3) for the Radon transform was then run over the line image. The image contains two lines, seen in Figure 8, with line parameters of \((p_1 = 1, \tau_1 = 0)\) and \((p_2 = 0.5, \tau_2 = -50)\).
Due to the linearity property of the Radon transform the transform of the image should have two peaks in the Radon domain corresponding to the lines’ parameters in the image domain. As can be seen in Figure 9, the corresponding peaks in the Radon domain are analogous to the parameters of the lines in Figure 8, the image space. This shows that from the location of the peaks it is possible to tell the parameters of the lines and therefore their position and orientation in the image domain.
By changing the sampling parameters it is possible to zoom in on a single peak. It is also possible to alter the parameters to zoom out in the Radon domain which can lead to missing a line altogether. These examples, shown in Figure 10 and Figure 11, show how important the sampling of parameters is to the detection of the line in the image domain.

As can be seen in Figure 10 the broadness of the peak is increased because of the increased density of the parameter values, as shown in Table 2. This zoom will be...
helpful when comparing the peaks of the nearest neighbor interpolation and the linear interpolation Radon transform methods. When sampled thoroughly the peak is broader and clear. Though this image is zoomed in to show the effect this density can have on the Radon domain, if it were done on the same scale as Figure 9 with the density of Figure 10, this program would take a significantly long time to run and would be computationally impractical for use.

Despite the issue of high density increasing the run time of the program significantly, problems can also occur if the transform is run with insufficient density. If the image is sampled very broadly it is possible for the transform to completely miss lines in the image. Running the broadly sampled Radon produced the resulting image in Figure 11. As can be seen only one peak is found in the Radon domain, however two were present in the image. This is due to the sampling parameters, shown in Table 2, being insufficiently dense. Not having enough density in the parameter domain results in a prominent line left out of the Radon domain.

As can be seen with both of these examples one must be extremely careful when picking sampling parameters and ranges for the Radon domain. If previous line parameter knowledge is not known in advance it is best to use a finer density and a broader range to avoid missing lines. Despite the fact that and increased density would help detect more lines, the down side is computing time. This trade-off can be tricky to maneuver and may lead to the acceptance of an increased computing time or an acceptable error in detection.
Table 3: Discrete Slant Stacking Nearest Neighbor Radon Transform Code [28]
(NOT OPTIMIZED)

```matlab
function [g_radon] = SS_RadonT_NN(image, p, tau, Delta_x, Delta_y, x_min, y_min)
% Discrete Slant Stacking with nearest neighbor Radon Transform
[M,N] = size(image); |wx| gives the size of the image
K = length(p);
H = length(tau);
g_radon = zeros(K,H); |wx| pre-allocation size of g_radon
for k = 1:K |wx| % for all values of p
    alpha = p(k)*Delta_x/Delta_y; |wx| % Calculate digital slope
        for h = 1:H |wx| % For all values of tau
            beta = (p(k)*x_min + tau(h)-y_min)/Delta_y; |wx| % Calculate digital offset
                sum = 0; |wx| % initializing the sum
                    for m = 0:M-1
                        n=round(alpha*m+beta);
                            if (inside(m+1,n+1,M,N))
                                sum = sum + image(m+1,n+1); |wx| % increment the sum
                            end
                    end
                    g_radon(k,h) = sum*Delta_x; |wx| % update matrix element
            end
        end
end
```

Table 3: Discrete Slant Stacking Nearest Neighbor Radon Transform Code

3.2.2 Comparison of Nearest Neighbor and Linear Interpolation

There are many different ways to compute the Radon transform. When dealing with slant stacking the interpolation method that is used is extremely important - not only for how it affects processing time but also for determining the effectiveness of peak detection. In this section the interpolation methods for both linear interpolation (see implementation code in Table 6, and the nearest neighbor Radon transforms are examined.
Figure 12: Image domain containing two lines \((p_1 = 1, \tau_1 = 0)\) and \((p_2 = 0.5, \tau_2 = -50)\).

Figure 13: Corresponding discrete Radon transform done by Linear Interpolation slant stacking

<table>
<thead>
<tr>
<th>IMAGE DOMAIN</th>
<th>RADON DOMAIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>Value</td>
</tr>
<tr>
<td>M</td>
<td>200</td>
</tr>
<tr>
<td>N</td>
<td>200</td>
</tr>
<tr>
<td>(\Delta x)</td>
<td>1</td>
</tr>
<tr>
<td>(\Delta y)</td>
<td>1</td>
</tr>
<tr>
<td>(x_{\text{min}})</td>
<td>-100</td>
</tr>
<tr>
<td>(y_{\text{min}})</td>
<td>-100</td>
</tr>
</tbody>
</table>

Table 4: Set sampling parameters of image and Radon domains for Figure 12 and Figure 13

In Figure 13 a linear interpolation method was used to create the Radon domain of the image in Figure 12. Comparing the two methods shown in Figure 9 and Figure 13 (both transform the same image), at this density and distance the transforms appear the same. Allowing for computing time, the nearest neighbor approach is slightly faster than the linear interpolation approach due to the linear interpolation method requiring a summation of two samples for every pixel \(m\).
When performing a parameter zoom with the linear interpolation Radon transform on the image a difference can be seen between the two methods. Not only is the program execution more time intensive when using linear interpolation as compared to nearest neighbor interpolation, but the peaks that are present using linear interpolation are less defined and “foggy”. When performing peak detection for automated recognition of the parameters for lines in the Radon domain these poorly
defined peaks can cause issues in revealing what lines present in the original image
and/or lines can potentially be missed or attributed to noise.

<table>
<thead>
<tr>
<th>Table 6: Discrete Slant Stacking Liner Interpolation Radon Transform Code [28] (NOT OPTIMIZED)</th>
</tr>
</thead>
<tbody>
<tr>
<td>function [g_radon] = SS_RadonT(image, p, tau, Delta_x, Delta_y, x_min, y_min)</td>
</tr>
<tr>
<td>% Discrete Slant Stacking with Linear Interpolation Radon Transform</td>
</tr>
<tr>
<td>[M,N] = size(image); % gives the size of the image</td>
</tr>
<tr>
<td>K = length(p);</td>
</tr>
<tr>
<td>H = length(tau);</td>
</tr>
<tr>
<td>g_radon = zeros(K,H); % pre-allocation size of g_radon</td>
</tr>
<tr>
<td>for k = 1:K % for all values of p</td>
</tr>
<tr>
<td>alpha = p(k)*Delta_x/Delta_y; % Calculate digital slope</td>
</tr>
<tr>
<td>for h = 1:H % for all values of tau</td>
</tr>
<tr>
<td>beta = (p(k)*x_min + tau(h)-y_min)/Delta_y; % Calculate digital offset</td>
</tr>
<tr>
<td>% setting m_min to m_max</td>
</tr>
<tr>
<td>if alpha &gt; 0;</td>
</tr>
<tr>
<td>m_min = max(1, ceil((-beta+1)/alpha)); % where N is the image n</td>
</tr>
<tr>
<td>m_max = min(M, floor((N-beta)/alpha)); % where M is the image m</td>
</tr>
<tr>
<td>else alpha &lt; 0;</td>
</tr>
<tr>
<td>m_min = max(1, ceil((N-beta)/alpha)); % where N is the image n</td>
</tr>
<tr>
<td>m_max = min(M, floor((-beta+1)/alpha)); % where M is the image m</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>sum = 0; % initializing the sum</td>
</tr>
<tr>
<td>% using m_min and m_max</td>
</tr>
<tr>
<td>for m = m_min:m_max % for all valid values of x</td>
</tr>
<tr>
<td>nfloat = alpha*m + beta; % use nearest neighbor approx.</td>
</tr>
<tr>
<td>n = floor(nfloat); % calculate lower index</td>
</tr>
<tr>
<td>w = nfloat - n; % calculate weight factor</td>
</tr>
<tr>
<td>if (inside(m,n,M,N) &amp;&amp; inside(m,n+1,M,N))</td>
</tr>
<tr>
<td>sum = sum + image(m,n)*(1-w) + image(m,n+1)*w; % increment the sum</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>g_radon(k,h) = sum*Delta_x; % update matrix element</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>end</td>
</tr>
</tbody>
</table>
3.3 The Normal Radon Transform

3.3.1 Defining the \((\rho, \theta)\) Radon Transform

An alternative form of the linear general Radon transform can be defined as follows:

\[\begin{equation}
\tilde{g}(\xi_0, \xi_1, \xi_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y) \delta(\xi_0 - \xi_1 x - \xi_2 y) \, dx \, dy
\end{equation}\]

In this form a line is defined with three degrees of freedom. Allowing for the fact that a line only has two degrees of freedom this means two of the parameters must be “linked”, allowing for the removal of one degree of freedom. There are two most common forms used to describe the parameters [28]. The first form is slant stacking, where the parameters are defined as:

\[\begin{equation}
(\xi_0, \xi_1, \xi_2) = (-\tau, p, -1)
\end{equation}\]

Another definition of the Radon transform, common in the fields of tomography, astronomy, and microscopy; uses the fundamental function \(g(x, y)\) in a form where it has no preferred orientation. In this form a line in its normal form is given by: [28]

\[\begin{equation}
\rho = x \cos \theta + y \sin \theta
\end{equation}\]
Using this as the definition for the line sets the parameter values equal to:

\( \rho, x \cos \theta, y \sin \theta \); i.e., \((\xi_0, \xi_1, \xi_3) = (\rho, \cos \theta, \sin \theta)\). For practical purposes this will be referred to as the normal Radon transform to distinguish it from the slant stacking Radon transform. The normal Radon transform for these purposes can be expressed as:

\[
\tilde{g}(\rho, \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y) \delta(\rho - x \cos \theta - y \sin \theta) \, dx \, dy
\]

The parameters used for the normal Radon transform are used to specify the position of the line by the parameters \( \rho \), the shortest distance from the origin of a coordinate system to the line, and \( \theta \), the angle of the corresponding line orientation. These are illustrated in Figure 16: [28]

![Figure 16: The two parameters \( \rho \) and \( \theta \) used to specify the position of the line.](image)

Connected to slant stacking, the normal Radon transform can be expressed from the manipulation of Equation 16 [28] by:

\[
\tilde{g}(\rho, \theta) = \int_{-\infty}^{\infty} g(\rho \cos \theta - s \sin \theta, \rho \sin \theta + s \cos \theta) \, ds
\]
As can be seen the connection between the normal and slant stacking Radon transforms two parameters can be found directly without complex computations. [28]

One way of approximating the normal Radon transform to change it into a one-dimensional interpretation in the image domain is to reformulate the slant stacking as is presented in Equation 16. If done correctly, as below, then the solution will only require one-dimensional interpolation. This can be done through nearest neighbor interpolation (as shown below in Equation 18) or through linear interpolation. In Equation 18 \( \Delta x = \Delta y, M = N, \text{ and } x_{min} = y_{min}. \)

\[ \sin \theta > \frac{1}{\sqrt{2}} : \hat{g}(\rho, \theta) = \frac{1}{|\sin \theta|} \hat{g} \left( p = -\cot \theta, \tau = \frac{\rho}{\sin \theta} \right) \]

\[ \approx \frac{\Delta x}{|\sin \theta|} \sum_{m=0}^{M-1} g \left( m, \left[ \alpha m + \beta \right] \right) \]

where \( \alpha = -\cot \theta \text{ and } \beta = \frac{\rho - x_{min}(\cos \theta + \sin \theta)}{\Delta x \sin \theta} \)

\[ \sin \theta \leq \frac{1}{\sqrt{2}} : \hat{g}(\rho, \theta) = \frac{1}{|\cos \theta|} \hat{g}(r = -\tan \theta, \eta = \frac{\rho}{\cos \theta}) \]

\[ \approx \frac{\Delta x}{|\cos \theta|} \sum_{n=0}^{M-1} g \left( \left[ \alpha n + \beta \right], n \right) \]

where \( \alpha = -\tan \theta \text{ and } \beta = \frac{\rho - x_{min}(\cos \theta + \sin \theta)}{\Delta x \cos \theta} \)

\[ 3.4 \text{ Radon Transform for Curve Detection} \]
A frequent and popular problem in image processing revolves around the identification of curves in an image. With the Radon transform this issue can be considerably simplified due to its ability of mapping lines in the image domain to a parameter domain. Through an extension of the Radon transform can be made to deal with curve detection where the extended Radon transform’s parameter domain can be used to characterize detected curves. Thus through the use of an extended Radon transform a difficult curve detection issue in the image domain can become a simple local peak detection in the parameter domain.

3.4.1 The Generalized Radon Transform

For the purposes of this section consider $g(x, y)$ to be a continuous signal of the continuous variables $x$ and $y$. Let $\xi$ be a $\eta$-dimensional parameter vector defined as

$$\xi = (\xi_1, ..., \xi_i, ..., \xi_\eta)$$

where $\xi$ spans the parameter domain.

3.4.1.1 The Continuous Generalized Radon Transform

Because the Radon transform can be defined in many ways, a general form can be given by:

$$\tilde{g}(\xi) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y) \delta(\phi(x, y; \xi)) \, dx \, dy$$
Using this form it is possible to find shapes in the image which are expressed where the parameters satisfy $\Phi(x, y; \xi) = 0$. Letting $\tilde{g}(\xi)$ be the continuous generalized Radon transform of $g(x, y)$, Equation 20 becomes:

\[
\tilde{g}(\xi) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y) \, \delta(y - \phi(x; \xi)) \, dx \, dy = \int_{-\infty}^{\infty} g(x, \phi(x; \xi)) \, dx.
\]

Here $y = \Phi(x; \xi)$ is the characterized transformation curve. Thus the Radon transform is now an integration of $g(x, y)$ along the transformation curve. [28]

Equation 21 is an interpreted from of the generalized slant stacking technique previously described. In its current form this equation only allows for a singular value of $y$ for each value of $x$ and the parameter $\xi$, meaning this form excludes any closed curves [28].

The curve to be detected can now be modeled by using a delta function which follows the shape of the curve.

\[
g(x, y) = \delta(y - \psi(x; \xi^*)
\]

Letting $g(x, y)$ be given by Equation 22 the generalized Radon transform can now be written as:
Equation 23 [28]

$$
\tilde{g}(\xi) = \int_{-\infty}^{\infty} \delta(\phi(x; \xi) - \phi(x; \xi^*)) \, dx \\
= \int_{-\infty}^{\infty} \sum_{i=1}^{l} \frac{\delta(x - x_i)}{\left| \frac{\partial \phi(x; \xi)}{\partial x} - \frac{\partial \phi(x; \xi^*)}{\partial x} \right|} \, dx \\
= \sum_{i=1}^{l} \frac{1}{\left| \frac{\partial \phi(x_i; \xi)}{\partial x} - \frac{\partial \phi(x_i; \xi^*)}{\partial x} \right|}
$$

where each of the \( l \) values \( x_i \) will fulfill \( \Phi(x_i; \xi) = \Phi(x_i; \xi^*) \). [28]

In Equation 23 the generalized Radon transform gives an infinite peak at \( \xi = \xi^* \). Unfortunately, this equation has a transform malfunction when the slope of the curve is infinite, i.e. when the tangent to the curve in that \((x, y)\) plane is vertical, meaning that this definition would be suitable to be used with the curve detection of parabolas, hyperbolas, and other curves which have limited slopes. [28]

3.4.1.2 The Discrete Generalized Radon Transform

Letting \( j \) be the \( \eta \)-dimensional discrete index parameter vector defined as [28]:

Equation 24 [28]

$$
j = (j_1, \ldots, j_i, \ldots, j_n)
$$

Linking the index vector \( j \) and the sampled parameter vector, \( \xi_j \):

Equation 25 [28]

$$
\xi = \xi_j = \theta(j), \quad \text{where} \quad \xi_i = \theta_i(j_i)
$$

where the parameter sampling frequency is \( \theta_i(j_i) \). [28]
Using uniform sampling of the parameter domain then \( \theta_i \) can be written as:

**Equation 26** [28]

\[
\xi_i = \theta_i(j_i) = \xi_{i,\text{min}} + j_i \Delta \xi_i, \quad j_i = 0, \ldots, J_i - 1
\]

Here, \( \xi_{i,\text{min}} \) is the lower limit and \( \Delta \xi_i \) of the sampling interval \( \xi_i \). [28]

If sampling is done uniformly in both the image and parameter domains then Equation 25 can be written as a transformation curve:

**Equation 27** [28]

\[
y = y_{\text{min}} + n \Delta y = \phi(x_{\text{min}} + n \Delta x; \theta(j))
\]

Using this approximation alongside the nearest neighbor interpolation, the discrete transformation curve can be denoted as [28]:

**Equation 28** [28]

\[
\phi(m; j) = n = \left[ \frac{\phi(x_{\text{min}} + m \Delta x; \theta(j)) - y_{\text{min}}}{\Delta y} \right]
\]

Letting \( g(m, n) \) be the discretized version of the signal \( g(x, y) \), where \( g(m, n) = g(x_m, y_n) \), then the discrete generalized Radon transform is finally defined by [28]:

**Equation 29** [28]

\[
\tilde{g}(j) = \sum_{m=0}^{M-1} g(m, \phi(m; j))
\]
3.4.2 Image Point Mapping

To develop an alternative method of estimating Equation 21 in a discrete parameter domain we can represent $\hat{g}(j)$ as:

Equation 30 [28]

$$\hat{g}(j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} g(m, n) \delta(n - \phi(m; j))$$

This representation allows for mapping each image point $(m, n)$ into the parameter curve domain where $n = \Phi(m; j)$. [28]
4 Generalized Quadratic Radon Transform

In order to remove curved hairs in an image, a quadratic Radon transform was developed to detect and locate curvature in an image. Initial hair masking of the image was done to differentiate the hairs from lesion and skin. This process transforms an image into a binary hair mask containing hair lines and noise. The quadratic Radon transform is then used across the image to separate hairs from noise pixels. By detecting peaks in the Radon domain, pixels can then be classified as hair or noise. If a pixel is classified as part of a hair an interpolation method is used to replace it, effectively “shaving” the hair from the image.

4.1 Initial Hair Masking: The DullRazor® [28] Method

DullRazor®’s [8] dark hair image masking technique was used for the initial determination of hair locations. Though other masking techniques can be used to detect a variety of different hair types, such as the method presented in E-Shaver [9] for the detection of thin light hairs, the focus in this research was to study curved hair detection and not on obtaining and creating an optimal hair detection mask.

The masking technique from DullRazor® creates a binary mask of the original image consisting of pixels having values of ones or zeros, where a one is associated with a hair or noise pixels. This method uses a grayscale morphological closing operation applied to the three color bands in the original Red/Green/Blue (RGB) image, [8]. From their experimental results it was inferred that three structural elements, all in different
directions, are adequate to detect all of the dark hairs. These elements consist of three image masks in directions of 0° (horizontal), 45° (diagonal), and 90° (vertical), as seen below in Figure 17 [8].

![Figure 17: Structure elements from DullRazor® for the generalized closing operation. (a) horizontal structure element, (b) diagonal structure element, and (c) vertical structure element [8]](image)

The generalized grayscale closing image is obtained by using the maximum response of the three specified grayscale closings on each color band image. Finally the binary hair mask image is then created by thresh holding the absolute difference between the original color band and the generalized grayscale closing image, therefore the hair mask divides the hair and non-hair regions into disjointed areas. [8]

Let $G_r$ be the generalized grayscale closing image of the original image red band, $O_r$. $S_0$, $S_{45}$, and $S_{90}$ will be the structure elements presented in Figure 17, where the subscripts correspond to the angle of rotation. This allows $G_r$ to be represented as [8]
Equation 31 [8]

\[ G_r = |O_r - \max\{O_r \cdot S_0, O_r \cdot S_{45}, O_r \cdot S_{90}\} | , \]

where \( \cdot \) is the grayscale closing operation. The binary mask of the image, \( M_r(x, y) \), is then thresholded with a predefined value at the location of \( (x, y) \) by computing [8]

Equation 32 [8]

\[
M_r(x, y) = \begin{cases} 
1, & \text{if } G_r(x, y) > T \\
0, & \text{if otherwise}
\end{cases}
\]

This is then repeated for the green and blue band images with the final hair mask for the original image, \( M \), given by the union of all three masks.

Equation 33 [8]

\[ M = M_r \cup M_g \cup M_b , \]

where the \( M_r, M_g, \) and \( M_b \) are the analogous hair masks for the corresponding color bands. The following, in Table 7, is the code written and used to mask RGB images into binary hair masks. Additional alteration to the code was done through a median filter being used on the image RGB layers after separation. This process was found to be helpful for the initial masking process of the image.

<table>
<thead>
<tr>
<th>Table 7: Initial Hair Masking Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>%% Loading in image and finding size</td>
</tr>
<tr>
<td>O_image = imread('ImageName.type');</td>
</tr>
<tr>
<td>[mimage, nimage] = size(O_image);</td>
</tr>
<tr>
<td>%%% Masks for closing operations</td>
</tr>
<tr>
<td>mask_0 = [0 1 1 1 1 1 1 1 1 1 0];</td>
</tr>
<tr>
<td>mask_45 = [0 0 0 0 0 0 0 0 0 0;</td>
</tr>
<tr>
<td>0 1 0 0 0 0 0 0];</td>
</tr>
</tbody>
</table>
% DullRazor Hair Detection method

H_image = double(O_image);

%% Separating image into RGB layers (addition of median filter for better hair masking)
O_r = medfilt2(double(O_image(:,:,1)),[3,3]);
O_g = medfilt2(double(O_image(:,:,2)),[3,3]);
O_b = medfilt2(double(O_image(:,:,3)),[3,3]);

%% closing mask on red image
mr1 = imclose(O_r, mask_0);
mr2 = imclose(O_r, mask_45);
mr3 = imclose(O_r, mask_90);
max_r = max(mr1, mr2);
max_r = max(max_r, mr3);  % max of all masks
G_r = abs(O_r - max_r);  % Generalized grayscale closing image (red)
M_r = zeros(mimage, nimage);
for i = 1:mimage
    for j = 1:nimage
        if G_r(i,j) > 25
            M_r(i,j) = 1;
        else
            M_r(i,j) = 0;
        end
    end
end

%% closing mask on green image
mg1 = imclose(O_g, mask_0);
mg2 = imclose(O_g, mask_45);
mg3 = imclose(O_g, mask_90);
max_g = max(mg1, mg2);
max_g = max(max_g, mg3); %max of all masks
G_g = abs(O_g - max_g); %Generalized grayscale closing image (green)
M_g = zeros(mimage, nimage);
for i = 1:mimage
    for j = 1:nimage
        if G_g(i,j) > 25
            M_g(i,j) = 1;
        else
            M_g(i,j) = 0;
        end
    end
end

%% closing mask on blue image
mb1 = imclose(O_b, mask_0);
mb2 = imclose(O_b, mask_45);
mb3 = imclose(O_b, mask_90);
max_b = max(mb1, mb2);
max_b = max(max_b, mb3); %max of all masks
G_b = abs(O_b - max_b); %Generalized grayscale closing image
M_b = zeros(mimage, nimage);
for i = 1:mimage
    for j = 1:nimage
        if G_b(i,j) > 25
            M_b(i,j) = 1;
        else
            M_b(i,j) = 0;
        end
    end
end

%% Union of hair masks for three color bands
M_image = zeros(mimage, nimage);
for i = 1:mimage
    for j = 1:nimage
        if M_g(i,j) * M_b(i,j) == 1
            M_image(i,j) = 1;
        else
            M_image(i,j) = 0;
        end
    end
end
if (M_r(i,j) | M_g(i,j) | M_b(i,j)) == 1;
    M_image(i,j) = 1;
else
    M_image(i,j) = 0;
end
end
image = M_image; % setting variable name for final coding use

Table 7: Reproduction code of DullRazor®’s description of hair masking code [28]

In order to decrease the computational complexity of the implementation of the final hair detection algorithm, the transform was only run on pixels that could possibly be hairs, i.e., those pixels having a masked value of 1.

4.2 Generalized Quadratic Radon Transform

Adapting the generalized Radon transform

Equation 34:

$$g(\xi) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y) \delta(y - \phi(x; \xi)) \, dx \, dy = \int_{-\infty}^{\infty} g(x, \phi(x; \xi)) \, dx$$

for quadratic curve detection means defining the equation for $\phi$ based on the generalized quadratic equation. This leads to:

Equation 35:

$$\phi(x: \alpha, \theta, p) = \text{rot}_\theta(\alpha(x - p_1)^2) + p_2$$

where $\alpha$ is the scaling constant determining the curvature of the quadratic curve, $p_1$ is the lateral shifting of the quadratic in the $x$-direction, $p_2$ is the vertical shifting of the
quadratic in the $y$-direction, and $rot_\theta$ is the rotation transformation of the quadratic by $\theta$.

### 4.2.1 Generalized Centered Quadratic

A quadratic function is a single variable polynomial of a second degree. The general function for the quadratic can be expressed as

**Equation 36: General Quadratic Equation**

$$y = \alpha x^2 + \beta x + c$$

where $y$ is the function of $x$, $x$ is the representative unknown variable and $\alpha, \beta,$ and $c$ are constant quadratic coefficients, where $a \neq 0$.

When the quadratic is positioned on the origin its $\beta$ and $c$ coefficients are equal to zero, i.e. $\beta = c = 0$. This makes the generalized formula

**Equation 37: General Non-Shifted Quadratic Equation- centered at origin**

$$y = \alpha x^2.$$ 

Depending on the value of $\alpha$ the graph of the quadratic about the origin will be altered.

Consider the examples given in Figures 18 and 19:
Figure 18 and Figure 19 show the graphs of the general non-rotated quadratic with both positive and negative values of $\alpha$. As can be seen, negative values of $\alpha$ simply correspond to a reflection of the positive across the $x$-axis. When $\alpha > 1$, the quadratic undergoes compression where the width between the two tails becomes smaller. Similarly, when $\alpha < 1$ an expansion occurs increasing the width between the two tails, until finally when $\alpha = 0$ the quadratic degenerates into a linear function.

4.2.1.1 Rotation ($\theta$)

To address the problem of curve detection it was necessary to alter the generalized (centered) quadratic so that it could be rotated around the origin. Because the previous equation does not pick up all orientations of curves it was necessary to use a rotational transform on the quadratic, therefore allowing the parameters to be $\alpha$ (curvature) and $\theta$ (rotation) for the function and Radon domain. The rotational
transform allows the point \((x, y)\) to be rotated around the origin ending in the location \((x', y')\). To carry out this rotation the following transformation was used,

\[
\begin{bmatrix}
    x' \\
    y'
\end{bmatrix} =
\begin{bmatrix}
    \cos \theta & -\sin \theta \\
    \sin \theta & \cos \theta
\end{bmatrix}
\begin{bmatrix}
    x \\
    y
\end{bmatrix}
\]

Using this transformation it was possible to rotate the quadratic around the origin of the image. Once this was accomplished the discrete Radon transform code could be altered to allow for the development of a curve detector.

4.2.1.2 Curvature \((\alpha)\)

With the quadratic function it was shown that the \(\alpha\) will alter not only the width of the graph but also the orientation on the axes. Also, as full 360° rotations are considered, it is not necessary to consider negative values of \(\alpha\). This only factor remaining is the full width of the oriented curve. Generalizing the expansion curves for all possible hairs, the \(\alpha\) was set to a parameter range of \(\min \alpha = 0, \text{step size } \alpha = 0.01, \text{and } \max \alpha = 1\).

4.2.2 Quadratic Radon Transform Implementation

When implementing the quadratic Radon transform it was essential to optimize the code for fast processing. In order to do this a vectorization of the algorithm was necessary to improved computational speed. As the transform allows for a multitude of parallel computations it was possible to improve the computational speed considerably through preprocessing, calculations, and vectorization of variables.
Inputs for the Radon Transform include a binary image containing curves (hairs), a scalar \( \Delta x \) (normally equal to 1 because it refers to the step of the pixels in the \( x \) direction), a vector of \( \alpha \) values to be run, and a vector of \( \theta \) values. In the Radon function the process of rotational transformation of the quadratic depending on \( \alpha \) and \( \theta \) is performed through a series of vectorized steps to decrease computational time.

Essentially the rotational transformation matrix presented in Equation 38 is applied to the quadratic equation in Equation 37. These equations and steps are presented as used with the code in Table 8. The original \( x \) values are computed through:

Equation 39

\[
x = x_{Min} + m \Delta x
\]

where \( M \) is the number of rows in the image, \( x_{Min} = -\frac{M}{2} \), \( m \) is the current row index, and \( \Delta x \) is the step size of \( x \) (equal to one when dealing with arrays/images.) Using these values of \( x \), the range \( y \) is generated through the use of the general non-shifted quadratic equation.

Equation 40

\[
y = \alpha x^2
\]

In Equation 40, \( \alpha \) is the scaling factor attributed to the expansion or compression of the quadratic curve. Once the values of \( x \) and \( y \) are computed, a rotational transformation can be performed to map their locations in accordance with a designated \( \theta \) through the following equations:
Equation 41

\[ x_r = x \cos \theta - y \sin \theta \]

Equation 42

\[ y_r = x \sin \theta + y \cos \theta \]

Where \( x_r \) and \( y_r \) define the new points of the rotated quadratic function.

The \( \alpha \) and \( \theta \) indices are then cycled through all of their possible values to test the corresponding quadratic rotations for inclusion in the input image. This is done through a stack checking integration/summation process for the individual pixels and their inclusion along the line of an \( \alpha \) and \( \theta \) rotated quadratic line. The pixels values are also tested for their inclusion within the boundaries of the image because of the use of indexing.

Once verified that the program is working within the image boundaries, the scaled and rotated quadratic is checked through summation for overlapping with curves in the original image. If the desired curve is present in the original image then this overlap will yield a positive contribution to the summation. Finally the summation value is recorded in a new two dimensional array known as the Radon domain image. For the Radon domain, the axis are \( \alpha \) and \( \theta \), respectively, and any curves detected through the summation is present at the designated parameter location as a peak value.

### Table 8: Generalized Quadratic Radon Transform Code

```
%% Quadratic Radon Transform
function [Q_radon] = RadonT_QQ(image, Delta_x, alpha, theta)
% Quadratic Radon transform for discrete Slant Stacking
ArrayDelta_x = 1;
```
ArrayDelta_y = 1;

[M,N] = size(image); % gives the size of the image
x_min = -M/2;
y_min = -N/2;

K = length(alpha); % setting size of K
H = length(theta); % setting size of H

Q_radon = zeros(K,H); % pre-allocation size of g_radon

% Calculation of x and y, equations
    %x = x_min + m*Delta_x;
    %y = alpha(k)*x^2;
x = x_min:Delta_x:M;
x2 = x.*x;
y = alpha'*x2;
sinxy = sin(theta);
cosxy = cos(theta);
cosxy = cosxy'*x;
sinxy = sinxy'*x;

for k = 1:K ; % for all values of alpha
    for h = 1:H; % For all values of theta
        sumQ = 0; % initializing the sum_1
        for m = 1:(M/Delta_x)
            % compute quadratic rotation transform
                % rotation transform equations
                % xr = x*cos(theta(h)) - y*sin(theta(h));
                % yr = x*sin(theta(h)) + y*cos(theta(h));
xr = xcosxy(h,m) - y(k,m)*sinxy(h);
yr = xsinxy(h,m) + y(k,m)*cosxy(h);
            if isreal(yr)% checking indexing of y
                mm = round(((xr - x_min)/ArrayDelta_x));
                n = round(((yr - y_min)/ArrayDelta_y));
                if (mm > 0) && (mm < M) && (n>0) && (n<N)
                    && image(mm,n) > 0
                    %image(mm,n); % increment the sum of n_1
            end
        end
    end
end
Table 8: Generalized Quadratic Radon Transform Code

4.3 Translation and Windowing

As noted earlier the curved hair detection technique is only applied to pixels that could possibly be hairs, i.e., those pixels having a masked value of 1.

Because the quadratic Radon transform works by considering rotation and curvature determination around the center, or “origin,” of an image, a windowing strategy was used to run the transform over a complete image. By shifting a window of 40x40 pixels across the entire image, and allowing for overlap from one pixel to the next, the generalized quadratic Radon transform is run on all possible sub-sections of the masked image. By doing this the image is easily decomposed for the application of the centered quadratic Radon transform and allows the function to detect if a quadratic curvature is present in the image on a sectioned basis. If a quadratic line is detected a peak will be produced in the Radon domain in the position of the appropriate $\alpha$ and $\theta$ values for the curve.

4.4 Multiple Peak Detection
Due to the nature of the quadratic Radon transform, when a quadratic curve is detected a peak is created at the corresponding parameter location in the Radon domain. In order to automatically remove the hair, the detection of a curve is not sufficient, but the actual location of the peak in the Radon domain must also be determined before removal can occur. To do this a peak detection algorithm needs to be used to automatically determine the presence of a quadratic peak. Alternatively, the peak can be detected through manual means.

### 4.4.1 Peak Detection Algorithm

Several existing Matlab functions for peak detection were tested to determine if they could be used for peak detection in the developed Radon Transform. The primary peak detector that was used was FastPeakFind [29]. According to the author this code was designed to analyze noisy 2D images and find the \(x,y\)-positions of peaks in the image to within 1 pixel accuracy. The peaks are assumed to be relatively sparse and the code works through a set of threshold or user defined filters. The function outputs location indicators of peaks and allows marks to be plotted over peaks. Additional code then tests whether this resulting plot is empty or not and therefore determines if there is a peak in the Radon domain indicating that there is a curve present at a given pixel.

<table>
<thead>
<tr>
<th>Table 9: FastPeakFind [29] Code used for peak detection of quadratic Radon Transform</th>
</tr>
</thead>
<tbody>
<tr>
<td>function cent=FastPeakFind(d, threshold, filt, edg, fid)</td>
</tr>
<tr>
<td>% % Analyzes noisy 2D images, finds x-y positions of peaks to 1 pixel accuracy</td>
</tr>
<tr>
<td>if (nargin &lt; 1)</td>
</tr>
<tr>
<td>d=uint16(conv2(reshape(single( 2^14*(rand(1,1024<em>1024)&gt;0.99995) ),[1024 1024]) ,fspecial('gaussian', 15,3),'same')+2^8</em>rand(1024));</td>
</tr>
<tr>
<td>imagesc(d);</td>
</tr>
</tbody>
</table>
if ndims(d)>2 %I added this in case one uses imread (JPG\PNG\...).
    d=uint16(rgb2gray(d));
end

if isfloat(d) %For the case the input image is double, casting to uint16 keeps enough
dynamic range while speeds up the code.
    if max(d(:))<=1
        d = uint16( d.*2^16./(max(d(:))));
    else
        d = uint16(d);
    end
end

if (nargin < 2)
    threshold = (max([min(max(d,[],1)) min(max(d,[],2))])) ;
end

if (nargin < 3)
    filt = (fspecial('gaussian', 7,1)); %if needed modify the filter according to the expected
peaks sizes
end

if (nargin < 4)
    edg =3;
end

if (nargin < 5)
    savefileflag = false;
else
    savefileflag = true;
end

%% Analyze image
if any(d(:)) ; %for the case of non-zero raw image
    d = medfilt2(d,[3,3]);
% apply threshold
   d=d.*uint16(d>threshold);

if any(d(:)) ; %for the case of the image is still non zero

% smooth image
   d=conv2(single(d),filt,'same') ;

% Apply again threshold (and change if needed according to SNR)
   d=d.*(d>0.9*threshold);

% peak find - using the local maxima approach - 1 pixel resolution
% d will be noisy on the edges, since no hits are expected there anyway we'll skip
'edge' pixels.

   [x y]=find(d(edg:size(d,1)-edg,edg:size(d,2)-edg));

% initialize peak find outputs
   cent=[];%
   x=x+edg-1;
   y=y+edg-1;
   for j=1:length(y)
      if (d(x(j),y(j))>=d(x(j)-1,y(j)-1 )) &&...
         (d(x(j),y(j))>d(x(j)-1,y(j))) &&...
         (d(x(j),y(j))>=d(x(j)-1,y(j)+1)) &&...
         (d(x(j),y(j))>d(x(j),y(j)-1)) && ... 
         (d(x(j),y(j))>d(x(j),y(j)+1)) && ... 
         (d(x(j),y(j))>=d(x(j)+1,y(j)-1)) && ...
         (d(x(j),y(j))>d(x(j)+1,y(j))) && ... 
         (d(x(j),y(j))>=d(x(j)+1,y(j)+1));

         cent = [cent ; x(j) ; y(j)];
         %cent(x(j),y(j))=cent(x(j),y(j))+1; % if a matrix output is desired
      end
   end
if savefileflag
    % previous version used dlmwrite, which can be slower than fprintf
    dlmwrite([filename '.txt'],[cent], 'append', ...
    'offset', 0, 'delimiter', '\t', 'newline', 'unix');+
    fprintf(fid, '%f ', cent(:));
    fprintf(fid, '\n');
end

else % in case image after threshold is all zeros
    cent=[];
    return
end

else % in case raw image is all zeros (dead event)
    cent=[];
    return
end

if (nargin < 1);
    colormap(bone);
    hold on;
    plot(cent(2:2:end),cent(1:2:end),'rs');
    hold off;
end

Table 9: FastPeakFind [29] Code used for peak detection of quadratic Radon Transform

Setting a threshold for the function was complicated due to the variety of values which can be produced in the Radon domain solely based on how many curves are present. Initially this threshold value was set to be approximately 500, this value determined through manual inspection of the Radon domain of both curves and noise pixels.

4.4.2 Manual Threshold Selection for Peak Detection
The FastPeakFind function is considered a naïve, automated, peak detection system and was not found to be effective or reliable enough for sole or automated use in peak detection. To avoid additional errors induced by possible missed peaks in the automation of peak detection, this function was only used after an average threshold was determined. It was determined that to get the best results of peak detection from the Radon transform for curve detection the threshold needed to be selected individually for each image being processed because of the effect of the maximum image intensity of the Radon transform based on the amount of hairs and their closeness. Peak recognition could have been done manually but would have been too time consuming. Automation of peak detection is clearly an important area for further research.

4.5 Curve Removal

Once peaks are detected and a pixel is determined to be part of a hair, the next step is to replace the value of the pixel with a value that is closely related to those skin pixels values surrounding it. In order to do this an interpolation method was selected and the value of the hair pixel replaced with the value determined from the interpolated value. For this thesis the interpolation method from the DullRazor® code was adapted and used for the interpolation replacement of the hair pixels.

In the DullRazor® work the initial hair mask is used as a guide for the replacement pixel operation. This is done by an identification method through which pixels are determined to belong to a hair and then replaced by the smoothed values of
nearby non-hair pixels. Prior to replacement, each pixel is checked for its inclusion in a thin long structure, otherwise it is rejected as noise. This is done by defining eight line segments drawn in eight different directions, up, down, left, right, and the four diagonals, starting from the focused pixel and drawn until the line segments reach a non-hair region. Of the eight lines coming from the centered pixel, one is determined, through counting, to be the longest included in the structure. The longest line must be greater than 50 pixels and the other lines must be shorter than 10 pixels, otherwise the center pixel is rejected. Once it is verified that a pixel is within the hair structure, bilinear interpolation is used to replace the center pixel in the original image by two nearby non-hair pixel values. Using the shortest line segment of the eight as a direction two pixel are selected from 11 pixels away and used for the interpolation algorithm. [8]

Letting the intensity value of the pixel being replaced be \( I(x, y) \) and the non-hair interpolation pixel intensities to be \( I_1(x_1, y_1) \) and \( I_2(x_2, y_2) \) the new intensity value, \( I_n(x, y) \) can be expressed as [8]

\[
I_n(x, y) = I_1(x_1, y_1) + I_2(x_2, y_2) = I_1(x_1, y_1) \frac{D(I(x, y), I_1(x_1, y_1))}{D(I(x, y), I_2(x_2, y_2))} + I_2(x_2, y_2) \frac{D(I(x, y), I_2(x_2, y_2))}{D(I(x, y), I_1(x_1, y_1))}
\]

where

\[
D(A(a, b), B(c, d)) = \sqrt{(c - a)^2 + (d - b)^2}.
\]
Although the concept is similar to, and the method of calculation the same as the code written (Table 10) and used for this thesis, this was done solely to demonstrate that the proper pixels were being selected for removal. For simplicity the direction for replacement is always along the left bottom to right top diagonal and two arbitrary pixels are chosen. Because of this it is not fully reliable and could produce errors in the image.

**Table 10: Interpolation Code Version 1 for Replacement of Image Pixels**

```
%%Follow interpolation from dull razor
ndisC = sqrt((i+10-i)^2 +(j+10-j)^2)/ ... 
sqrt((i+10-(i-10))^2 +(j+10-(j-10))^2);
disC = sqrt((i-10-i)^2 +(j-10-j)^2)/ ... 
sqrt((i+10-(i-10))^2 +(j+10-(j-10))^2);
interpValueR = O_r(i+10, j+10)*disC+...
o_r(i-10, j-10)*ndisC;
interpValueG = O_g(i+10, j+10)*disC+...
o_g(i-10, j-10)*ndisC;
interpValueB = O_b(i+10, j+10)*disC+...
o_b(i-10, j-10)*ndisC;
% interpolating the pixel value and placing into new image layers
nO_r(i,j) = interpValueR;
nO_g(i,j)= interpValueG;
nO_b(i,j) = interpValueB;
```

As the version 1 interpolation code in Table 10 could not fully remove the hairs in the validation images, an extension of the code was created for the testing in some images. Table 11 shows the modified interpolation code that simply replaces the eight surrounding pixels of the current location with the same interpolated value.

**Table 11: Interpolation Code Version 2 for Replacement of Image Pixels**

```
%%Follow interpolation from dull razor
```

---

84
\[
\text{ndisC} = \sqrt{(i+10-i)^2 + (j+10-j)^2)/ \ldots}
\]
\[
= \sqrt{(i+10-(i-10))^2 + (j+10-(j-10))^2);}
\]
\[
\text{disC} = \sqrt{(i-10-i)^2 + (j-10-j)^2)/ \ldots}
\]
\[
= \sqrt{(i+10-(i-10))^2 + (j+10-(j-10))^2);}
\]
\[
\text{interpValueR} = O_r(i+10, j+10) \times \text{disC} + \ldots
\]
\[
O_r(i-10, j-10) \times \text{ndisC};
\]
\[
\text{interpValueG} = O_g(i+10, j+10) \times \text{disC} + \ldots
\]
\[
O_g(i-10, j-10) \times \text{ndisC};
\]
\[
\text{interpValueB} = O_b(i+10, j+10) \times \text{disC} + \ldots
\]
\[
O_b(i-10, j-10) \times \text{ndisC};
\]

% interpolating the pixel value and placing into new image layers
\[
\text{nO}_r(i,j) = \text{interpValueR};
\]
\[
\text{nO}_g(i,j) = \text{interpValueG};
\]
\[
\text{nO}_b(i,j) = \text{interpValueB};
\]

%Replacement of eight surrounding pixels
\[
\text{nO}_r(i+1,j) = \text{interpValueR};
\]
\[
\text{nO}_g(i+1,j) = \text{interpValueG};
\]
\[
\text{nO}_b(i+1,j) = \text{interpValueB};
\]
\[
\text{nO}_r(i-1,j) = \text{interpValueR};
\]
\[
\text{nO}_g(i-1,j) = \text{interpValueG};
\]
\[
\text{nO}_b(i-1,j) = \text{interpValueB};
\]
\[
\text{nO}_r(i,j+1) = \text{interpValueR};
\]
\[
\text{nO}_g(i,j+1) = \text{interpValueG};
\]
\[
\text{nO}_b(i,j+1) = \text{interpValueB};
\]
\[
\text{nO}_r(i+1,j+1) = \text{interpValueR};
\]
\[
\text{nO}_g(i+1,j+1) = \text{interpValueG};
\]
\[
\text{nO}_b(i+1,j+1) = \text{interpValueB};
\]
\[
\text{nO}_r(i-1,j-1) = \text{interpValueR};
\]
\[
\text{nO}_g(i-1,j-1) = \text{interpValueG};
\]
\[
\text{nO}_b(i-1,j-1) = \text{interpValueB};
\]
\begin{equation}
\begin{aligned}
nO_r(i-1,j+1) &= \text{interpValueR}; \\
nO_g(i-1,j+1) &= \text{interpValueG}; \\
nO_b(i-1,j+1) &= \text{interpValueB}; \\
nO_r(i+1,j+1) &= \text{interpValueR}; \\
nO_g(i+1,j+1) &= \text{interpValueG}; \\
nO_b(i+1,j+1) &= \text{interpValueB};
\end{aligned}
\end{equation}

Table 11: Interpolation Code Version 2 for Replacement of Image Pixels

In changing this code the hope was to slightly extend the interpolation process outward to capture the borders of the hair that were being omitted from this process. Despite this procedure working in sections of images having synthetically added hairs, this code is not found to be suitable for actual hair images because it induced further errors by duplicating surrounding hair pixels.
5 Results

5.1 Validation

As with any process development, performance validation is extremely important. For the use of the quadratic Radon transform for curved hair detection and removal this was done in several. First the generation of an image with a synthetic quadratic curve was used for initial detection and attempted removal. Next the same method was used on multiple curves. Once the basic operation of the transform was validated, the implementation was adapted for testing on actual hair images. Initially an adapted actual image containing only a single hair was used for detection and removal testing. Once single hair detection and removal was shown to be successful, Adapted images consisting of multiple hairs were then used for testing. Finally the algorithm was run on an actual non-adapted image for automated hair detection and then removal. To finish testing the algorithm was run on an image containing hairs as well as a lesion. The effectiveness of the program was evaluated at every stage.

5.1.1 Single Quadratic Curve

In order to validate the implementation of the quadratic Radon transform the code was initially tested on a single image containing a single quadratic graph. Using MATLAB to create an array with a centered quadratic, the transform was run on the image as a whole (no windowing, only using the origin as the point of reference) to produce a peak in the Radon domain. The peak was then compared to the
corresponding quadratic parameters used, $\alpha$ and $\theta$. Below are some examples of the quadratic Radon transform run on various (rotated) quadratic curves generated by the code in Table 12.

**Table 12: Code for Generating Rotated Quadratic Images**

```matlab
function [image] = RotQuadIm(image,Delta_x, A, theta)
%% A function to graph the rotational quadratic in an array
%getting size of image
[M,N] = size(image);
ArrayDelta_x = 1;
ArrayDelta_y = 1;
x_min = -M/2;
y_min = -N/2;
for m = 1:(M/Delta_x) %Delta_x is the step of the function, not the array
    %compute x
    x = x_min + m*Delta_x;
    %compute y
    y = A*x^2;
    % rotation transform
    xr = x*cos(theta) - y*sin(theta);
    yr = x*sin(theta) + y*cos(theta);
    %compute n associated with y
    if isreal(yr)
        mm = round(((xr - x_min)/ArrayDelta_x));
        n = round(((yr - y_min)/ArrayDelta_y));
        if inside(mm,n, M, N)
            image(mm,n) = 1;
        end
    end
end
```

The image in Figure 22 shows the representative basic quadratic with $\alpha = 1$ and $\theta = 0$. This quadratic was processed with the quadratic Radon transform producing the
image in Figure 23, and, through parameter zoom, the image in Figure 23. As can be seen in the image, the peak associated with the parameters appears sharply at $\alpha = 1$ and $\theta = 0$, showing that the transform is properly detecting the quadratic in the image. To show a more general form of the Radon domain the transform was also done with a broader range of parameter values (see Figure 21). As can be seen there are three representative peaks in this image. These can be easily explained based on the extended nature of the quadratic in Figure 22. When the tails of the quadratic graph reach the end of the image they are close to straight and perpendicular to the $x$-axis. Because the theta and alpha values include the parameters associated with straight lines their inclusion in the Radon domain is apparent. However, if these tails were to be removed, meaning that the image would no longer contain straight lines with alpha approaching 0 (virtually straight), these peaks would not be present.

In the Radon domain in Figure 21 it can be seen that there are two peaks at the top and one in the center. The centered peak, at $\alpha = 1$ and $\theta = 0$, clearly outlines the original parameters of the quadratic in Figure 22, as expected. The two peaks at the top are from the tails of the quadratic and its perpendicular nature. These peaks are seen at the top of the image and correspond to the parameters of the tail lines in the quadratic image.
Figure 20: Quadratic curve with $\alpha=1$ and $\theta = 0$

Figure 21: Zoomed out quadratic Radon transform of image in Figure 20

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>VALUE</th>
<th>PARAMETER</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Delta x$</td>
<td>1</td>
<td>$\Delta \alpha$</td>
<td>0.01</td>
</tr>
<tr>
<td>$\Delta y$</td>
<td>1</td>
<td>$\Delta \theta$</td>
<td>0.01</td>
</tr>
<tr>
<td>$x_{\min}$</td>
<td>-100</td>
<td>$\alpha_{\min}$</td>
<td>0</td>
</tr>
<tr>
<td>$y_{\min}$</td>
<td>-100</td>
<td>$\theta_{\min}$</td>
<td>-\pi</td>
</tr>
<tr>
<td>$x_{\max}$</td>
<td>100</td>
<td>$\alpha_{\max}$</td>
<td>2</td>
</tr>
<tr>
<td>$y_{\max}$</td>
<td>100</td>
<td>$\theta_{\max}$</td>
<td>\pi</td>
</tr>
</tbody>
</table>

Table 13: Parameters for Figure 20 and Figure 21

When the parameter values are zoomed in the image it can be seen that the peak is very sharp. Due to the parameter resolution of $\theta$ the peak along axis is extremely sharp whereas the peak for the parameter $\alpha$ is slightly broader. When examined through intensity values, the maximum peak for $\alpha$ is directly associated with the parameter $\alpha = 1$. 
Figure 22: Quadratic curve with $\alpha = 1$ and $\theta = 0$

![Quadratic curve with $\alpha = 1$ and $\theta = 0$](image)

Figure 23: Zoomed in quadratic Radon transform of image in Figure 22

![Zoomed quadratic Radon transform](image)

<table>
<thead>
<tr>
<th>IMAGE DOMAIN</th>
<th>RADON DOMAIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>Value</td>
</tr>
<tr>
<td>$\Delta x$</td>
<td>1</td>
</tr>
<tr>
<td>$\Delta y$</td>
<td>1</td>
</tr>
<tr>
<td>$x_{\text{min}}$</td>
<td>-100</td>
</tr>
<tr>
<td>$y_{\text{min}}$</td>
<td>-100</td>
</tr>
<tr>
<td>$x_{\text{max}}$</td>
<td>100</td>
</tr>
<tr>
<td>$y_{\text{max}}$</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 14: Parameters for Figure 22 and Figure 23

Similarly, the rotated quadratic image in Figure 24 with $\alpha = 0.5$ and $\theta = \frac{\pi}{2}$ and its corresponding zoomed quadratic Radon transform shown in Figure 25 also has a broad peak; however, the peak is sharper than that in Figure 23. Figure 24 and Figure 25 show that even when the quadratic is rotated the quadratic Radon transform properly identifies the parameters of the curve.
Figure 24: Quadratic curve with $\alpha = 0.5$ and $\theta = \pi/2$

Figure 25: Zoomed in quadratic Radon transform of Error! Reference source not found.

<table>
<thead>
<tr>
<th>IMAGE DOMAIN</th>
<th>RADON DOMAIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>Value</td>
</tr>
<tr>
<td>$\Delta x$</td>
<td>1</td>
</tr>
<tr>
<td>$\Delta y$</td>
<td>1</td>
</tr>
<tr>
<td>$x_{\min}$</td>
<td>-100</td>
</tr>
<tr>
<td>$y_{\min}$</td>
<td>-100</td>
</tr>
<tr>
<td>$x_{\max}$</td>
<td>100</td>
</tr>
<tr>
<td>$y_{\max}$</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 15: Parameters for images in Figure 24 and Figure 25

While this broad peak is found in many of the transformed images, this is not always the case. This broadness appears to be associated with a value dependent sensitivity of the $\theta$.

This becomes more evident in the following figures where the image when a quadratic curve with parameters $\alpha = 0.1$ and $\theta = \pi/4$ is transformed by the quadratic Radon transform. The resulting Radon domain has a clear singular peak at the parameter values of $\alpha = 0.1$ and $\theta = \pi/4$, exactly corresponding to those of the original
image. As can be seen, the sharpness of this curve is much clearer than that of the previous transforms and leaves little uncertainty as to location of the quadratic.

![Figure 26: Quadratic curve with $\alpha=0.1$ and $\theta = \pi/4$](image)

![Figure 27: Quadratic Radon transform of image in Figure 26](image)

<table>
<thead>
<tr>
<th>IMAGE DOMAIN</th>
<th>RADON DOMAIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>Value</td>
</tr>
<tr>
<td>$\Delta x$</td>
<td>1</td>
</tr>
<tr>
<td>$\Delta y$</td>
<td>1</td>
</tr>
<tr>
<td>$x_{\text{min}}$</td>
<td>-100</td>
</tr>
<tr>
<td>$y_{\text{min}}$</td>
<td>-100</td>
</tr>
<tr>
<td>$x_{\text{max}}$</td>
<td>100</td>
</tr>
<tr>
<td>$y_{\text{max}}$</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 16: Parameters of images in Figure 26 and Figure 27

In order to get a clearer view of the peak, a zoomed parameter transform was taken of the image in Figure 26 and is displayed below in Figure 28.
Figure 28 shows that the peak of this quadratic is extremely sharp as compared to those of the previous transforms.

It is evident that the quadratic Radon transform is effective at identifying and displaying the parameters of the quadratics in the generated images. With no other graph present in the original image, the peaks of the parameters are sharp and high in value. Despite some broadness seemingly associated with the sensitivity of $\theta$, the peaks are easily identifiable as well as the true parameter locations.

### 5.1.2 Multiple Quadratic Curves

It has been demonstrated above that for various orientations of a single curve the quadratic Radon transform will produce a corresponding parameter peak in the Radon domain at the correct parameter locations. Because in practice there is often more than one hair in an image, and there may be more than one in the sampling window, the transform must be able to handle multiple curves and produce

**Table 17: Parameters of images in Figure 26 and Figure 28**

<table>
<thead>
<tr>
<th>IMAGE DOMAIN</th>
<th>RADON DOMAIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>Value</td>
</tr>
<tr>
<td>$\Delta x$</td>
<td>1</td>
</tr>
<tr>
<td>$\Delta y$</td>
<td>1</td>
</tr>
<tr>
<td>$x_{\min}$</td>
<td>-100</td>
</tr>
<tr>
<td>$y_{\min}$</td>
<td>-100</td>
</tr>
<tr>
<td>$x_{\max}$</td>
<td>100</td>
</tr>
<tr>
<td>$y_{\max}$</td>
<td>100</td>
</tr>
</tbody>
</table>
corresponding parameter peaks. Thus, the transform was also tested on synthetic images produced with multiple quadratic curves.

Using multiple curves generated within a single image allowed for testing the Radon transform on multi-quadratic curve images. The following examples show that effectiveness of the quadratic Radon transform and its ability to detect multiple curves in various orientations. These examples contain two or more superimposed, and possibly overlapping, rotated and scaled quadratics with varying values of \( \alpha \) and \( \theta \).

The first example tested contained quadratics parameterized with \( (\alpha_1 = 0.1, \theta_1 = \frac{\pi}{4}) \) and \( (\alpha_2 = 0.5, \theta_2 = \frac{\pi}{2}) \), as shown in Figure 29. Despite the fact that they overlapped the Radon transform was able to produce two peaks relating to the parameter values of the curves in the image. Because the curves were identical to those used in previous single curve examples it is easy to compare the two individually produced Radon images, Figure 24 and Figure 27, with the multiple Radon image in Figure 30. As can be seen, the peaks are practically identical when compared to its corresponding one in the multiple Radon transform. This shows that the presence of another curve, while dimming the overall intensity, still produces a prominent peak in the Radon domain.
Expanding these tests, three curves were then generated in one image and the transform taken. As seen in Figure 31, three separate quadratics were drawn with parameter values given by $(\alpha_1 = 0.1, \theta_1 = \frac{\pi}{4}), (\alpha_2 = 1, \theta_2 = \pi)$, and $(\alpha_3 = 0.5, \theta_3 = \frac{\pi}{2})$. When the corresponding transform is taken the peaks appear as they did with their individual counter parts (as in the previous example, with the position exception of the altered $\theta$ for the second quadratic).
In order to test a full range of quadratics diagonally oriented functions were created to ensure that the transform work for the full range of $\theta$. Using quadratics generated at $(\alpha_1 = 0.1, \theta_1 = \frac{3\pi}{4})$, $(\alpha_2 = 0.5, \theta_2 = \frac{\pi}{4})$, $(\alpha_3 = 0.1, \theta_3 = \frac{5\pi}{4})$, and $(\alpha_4 = 1, \theta_4 = \frac{7\pi}{4})$ the quadratic Radon was taken and is shown in Figure 34.
Figure 33: Multi-quadratic curve image with $\alpha=0.1, 0.5, 0.1, 1$ and $\theta = 3\times\pi/4, \pi/4, 5\times\pi/4, 7\times\pi/4$

Figure 34: Multi-Quadratic Radon transform of image in Figure 33

<table>
<thead>
<tr>
<th>IMAGE DOMAIN</th>
<th>RADON DOMAIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>Value</td>
</tr>
<tr>
<td>$\Delta x$</td>
<td>1</td>
</tr>
<tr>
<td>$\Delta y$</td>
<td>1</td>
</tr>
<tr>
<td>$x_{\min}$</td>
<td>-100</td>
</tr>
<tr>
<td>$y_{\min}$</td>
<td>-100</td>
</tr>
<tr>
<td>$x_{\max}$</td>
<td>100</td>
</tr>
<tr>
<td>$y_{\max}$</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 20: Parameters for images in Figure 33 and Figure 34

As seen in Figure 34, the image of the quadratic Radon transform, four characteristic peaks are visible for the representation of the parameters of the curves. Interestingly two of the peaks are very broad while the other two are sharp, although all peaks describe their intended parameters.

The final example for the multiple quadratic curve verification testing is an image where only the $\alpha$ parameter is altered. Letting $\theta$ remain constant allowed for the variation in $\alpha$ to be fully examined.
As can be seen by the representative peaks in Figure 36, all parameters of the curves in Figure 35 are accounted for. There also appears to be some peaks that have been generated but have no corresponding curve in the image. These peaks are generated from overlapping values created from the Radon transform. Each peak has a symbolic “X” pattern that appears around the peak in every quadratic Radon transform image generated up to this point. These peaks are simply the overlap of the tail ends of the transform as it builds up intensity towards the correct parameter peak. While previously this has been avoided through the generation of curves with differing $\theta$
parameters it is worthwhile to note this phenomenon and recognize that these aliased peaks can be produced as false indicators.

As is evident by the figures presented in this section, the quadratic Radon transform is effective at identifying and displaying the parameters of multiple quadratics in a generated image of the Radon parameter domain. Despite the presence of other curves, the peaks of the curve parameters are still present and clearly identifiable. Despite some broadness and the possibility of aliasing the peaks are easily located and discernible.

5.1.3 Actual Hair Images

In order for the method developed in this thesis to be used for hair removal as a precursor to automated skin lesion detection and diagnosis, its effectiveness on actual hair images has to be examined. In this section various different hair images are tested and their effectiveness demonstrated through the resulting output images. It should be noted that the following section uses the shifted window technique with a window size of 40x40. Thus all transforms done on the image will not be shown but a select few generated on hairs will be provided to show curve detection. All of the following transform code has been run with the parameter values as provided in Table 22.
### Table 22: Parameters used for image transforms based in actual images

<table>
<thead>
<tr>
<th>IMAGE DOMAIN</th>
<th>RADON DOMAIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>Value</td>
</tr>
<tr>
<td>Δx</td>
<td>1</td>
</tr>
<tr>
<td>Δy</td>
<td>1</td>
</tr>
<tr>
<td>x_{min}</td>
<td>-20</td>
</tr>
<tr>
<td>y_{min}</td>
<td>-20</td>
</tr>
<tr>
<td>x_{max}</td>
<td>20</td>
</tr>
<tr>
<td>y_{max}</td>
<td>20</td>
</tr>
</tbody>
</table>

#### 5.1.3.1 Single Hair

To begin testing the effectiveness of the quadratic Radon transform on practical images a hair image was manually cleared of all hairs except for one hair (extracted from an actual skin image). This image, shown in Figure 37, was run through the transform code, processed with the shifting window method, and then produced the following results of the new image in Figure 42. Shown in the following image examples is the process and transformation done automatically by the code.

![Figure 37: Original single hair skin image, .gif image](image1)

![Figure 38: Hair selection image for designated Radon transform image in Figure 39 and Figure 40](image2)
Starting with the original image in Figure 37, displayed in MATLAB as shown in Figure 41, hair masking is first used on the image to create a binary mask. This image is then run through the transform code using the windowing method to determine if a hair is present. Once determined through peak detection the image pixel undergoes the version 2 interpolation process defined in Table 11. Figure 42 shows the result when the transform is run on all possible hair pixels. As can be seen, the replacement of the
hair pixels worked relatively well and the hair is now practically indistinguishable from the image background.

5.1.3.2 Multiple Hairs without Lesion

To test the code on multiple hair images two processes were used. The first process was simply an extension of the single hair verification method in which the single hair in the image was duplicated, rotated, and added to a separate part of the image to test the transform on multiple curves in the same image. Next an image with no skin lesion but multiple dark hairs was used to show the effectiveness of the hair masking and quadratic Radon detection on an actual non-altered hair image.

5.1.3.2.1 Synthetically Added Hair Image: Continuation of Single Hair

To ensure that the quadratic Radon transform can handle multiple hairs, verification tests were run on an image similar to that of used in the 5.1.3.1 Single Hair section. Duplicating the single hair in various places within the image, as well as rotating the hair, generated several multi-hair synthetic images for testing. These images, tests and images provided below consist of a two hair image, three hair image with rotated hair, and a multi-hair with various rotation image.

The first example of two hairs in the image produced exemplary results. The hair masking image is the first one to be generated. Because of the simplicity and synthetic nature of this image, the hair masking captures the whole of both hairs allowing for swift detection and initial error elimination. Figure 45, Figure 46, and Figure 47 show the effectiveness of both the quadratic Radon transform and the peak detection on the
windowed section. Finally by replacing the pixel values, as is seen in Figure 48, virtually no hair pixels remain in the final image meaning the program completely removed the hairs from the image.
Figure 43: Synthetic hair image with two hairs

Figure 44: Initial Hair masking of image in Figure 43

Figure 45: Windowed image containing hair to be quadratic Radon transformed

Figure 46: Quadratic Radon transform of image in Figure 43

Figure 47: Peak detection, threshold of 600, quadratic Radon transform of image in Figure 45

Figure 48: Hair Removal results of original image after full thesis code is performed on image in Figure 43, uses Interpolation Version 2
Increasing the amount of hairs by one in the image and rotating one of them as shown in Figure 49, allows for testing for the detection and location of rotational hairs as well. As can be seen the effectiveness of this process was equal to that of the two hair image allowing for virtually complete hair removal.
Figure 49: Synthetic hair image with three hairs, one rotated

Figure 50: Initial Hair masking of image in Figure 49

Figure 51: Windowed image containing hair to be quadratic Radon transformed

Figure 52: Quadratic Radon transform of image in Figure 49

Figure 53: Peak detection, threshold of 600, quadratic Radon transform of image in Figure 51

Figure 54: Hair Removal results of original image after full thesis code is performed on image in Figure 49, uses Interpolation Version 2
To fully test the code on the synthetic images a variety of hairs and orientations were added to the three hair image to allow for maximum hair removal. Masking of the hair image once again went flawlessly which allowed the most effective hair detection and removal. As can be seen in the resulting hair image virtually no hairs remain in the processed image.
Figure 55: Synthetic hair image with multiple rotated hairs

Figure 56: Initial Hair masking of image in Figure 55

Figure 57: Windowed image containing hair to be quadratic Radon transformed

Figure 58: Quadratic Radon transform of image in Figure 55

Figure 59: Peak detection, threshold of 1000, quadratic Radon transform of image in Figure 55

Figure 60: Hair Removal results of original image after full thesis code is performed on image in Figure 55, uses Interpolation Version 2
5.1.3.2.2 Non-Lesion Hair Image

The test image, SkinHair001.jpg [30], shown in Figure 61, used for this section was found through the use of Google image search. As the image does not have thick or dark hairs required for the DullRazor® image mask, the detection and replacement is lacking in accuracy simply because of the initial hair masking. The penumbra effect is also coming in to play with the masking of this image as well as with the hair replacement. It should be noted that the image being used in the quadratic Radon transform is solely based on the initial image mask and its effectiveness can be no better than the accuracy of the image mask.

Running the code on a non-lesion hair image was important to show how it works while avoiding any errors which can be generated by the discoloration of a lesion in the image. Because the selected image is not representative of the type of hairs meant to be selected by the hair masking, some liberty in the interpretation of the results can be allowed. As can be seen in Figure 63, Figure 64, and Figure 65 the program was effective in recognizing the curvature in the image and determining the peaks through the use of the quadratic Radon transform.
Running the code over the full image, Figure 61, was ineffective and the final resulting image, Figure 66, was cleared of hairs as expected. This lead to the testing of
the smaller sections of the image: Section 1, Figure 67, and Section 2, Figure 73, to attempt to detect the issue as well as generate more successful hair removal.

Section 1 was taken from a densely populated hair part of the image. The full code was used to analyze this section and the results are presented below:
Figure 67: Section 1 of Non-Lesion Hair Image

Figure 68: Initial Hair Mask of Section 1 of Non-Lesion Hair Image

Figure 69: Windowed image containing hair to be quadratic Radon transformed

Figure 70: Quadratic Radon transform of Windowed Image

Figure 71: Peak detected quadratic Radon transform of windowed image, threshold of 800

Figure 72: Final interpolated image of Figure 67
Figure 69 was chosen for two reasons. The first reason is that it effectively shows that despite the fact that the hair mask is unable to present the whole curved hair, the quadratic Radon transform is able to pick out enough information to recognize that these pixels are not noise. The transform was able to replace the pixel which was masked as hairs despite their discontinuity. The second reason is that it also shows a short coming in the program. Apart from the fact that the program is able to identify the curvature despite the discontinuity, it is unable to do so for the missing pixels. The discontinuous pixels which are likely part of the hair are never run through the program and accounted for in the hair removal.

Notwithstanding this slight flaw the hair removal performs marginally well. The fact that the hair masking image is missing bordering pixels of the hair could be fixed with the alteration of that part of the code. The main point to note is that the quadratic Radon transform successfully detected curvature of hair pixel.

Section 2, Figure 73, was generated and run to compare if the density of the hair in the image affected the detection and removal of the hairs in the image. While the resulting image, Figure 78, is not perfect the hair removal process is definitely improved as compared to that of Figure 72.

As demonstrated in these results, the quadratic Radon transform appears to be working effectively; however the algorithms for image masking and hair removal appear to be ineffective. Using the method on real images will always produce problems due to the uncertainty of the hair locations and borders. In this instance the hair mask misses
the border completely leaving it in the image without the chance for interpolation replacement.
Figure 73: Section 2 of Non-Lesion Hair Image

Figure 74: Initial Hair Mask of Section 2 of Non-Lesion Hair Image

Figure 75: Windowed image containing hair to be quadratic Radon transformed

Figure 76: Quadratic Radon transform of Windowed Image

Figure 77: Peak detected quadratic Radon transform of windowed image, threshold of 800

Figure 78: Final interpolated image of Figure 73
To attempt to ease this problem, the image in Figure 79 was rerun through the code using the version 2 interpolation code (Table 11) and an image dilation function on the hair masked image in Figure 80. When a binary image is dilated a structuring element is used to perform a binary image dilation or expansion on the objects in the image. In this case the built in Matlab functionimdilate [31] was used for image dilation. In addition, the built in function, strel [32], for creating a morphological structuring element function, was used. The structuring element parameters were set to strel('disk', 5) meaning that the element will be an oval with a pixel maximum radius of 5. Using this structuring element and dilation on the initial hair mask, the image in Figure 81 was produced. The rest of the code was then run on this image along with the version 2 interpolation and the resulting image can be seen in Figure 82.
Despite that the hair borders can still be seen, their intensity is lessened.

However the interpolation method being used remains highly simplified and requires significant improvement. Clearly hair masking of the image is essential to the success of the program’s effectiveness. Without an excellent hair mask many hairs will never be fully detected and removed.

5.1.3.2 Multiple Hairs with Lesion
The final step of the verification process was the use of the hair mask, quadratic Radon transform, and hair removal interpolation methods on an image with both hair and lesion components. Using the primary example image [8] from the DullRazor® paper, the method developed in this thesis was tested on a hair image containing both significant amounts of hair and a lesion.

The following examples show the process that occurred when running the code over the hairy skin/lesion image. Although the resulting hair removal image is not as good as expected there are significantly less hairs present in the final image as compared to the original image. These errors can partially be attributed to the noisy and inaccurate hair masking operation performed on the original image.

Despite this the representative quadratic Radon transform image, Figure 86, presented appropriate peaks for curvature detection. The use of the peak detection algorithm placed red “x”s in the appropriate areas of the observed peaks, Figure 87, and allows for pixel values to be interpolated and removed if needed.

The quadratic Radon transforms for this image all appeared very noisy. The use of a smaller sliding window may have helped eliminate the noise by reducing the amount of curves present. However, in reducing the window size this also reduced the amount of the curve which is present in the window. Doing this appeared to render the quadratic Radon transform obsolete due to the zoomed nature of the window and was therefore not used.
Figure 83: Skin Lesion Hair Image

Figure 84: Skin Lesion Hair Image Initial Hair Mask

Figure 85: Windowed image containing hair to be quadratic Radon transformed

Figure 86: Quadratic Radon transform of Windowed Image

Figure 87: Peak detected quadratic Radon transform of windowed image, threshold of 1500

Figure 88: Final Interpolated image of Figure 83
6 Conclusions and Future Work

6.1 Conclusion

In this thesis hair detection and removal is done through the use of the generalized Radon transform to allow for a broader class of hairs that can be detected and removed as compared to the use of the normal Radon transform. Though it worked effectively on general quadratic curves and when used for detection and location of hairs in actual hair images the results of the final images were not as expected. The main part of this is due to the fact that the image masking that was used is not effective enough to allow the full detection of hairs in the image. Parts of hairs are left out of the initial hair mask and this error permeates throughout the rest of the program.

The transform code is also very processor intensive and therefore slow to run. This is probably attributed to the lack of optimization in the interpolation and peak detection methods. The actual quadratic Radon transforms itself takes 1.7 second to run on a window of size 200x200 pixels using an Intel(R) Core(TM) i7-3610QM CPU @ 2.30GHz 2.30 GHz, 16.0 GB RAM, with a NVIDIA GeForce GT 650M Version 306.97, on Windows 7 64-bit operating system. However, due to the necessity of having to run the program on each window of every potential hair pixel, optimization must be done before it can be run in a timely and effective manner.

Overall the quadratic Radon transform works nicely in being able to detect curves in the image and ignore the majority of image spots which are considered noise.
Despite a slight inability to remove hairs in actual images there is still hope for the method to have improved performance. As a result of the inaccuracy of the peak detector, the unreliability of the thresholding value, and the simplistic hair masking a significant number of hair detection and location errors have occurred using the method developed in this thesis. The method for hair masking, peak detection, and hair removal are all simplistic approaches to deal with their respective problems and this appears to have had detrimental effects on the overall results. The actual quadratic Radon transform is fast and efficient. It is believed that if this function could be incorporated into a more effective and faster algorithm for peak detection and hair removal it would be an effective tool for hair removal in skin lesion images.
6.2 Future Work

As the primary focus of this thesis was on the development and use of the quadratic Radon transform there are several areas of the thesis that could be expanded and improved.

6.2.1 Hair Masking

Advances in hair masking will ultimately make automated hair detection and removal more effective because hair masking represent the initial step in these algorithms and vital to their success. Hair masking in this thesis is done by the same method used in the DullRazor® [8] application. As noted earlier in this thesis - and by [9] - this hair mask is only adapted for thick dark hairs. In response to this, the authors of E-Shaver [9] presented an algorithm designed to remove both dark and light hairs. Although their program uses different methods for detecting hair pixels by allowing for light-colored hair detection [9] as well, improvements can still be made. Despite both program [8] [9] showing success with their methods of hair removal errors still occur. DullRazor® [8] works with thick dark hairs and can effectively remove and replace these hairs with interpolated values; however, the program can still leave artifacts. Similarly, E-shaver [9] approaches hair removal by targeting both light and dark hairs but can also leave behind hairs and artifacts [9]. Both programs still have trouble with the similarities in color between hairs and lesions and therefore have trouble detecting the hairs covering the lesion. When this happens hairs can be missed completely or false indicators can be created on the skin lesion. A better hair masking algorithm would
clearly perform better and result in greater effectiveness for the detection and removal. A possible route to consider would be to use a method similar to the curved Radon transform to help distinguish curves in the image and allow for detection in this manner.

6.2.2 Peak Detector

A automated peak detector is needed to be able to handle single as well as multiple peaks for identification of the locations and centers of the peaks. This is essential to determining the curve parameters of the hairs detected. Any hair removal that will be done is dependent on a reliable and effective peak detector to increase the value of the program. This is not a trivial matter and considerable research has already been done to develop useful peak detection algorithms. Peak detection and location is imperative for full automation of this process. The Matlab algorithms testing in the course of this research were all found to be inadequate for automation. Thus, further research needs to be done to develop useful methods to be used in this application.

6.2.3 Interpolation to Replace Hair

While hair removal methods were not the main focus of this thesis, more advanced methods of hair replacement should be developed to increase the effectiveness of the developed method. With mathematical advances in image processing and interpolation algorithms, replacement of hairs once their position has been located can possibly be done successfully and in a variety of different ways. Using a method that will also ensure that pixels being used for interpolation are not part of
hairs themselves would increase the effectiveness of hair removal and decrease the likelihood of artifact replacements. It is extremely important to develop a process which will avoid leaving behind shadows and avoid the penumbra effect. When done effectively it is expected that the resulting output image will be extremely useful for skin cancer detection and diagnosis without fear of induced errors from occluding hairs.

6.2.4 Correlation in Radon Domain

Along with peak detection a method for correlation in the Radon domain could be helpful to increase the effectiveness of the process. Currently hair pixels are determined on a pixel by pixel basis. A more effective way of determining hair pixels and whether they are connected (all one hair) could increase the success of hair removal as well as speed the image processing. This might be done in many different ways. One possible way to achieve this would be though the correlation of peaks in the Radon domain with a known peak template, allowing for the connection of the curves relating to the hair pixels. Another possibility would be to “connect” the hair pixels in the image. This would allow for less computation of the Radon transform as well as ease the removal process through the connection of the pixels in the same hair.

6.2.5 Graphical Processor Unit (GPU) Speedup

Although a naïve MATLAB-based GPU speedup was attempted for the Quadratic Generalized Radon Transform, it was found not to be effective. However, because of the repetitive and parallel computations done by this program it is believed that, if done in a native manner, the curved hair detection and removal algorithm can be run - with
the aid of the GPU - to significantly improve runtime performance. Allowing for proper computer specifications this process could potentially be run in real-time thereby allowing for its incorporation and development into portable applications and/or clinical use as well as an addition to an automated lesion detection and diagnosis program.

6.2.6 Two Stage Processing

In order to decrease computation time and improve the hair detection results, an alternative method suggested by this research is using a two stage process for hair removal. Linear hair removal is generally quick and effective at removing straight hairs from an image. This primarily stems from the linear problem being two dimensional in the Radon parameter space. If linear hair removal were done as a preprocessing technique for curved hair detection this would decrease the $\alpha$ values necessary for coverage allowing for denser sampling as well as faster computation for curved hair detection and removal. Obviously this method would be limited by the hair masking used as well as the interpolation method but for hair detection and determination it has the potential to speed up the process and alleviate computation time significantly.
Bibliography


Review: 75 Years of Radon Transform," [Online]. Available:


[30] [Online]. Available:
http://www.davidtorno.com/MacTex_Uploads/SkinHair001.jpg.


[34] G. Williams, M.D and M. Katcher, M.D. Ph.D., "Nomenclature of Skin Lesions," Primary Care Dermatology Module, [Online]. Available: