

ABSTRACT

Title of dissertation: MODELING AND ANALYSIS OF WRINKLES
ON AGING HUMAN FACES

Nazr e Batool, Doctor of Philosophy, 2013

Dissertation directed by: Professor Rama Chellappa
Department of Electrical and Computer Engineering

The analysis and modeling of aging human faces has been extensively studied in the past decade. Most of this work is based on matching learning techniques focused on appearance of faces at different ages incorporating facial features such as face shape/geometry and patch-based texture features. However, we do not find much work done on the analysis of facial wrinkles in general and specific to a person. The goal of this dissertation is to analyze and model facial wrinkles for different applications.

Facial wrinkles are challenging low-level image features to analyze. In general, skin texture has drastically varying appearance due to its characteristic physical properties. A skin patch looks very different when viewed or illuminated from different angles. This makes subtle skin features like facial wrinkles difficult to be detected in images acquired in uncontrolled imaging settings. In this dissertation, we examine the image properties of wrinkles i.e. intensity gradients and geometric properties and use them for several applications including low-level image processing for automatic detection/localization of wrinkles, soft biometrics and removal of

wrinkles using digital inpainting.

First, we present results of detection/localization of wrinkles in images using Marked Point Process (MPP). Wrinkles are modeled as sequences of line segments in a Bayesian framework which incorporates a prior probability model based on the likely geometric properties of wrinkles and a data likelihood term based on image intensity gradients. Wrinkles are localized by sampling the posterior probability using a Reversible Jump Markov Chain Monte Carlo (RJCMCMC) algorithm. We also present an evaluation algorithm to quantitatively evaluate the detection and false alarm rate of our algorithm and conduct experiments with images taken in uncontrolled settings.

The MPP model, despite its promising localization results, requires a large number of iterations in the RJCMCMC algorithm to reach global minimum resulting in considerable computation time. This motivated us to adopt a deterministic approach based on image morphology for fast localization of facial wrinkles. We propose image features based on Gabor filter banks to highlight subtle curvilinear discontinuities in skin texture caused by wrinkles. Then, image morphology is used to incorporate geometric constraints to localize curvilinear shapes of wrinkles at image sites of large Gabor filter responses. We conduct experiments on two sets of low and high resolution images to demonstrate faster and visually better localization results as compared to those obtained by MPP modeling.

As a next application, we investigate the user-drawn and automatically detected wrinkles as a pattern for their discriminative power as a soft biometrics to recognize subjects from their wrinkle patterns only. A set of facial wrinkles from

an image is treated as a curve pattern and used for subject recognition. Given the wrinkle patterns from a query and gallery images, several distance measures are calculated between the two patterns to quantify the similarity between them. This is done by finding the possible correspondences between curves from the two patterns using a simple bipartite graph matching algorithm. Then several metrics are used to calculate the similarity between the two wrinkle patterns. These metrics are based on Hausdorff distance and curve-to-curve correspondences. We conduct experiments on data sets of both hand drawn and automatically detected wrinkles.

Finally, we apply digital inpainting to automatically remove wrinkles from facial images. Digital image inpainting refers to filling in the holes of arbitrary shapes in images so that they seem to be part of the original image. The inpainting methods target either the structure or the texture of an image or both. There are two limitations of existing inpainting methods for the removal of wrinkles. First, the differences in the attributes of structure and texture requires different inpainting methods. Facial wrinkles do not fall strictly under the category of structure or texture and can be considered as some where in between. Second, almost all of the image inpainting techniques are supervised i.e. the area/gap to be filled is provided by user interaction and the algorithms attempt to find the suitable image portion automatically. We present an unsupervised image inpainting method where facial regions with wrinkles are detected automatically using their characteristic intensity gradients and removed by painting the regions by the surrounding skin texture.

MODELING AND ANALYSIS OF WRINKLES ON AGING
HUMAN FACES

by

Nazr e Batool

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2013

Advisory Committee:
Professor Rama Chellappa, Chair/Advisor
Professor Min Wu
Professor Ramani Duraiswami
Professor Amitabh Varshney
Dr. Jonathan Phillips

© Copyright by
Nazr e Batool
2013

Dedication

To dear Autumn, Marc, Anne, Sima, Lawrence, Stephen and Raymond. This dissertation would not have been possible without their help and care. They have been supportive like a family when I was geographically distant from my own family while pursuing my doctoral studies.

Acknowledgments

First and foremost, I would like to thank deeply my supervisor Prof. Rama Chellappa for his guidance and supervision throughout my PhD studies. From his lectures, personal discussions, group meetings and even every day conversations, I was able to learn a lot about theory, research and applications in computer vision. As I realize now, interacting with him was a continuous fruitful process over the years for which I am very grateful for. He gave me the freedom and space to pursue new research ideas while always providing me with new directions and encouragement. I can recall numerous occasions when I was nervous about some problem I faced, had discussion with him about it and left his office very relieved afterwards. Despite his strict schedule, he always made himself available whenever I needed advice. I came to realize and be grateful for this even more after I had experiences of interacting with other people with busy schedules. It was a very stimulating experience working under his supervision due to the friendly, open and intellectually stimulating environment of his research group. I am grateful to him for providing me with high-end computational and lab facilities and comfortable working office space since the very beginning. I want to thank him for helping me at several times regarding my funding or financial situations and for letting me attend conferences inside and outside the country. Above all, I have been impressed by his generosity, humility and kindness in every way possible, the qualities I will always aspire to have as a person.

It is an honor to have Prof. Min Wu, Prof. Ramani Duraiswami, Prof.

Amitabh Varshney, Prof. Larry Davis and Dr. Jonathan Phillips from NIST in my dissertation committee. I am thankful to them for serving on my committee and providing insightful suggestions to improve this dissertation. Because of some certain circumstances, I had to reschedule my dissertation exam a couple of times and it was very kind of them to accommodate me in their schedules. Specially, I want to thank Prof. Min Wu for her mentorship over the years as an instructor of the graduate courses I took under her and as an advisor of the Women in ECE community.

I want to thank Prof. Prakash Narayan of ECE department and Prof. Eric Slud of Statistics department for taking time out to help me and answer the questions I had in their courses.

I want to thank Prof. Andre Tits of ECE department for his availability and valuable candid discussions regarding any issues I had. He keeps his office doors open to help any student facing any kind of problem and I definitely benefitted a lot from this habit.

I would like to thank Prof. Ahmad Fadzil M. Hani of Universiti Teknologi PETRONAS, Tronoh, Malaysia. The lessons I learnt from him under his supervision during my Masters degree helped me a lot while conducting research for my doctoral studies as well.

I specially want to express my gratitude towards Dr. Josiane Zerubia of AYIN, INRIA Sophia-Antipolis, France. It was a great opportunity to introduce myself to her at ICIP, 2012 and I am thankful to her for her mentoring since then, her feedback regarding my research, introducing me to great research opportunities and

for inviting me to give a seminar at AYIN, INRIA, Sophia-Antipolis.

I want to thank Dr. Yaser Yakoob for the important insightful discussions on facial skin analysis we had early in my doctoral research.

I would like to thank my senior group members, Dr. Kaushik Mitra, Dr. Pavan Turaga, Dr. Vishal Patel, Dr. Dikpal Reddy, Dr. Raghuraman Gopalan, Dr. Nitesh Shroff and Dr. Ming-Yu Liu for their stimulating conversations, mentoring and help whenever I needed it.

I would like to thank my fellow group members, Dr. Sima Taheri, Dr. Jaisankar Pillai, Dr. Hien Van Nguyen, Ming Du, Ashish Srivastava, Jie Ni, Garrett Warnell and Chris Reale for making my graduate life memorable.

I would like to thank Chiaki Aita, Yilu Feng and Roushanak Rahmat for their help and encouragement as friends.

The ECE and UMIACS staff members were always very cordial and helpful whenever I had a request. I would like to acknowledge Ms. Melanie Prange and Ms. Janice Perrone for always being there to promptly help me with my requests.

I want to thank Dr. Weina Ge, a PhD candidate in the Computer Science and Engineering Department at the Pennsylvania State University in 2010, and Dr. Xavier Descombes, a part of ARIANA project at INRIA Sophia-Antipolis in 2010, for their help in my early work at implementing Marked Point Process modeling.

It was somewhat of a cultural shock to me to realize how few female members were there in graduate engineering community. I am very grateful to Women in ECE (WECE) community in general and to its previous and current members, Masoumeh Hagpanahi, Filiz Yesilkoy, Aparna Kotha, Bing Shi, Maya Kabkab and

Maice Costa, for helping me settle in the beginning, giving me a sense of community and sisterhood and a forum to talk openly about the challenges inside and outside the school.

I want to acknowledge the Fulbright PhD scholarship for funding my PhD studies as it was next to impossible for me to be able to come to the USA on an F1 visa. I want to acknowledge the US Department of state, USAID and HEC Pakistan for their financial support. I am grateful to my Fulbright advisors at Institute of International Education (USA) namely Brian Diffley, Joel Santana, Stephanie Sasz and Sarah McCormick and the staff at US Education Foundation in Pakistan (USEFP), Islamabad for their help regarding Fulbright scholarship.

Not being able to see any family member over periods of more than 12 months would have been very tough had it not been for the kindness of the American friends and families. I want to thank Ms. Andrea Meyerhoff, Mr. Jacob Thiessen and the Stoecker, Dirks, Orme and Berkey families for their hospitality and for letting me have the comfort of family and home environment at times.

Last but not least, I am grateful to my parents, brothers and sisters-in-law for their support, encouragement and confidence in me. Specially I want to thank my brother, Salman Azam, who went out of his way several times to help me during my PhD.

Finally, life is rarely a smooth sailing journey and there were numerous times when I was in need of help. I want to thank all the anonymous strangers who bestowed gestures of kindness and humanity upon me to get through such tough times. They certainly make this world a safe and worthy place to live in.

Table of Contents

List of Figures	x
1 Introduction	1
1.1 Modeling of Wrinkles as Marked Point Process for Localization	2
1.2 Fast Detection of Facial Wrinkles based on Gabor Features using Image Morphology and Geometric Constraints	3
1.3 Assessment of Discriminative Power of Wrinkles as Soft Biometrics . .	3
1.4 Unsupervised Detection and Removal of Wrinkles from Facial Images using Image Inpainting	4
1.5 Organization of the Dissertation	5
2 Modeling of Facial Wrinkles using Marked Point Process (MPP)	6
2.1 Motivation and Related Work	6
2.2 Marked Point Process (MPP)	8
2.2.1 Prior Model	8
2.2.2 Data Likelihood Energy	9
2.3 Model 1: Representation as Free and Singly Connected Line Segments	10
2.4 Model 2: Representation as Singly and Doubly Connected Line Seg- ments	11
2.4.1 Seed Segments	12
2.5 The Reversible Jump Markov Chain Monte Carlo (RJMCMC) Algo- rithm	14
2.5.1 Calculations of RJMCMC Acceptance Ratios for Model 1	15
2.5.2 Calculations of RJMCMC Acceptance Ratios for Model 2	18
2.6 Experimental Results	20
2.6.1 Localization Results for Model 1	20
2.6.2 Localization Results for Model 2	21
2.6.2.1 Evaluation Setup	21
3 Fast Detection of Facial Wrinkles based on Gabor Features using Image Mor- phology and Geometric Constraints	27
3.1 Motivation	27

3.2	Related Work	27
3.3	Approach	29
3.3.1	Gabor Filters and Image Features	31
3.3.2	The Algorithm	33
3.3.3	Step ‘A’: Detection of Key Wrinkle Sites	34
3.3.4	Step ‘B’: Tracing Wrinkle Curves from Key Wrinkle Sites	36
3.3.4.1	Continuity of Wrinkles	36
3.3.4.2	Penalizing Intersection/Congestion of Wrinkles	38
3.4	Results and Discussion	39
3.4.1	Experiments and Parameter Settings	39
3.4.2	Evaluation Setup and Quantitative vs. Qualitative Results	41
3.4.3	Detection Results in Medium Resolution Images	42
3.4.3.1	Challenges	45
3.4.4	Comparison with Detection Results using MPP Modeling in Low Resolution Images	48
4	Assessment of Discriminative Power of Facial Wrinkles as Curve Patterns	51
4.1	Motivation	51
4.1.1	Related Work	54
4.2	Wrinkle Pattern Matching	55
4.2.1	Resolving Node/Curve Correspondences	56
4.2.2	Wrinkles Pattern Similarity Metrics	60
4.2.2.1	Modified Hausdorff Distance \mathbf{d}_{MHD}	60
4.2.2.2	Curve Proximity Distance \mathbf{d}_{CPD}	61
4.2.2.3	Directed Curve Proximity Distance \mathbf{d}_{DCPD}	61
4.2.2.4	Curve Shape Distance \mathbf{d}_{CSD}	62
4.3	Experiments and Discussion	64
4.3.1	Experiments on Hand-drawn Wrinkles	65
4.3.2	Experiments on Automatically Detected Wrinkles	66
5	Simultaneous Detection and Removal of Wrinkles from Facial Images using Image Inpainting	72
5.1	Motivation	72
5.2	Problem Statement	76
5.3	Related Work	77
5.4	Approach	79
5.4.1	Automatic Detection of Regions with Wrinkles	80
5.4.1.1	Computation of Orientation Fields using Gabor Filters	80
5.4.1.2	Gaussian Mixture Model based on Markov Random Field (GMM-MRF)	82
5.4.1.3	Fusion of Gabor Features and Texture Orientation Field	87
5.4.1.4	Expectation Maximization (EM) Algorithm	90
5.4.2	Automatic Removal of Facial Wrinkles	92
5.4.2.1	Patch based Image Quilting	93

5.4.2.2	Constrained Texture Synthesis to Fill Image Gaps . .	95
5.4.2.3	Selection of Texture Source Template	96
5.4.2.4	Compensation for Skin Tone Variations	96
5.5	Experiments	97
5.5.1	Eperimental Setup	97
5.5.2	Results and Discussion	98
5.5.2.1	Removal of Wrinkles	98
5.5.2.2	Removal of Moles/Dark Spots/Scars	99
5.5.3	Challenges	101
5.5.3.1	Effect of Age/Sagging Skin	101
5.5.3.2	Effect of Illumination	101
5.5.3.3	Artifacts due to the Repetition of Patches	102
6	Conclusion	112
6.1	Summary	112
6.2	Future Work	114
	Bibliography	116

List of Figures

2.1	(a) Wrinkles as broken edges (Canny edge detector). (b) False edges due to illumination discontinuities (Canny edge detector).	7
2.2	(Left) Rejection interaction. (Right) No rejection interaction.	11
2.3	LoG filter response to low resolution forehead image. (a) Original image. (b) $\sigma = \frac{1}{2\sqrt{2}}$. (c) $\sigma = \frac{1}{2}$. (d) $\sigma = \frac{1}{\sqrt{2}}$	11
2.4	(a)Attraction interaction creates doubly connected segment. (b) Rejection interaction.	14
2.5	Step-by-step process for the detection of wrinkles in an image.	19
2.6	(Top) Hand-drawn wrinkles. (Bottom) localization results.	19
2.7	Localization results.	21
2.8	Evaluation setup. (top) Calculation for r_{detect} . (bottom) Calculation for r_{false}	22
2.9	Detection of wrinkles as line segments and fitted curves with hand-drawn wrinkles for few subjects.	24
2.10	Detection of wrinkles as line segments and fitted curves with hand-drawn wrinkles for few subjects.	25
2.11	Detection and false alarm ratios for 36 images of different sizes. (Left) Wrinkles as line segments. (Right) Wrinkles as curves.	26
2.12	Canny edge detector. (a) threshold=0.3 (b) threshold=0.2. (c) threshold = 0.1. (d) Curves fitted to seed segments only. (e) Curves fitted to line segments.	26
3.1	A simple thresholding of Gabor image features. (a) Original forehead image. (b) Gabor image features scaled to grayscale. (c) Resulting binary image with threshold value of 0.5. (d) Resulting binary image with threshold value of 0.3.	30
3.2	Values of $I^N(x_1, x_2)$ (equation 3.4) in grayscale. (a) $p = 1$. (b) $p = 1.25$. Highlighting of image sites with larger Gabor responses can be observed in (b).	31
3.3	Boundary conditions to calculate Gabor features. (a) User-cropped forehead image embedded in a rectangular region with a margin. (b) Grayscale image with mirror boundaries. (c) Gabor image features scaled to grayscale.	31

3.4	Block diagram of wrinkle localization algorithm.	33
3.5	Detection of key wrinkle sites in step 'A' of the algorithm.	34
3.6	Graphical illustration of the geometric constraints of the connected components. (a) 'A' is connected to 'B' through 'D' as 'B' lies in the conic search area of 'A'. (c) 'A' is extended beyond site P_2 by including $\overline{P_2P_3}$. Site P_3 has the maximum Gabor response in the conic search area. (e) Penalizing congestion/intersestion of connected components.	38
3.7	Step 'B': Extension of wrinkle curves from key wrinkle sites.	39
3.8	Three subjects of the same age have visible differences in the appearance of their wrinkles.	39
3.9	Image of a subject having very deep wrinkles around eyes and mouth and light wrinkles on forehead.	40
3.10	A plot of false alarm vs. detection rates in percentage for the 105 images of medium resolution.	42
3.11	A few typical examples of images with a detection rate of $> 70\%$. (Left) Original (Middle) Hand-drawn (Right) Automatically Localized	43
3.12	A few typical examples of images with a detection rate in the range of $50 - 65\%$. (Left) Original (Middle) Hand-drawn (Right) Automatically Localized	44
3.13	A few typical examples of images with a detection rate of $< 50\%$. (Left) Original (Middle) Hand-drawn (Right) Automatically Localized	45
3.14	A few typical examples of images with very light wrinkles. (Left) Gabor features in grayscale. (Middle) Hand-drawn wrinkles. (Right) Automatically localized wrinkles.	46
3.15	A few typical examples of images with skin discolorations masking wrinkles. (Left) Gabor features in grayscale. (Middle) Hand-drawn wrinkles. (Right) Automatically localized wrinkles.	47
3.16	Two examples of images with variant illumination. (Left) Gabor features in grayscale. (Middle) Hand-drawn wrinkles. (Right) Automatically localized wrinkles.	48
3.17	Plots of detection results using current algorithm and MPP modeling for low resolution images used in [1].	49
3.18	Visual comparison of detection results using our algorithm and MPP modeling for low resolution images used in [1].	50
4.1	Different wrinkle patterns with the similar looking curve for three subjects: The curve has been highlighted.	52
4.2	Images of four subjects with variations in wrinkle patterns due to age, expression and/or image acquisition.	53
4.3	Three different patterns of wrinkles for the same subject and right correspondences between curves based on spatial proximity.	56
4.4	Step-by-step illustration of our algorithm for finding node correspondences.	59

4.5	Step-by-step illustration of a generic bipartite graph matching algorithm with global optimization.	59
4.6	The curve correspondences achieved using our algorithm.	60
4.7	Difference between shorter to longer vs. longer to shorter distances in \mathbf{d}_{DCPD}	63
4.8	(Left) Two registered curve patterns (Middle) Calculation of \mathbf{d}_{CSD} (Right) Calculation of \mathbf{d}_{DCPD}	63
4.9	Comparison of recognition rates for hand-drawn wrinkles with different methods.	64
4.10	Comparison of recognition rates for hand-drawn wrinkles with different methods.	65
4.11	Comparison of recognition rates for top two methods.	66
4.12	Four pairs of images where (CPD,CPD) performed better than MHD.	68
4.13	Four pairs of images where MHD performed better than (CPD,CPD).	68
4.14	(Curves fitted to detected wrinkles as line segments (images were taken from [2])).	69
4.15	Comparison of recognition rates for detected wrinkles with different methods.	70
4.16	Comparison of Recognition Rates for detected wrinkles with different methods	71
4.17	Comparison of recognition rates for top 2 methods for detected wrinkles.	71
5.1	Typical results of facial retouching (reproduced from [3]).	73
5.2	Removal of wrinkles by facial retouching (reproduced from [3]).	73
5.3	Reduction of wrinkles by facial retouching (reproduced from [3]).	74
5.4	Reduction of wrinkles by facial retouching (reproduced from [3]).	74
5.5	Typical results of facial retouching for a smart phone application [4]. (a) Original Image. (b) Image after retouching. Note that wrinkles on forehead and brown spots on cheeks are deemphasized due to blending but still visible.	76
5.6	Image features used for segmentation. (a) Forehead image in gray scale. (b) Maximum Gabor amplitude response (values [4.8, 132] scaled to the gray scale values [0,255]). (c) Texture orientation field.	77
5.7	Binary labeling results. (a) Initial labels obtained after using GMM functions in Matlab TM . (b) labeling results after 15 iterations of our GMM-MRF method. (c) labeling results using GMM-MRF method in [5]. (d) Resulting gaps in the skin image.	82
5.8	Results of thresholding. (a) Thresholding maximum Gabor amplitude at value 35. (b) Thresholding orientation field at absolute angle values of less than 5 degrees. (c) Product of images in (a) and (b).	83
5.9	Histogram of the Gabor features in Fig. 5.6(b).	84
5.10	(Right) Rectangle 'A' shows the skin texture used as template whereas rectangle 'B' shows change in skin texture due to a wrinkle. (Left) Orientation field at high resolution, note that sites corresponding to the wrinkle have orientation angle of zero degrees.	84

5.11	Seamless stitching of two patches. (a) Patches with side portions to be overlapped. (b) Minimum square distance boundary cut along the overlapped portions. (c) Resulting stitching of the two patches. . . .	89
5.12	The constrained texture synthesis algorithm divides an irregular shaped gap into a regular grid of patches. Each patch is then marked to be painted if it overlaps any pixels of the gap.	89
5.13	The sides of a patch used for seamless stitching are based on two factors; the patch's location in the grid and if the surrounding patches are marked to be painted or not.	89
5.14	Removing a wrinkle with texture synthesis. (a) Original skin image. (b) Template of the gap to be filled. (c) Skin Image after texture synthesis. Comparing with original skin image in (a), note that wrinkle 'A' has been removed since it was included in the gap whereas part of wrinkle 'B' is re-painted. (d) Stitching of skin patches to fill the gap.	103
5.15	A close up view of wrinkle removal. (a) Original Image. (b) Several wrinkled areas detected by GMM-MRF. (c) Inpainted image after removal of wrinkles. (d) Patches from regular grid fitted on one gap which were actually inpainted.	103
5.16	Flow chart for the algorithm to fill gaps using constrained texture synthesis.	104
5.17	Poisson compensation for color variation under eyes. (a) Inpainted skin without Poisson compensation. (b) Inpainted skin with Poisson compensation.	104
5.18	Level of user interaction. Users have to select polygonal skin patches to remove wrinkles/skin imperfections from.	104
5.19	A typical graph for the objective function vs. iterations of GMM-MRF function. There is little variation in the function for most images after 10 iterations.	105
5.20	Results of wrinkle detection and removal for a subject. (a) Original image. (b) Detected wrinkled areas. (c) Image after wrinkle removal.	105
5.21	Results of wrinkle detection and removal for a subject. (a) Original image. (b) Detected wrinkled areas. (c) Image after wrinkle removal.	106
5.22	Results of detection and removal of skin imperfections including wound scars, acne, brown spots and moles. (a) Original images. (b) Detected imperfections. (c) Images after inpainting.	106
5.23	Results of detection and removal of skin imperfections including wound scars, acne, brown spots and moles. (a) Original images. (b) Detected imperfections. (c) Images after inpainting.	107
5.24	Results of wrinkle and dark spot detection and removal for a subject with darker skin. (a) Original image. (b) Detected wrinkled areas and dark spots. (c) Image after inpainting, most of the dark spots are removed as well.	107

5.25	Results of wrinkle and dark spot detection and removal for a subject with darker skin. (a) Original image. (b) Detected wrinkled areas and dark spots. (c) Image after inpainting, most of the dark spots are removed as well.	108
5.26	Results of wrinkle detection and removal for a subject with sagging skin. (a) Original image. (b) Detected wrinkled areas . (c) Image after inpainting, most wrinkles are removed but skin sagging is visible.	109
5.27	Challenges in the detection of brown spots. (a) Brown spots are over segmented due to lower contrast with surrounding skin. (b) Brown spot is not detected due to the inhomogeneity of texture orientation field within the spot as shown in (b2).	110
5.28	Effect of illumination on detection of wrinkles. (1a,1b) Wrinkles are masked by bright spots. (2a,2b) Wrinkles are masked by shadows. . .	110
5.29	Results of wrinkle detection and removal for a subject. (a) Original image. (b) Detected wrinkled areas. (c) Image after wrinkle removal.	110
5.30	Results of wrinkle detection and removal for a subject. (a) Original image. (b) Detected wrinkled areas. (c) Image after wrinkle removal.	111
5.31	Results of wrinkle detection and removal for a subject. (a) Original image. (b) Detected wrinkled areas. (c) Image after wrinkle removal.	111

Chapter 1: Introduction

In this dissertation we present an analysis of facial wrinkles as key facial aging features for different applications. Wrinkles are important facial features for assessing the age of a subject. An accurate image-based analysis of wrinkles can play an important role in relevant aging applications. Much of the work done in aging applications, e.g. age estimation and simulation, is based on the incorporation of wrinkles as texture features [6–8]. A few attempts have been made to evaluate wrinkles as edges [9, 10]. However, image-based detection and analysis of wrinkles has mostly remained unaddressed. This can probably be attributed to the nature of the artifacts associated with wrinkles as well as the challenges posed by drastically varying skin appearances due to variations in illumination and acquisition angles [11]. The goal of this dissertation is to focus on modeling of wrinkles as lines or curves for several applications. We first present a stochastic approach to detect and localize wrinkles explicitly as curves or line segments from facial images acquired in uncontrolled settings. Then, we present a deterministic approach for faster and more accurate localization of facial wrinkles. Next, we assess the discriminative power of a group of wrinkle curves as a soft biometrics for the recognition of subjects from their wrinkle patterns only and conduct experiments on hand-drawn

and automatically localized wrinkles. Finally, we present our work on detection of skin texture included in and surrounded by facial wrinkles which are then removed using image inpainting techniques.

1.1 Modeling of Wrinkles as Marked Point Process for Localization

In the first part of the dissertation we propose a new generative model for wrinkles on aging human faces using Marked Point Processes (MPP). Wrinkles are considered as stochastic spatial arrangements of sequences of line segments, and detected in an image by proper placement of line segments. The MPP model is proposed under the Bayesian framework where a prior probability model dictates more probable geometric properties and spatial interactions of line segments. A data likelihood term, based on intensity gradients caused by wrinkles and highlighted by LoG filter responses, indicates more probable locations for the line segments. Wrinkles are localized by sampling MPP using the Reversible Jump Markov Chain Monte Carlo (RJCMCMC) algorithm. We also present an evaluation setup to measure the performance of the proposed model. We present results on a variety of images obtained from the Internet to illustrate the performance of the proposed model. Under our evaluation framework, the detection rate is observed to be in the range of 80-90%. As a first example of incorporation of detected wrinkles we evaluate them for their discriminative power as a soft biometrics. We will present a brief overview of that work in the next section and additional details in Chapter 4.

1.2 Fast Detection of Facial Wrinkles based on Gabor Features using Image Morphology and Geometric Constraints

In this part of the dissertation, we present our work on a fast deterministic algorithm based on Gabor filters and image morphology to improve wrinkle localization results obtained by MPP modeling. We propose image features based on Gabor filter banks to highlight subtle curvilinear discontinuities in skin texture caused by wrinkles. Then, image morphology is used to incorporate geometric constraints to localize curvilinear shapes of wrinkles at image sites of large Gabor filter responses. Experiments are conducted on two sets of low and high resolution images and results are compared with those of MPP modeling. The experiments show that not only the proposed algorithm is significantly faster than MPP modeling but also provides visually better results. Using localization results, we also highlight main challenges faced by any algorithm aiming at localizing wrinkles.

1.3 Assessment of Discriminative Power of Wrinkles as Soft Biometrics

In this part of the dissertation, we try to evaluate the discriminative power of wrinkles in human faces and specifically in forehead areas as a soft biometrics. Recently, several micro features of faces e.g. moles, scars have been used in addition to more common facial features for face recognition. In this work, both hand-drawn and automatically detected wrinkles using MPP modeling are tested. We treat a set of

facial wrinkles from an image as a curve pattern and use it for subject recognition. We try to find possible correspondences between curves from different patterns using a simple bipartite graph matching algorithm. Several metrics are introduced to quantify the similarity between two wrinkle patterns. The metrics are based on Hausdorff distance and the curve-to-curve correspondences. We conduct experiments on data sets of both hand drawn and automatically detected wrinkles. The recognition rate for these data sets using only the binary forehead wrinkle curve patterns exceeds 65% at rank 1 and 90% at rank 4.

1.4 Unsupervised Detection and Removal of Wrinkles from Facial Images using Image Inpainting

Facial retouching or beautification is a famous application of the alteration of wrinkle appearance used widely in media and entertainment industry. Recently, with the popularity of smart phones and social networking, user-friendly applications for facial retouching have become popular. Professional software usually require a minimum level of user expertise to achieve the desirable results. On the other hand, user-friendly smart phone applications are easier to use but offer limited efficacy in removing facial skin imperfections. In this part of the dissertation, we present an algorithm to detect facial wrinkles/imperfections which can allow these skin features to be processed differently than the surrounding skin without much user interaction. For detection, Gabor filter responses along with texture orientation field are used as image features. A bimodal Gaussian mixture model (GMM) repre-

sents distributions of Gabor features of normal skin vs. skin imperfections. Then a Markov random field model (MRF) is used to incorporate the spatial relationships among neighboring pixels for their GMM distributions and texture orientations. An Expectation-Maximization (EM) algorithm is used to classify skin vs. skin wrinkles/imperfections. Once detected automatically, wrinkles/imperfections are removed completely instead of being blended or blurred. We propose an exemplar-based constrained texture synthesis algorithm to inpaint irregularly shaped gaps left by the removal of detected wrinkles/imperfections. We present results conducted on images downloaded from the Internet to show the efficacy of our algorithms.

1.5 Organization of the Dissertation

The rest of the dissertation is organized as follows. In chapter 2, we present the theory and results on detection/localization of facial wrinkles using MPP models. Then, in chapter 3, we present results on localization of facial wrinkles using Gabor features and image morphology. In chapter 4, we present results on face recognition using the binary images of wrinkle curve patterns only. Finally, in chapter 5, we present results on unsupervised detection and removal of wrinkles and imperfections from images. Chapter 6 concludes this dissertation and provides future directions of research.

Chapter 2: Modeling of Facial Wrinkles using Marked Point Process (MPP)

2.1 Motivation and Related Work

Wrinkles and fine lines are important facial features for characterizing facial aging. An accurate image-based analysis of these features can play an important role in relevant aging applications. Much of the work done in aging applications, e.g. age estimation and simulation, is based on the incorporation of wrinkles as texture features [6–8]. A few attempts have been made to evaluate wrinkles as edges [9, 10]. However, image-based detection and analysis of wrinkles has mostly remained unaddressed. This can probably be attributed to the nature of the artifacts associated with wrinkles as well as the challenges posed by drastically varying skin appearance in images due to variations in illumination and acquisition angles [11]. In this work, we propose a model to detect and analyze wrinkles explicitly. The basis for our methodology is based on the following observations. First, wrinkles create the appearance of texture on skin, but at high resolution, wrinkles do not depict any repetitive/homogeneous pattern. Second, wrinkles cannot be categorized as boundaries between multiple textures. Third, a wrinkle is usually a discontinuity in

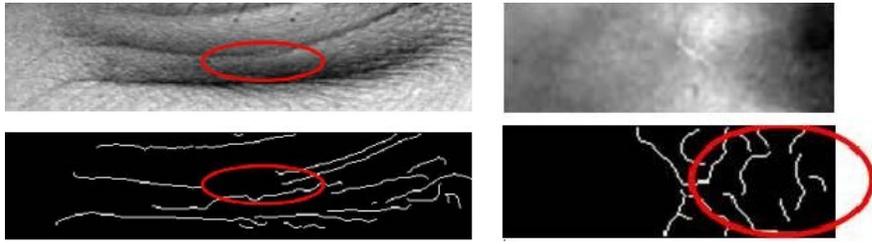


Figure 2.1: (a) Wrinkles as broken edges (Canny edge detector). (b) False edges due to illumination discontinuities (Canny edge detector).

inhomogeneous background skin texture. And finally, wrinkles do not always appear as continuous edges.

It can be argued that an accurate edge detector can detect wrinkles. However, as shown in Figure 2.1(a,b) and Figure 2.12(a-c), the intensity gradients are well picked up but no distinction is made between edges due to wrinkles, illumination discontinuities or specific skin texture. A method capable of filtering intensity gradients based on the probability of there being wrinkles is needed. We propose to incorporate prior semantic information available from typical appearances of wrinkles on human faces for that purpose. Wrinkles are modeled as a stochastic spatial process of ‘line segments’ in a Bayesian framework using Marked Point Process (MPP) [12]. The prior information imposes geometric constraints on the spatial arrangements of line segments. In terms of methodology, our work is related to the work by Soitca et al. [13] for detection of road networks in satellite images.

2.2 Marked Point Process (MPP)

A point process is defined with respect to a Poisson measure. A mark is a set of random parameters associated with a point describing specific properties. In this work, the points represent line segments and a sequence of connected line segments represents a wrinkle. A marked point (line segment) is given by $w_i = (s_i, m_i)$ where $s_i = (x_i, y_i) \in S \subset \mathbb{R}^2$ is the location of center of the segment and $m_i = (l_i, \theta_i) \in M$ is the mark consisting of two parameters denoting the length and orientation of the segment, respectively. The continuous space for parameters is given by $M = [l_{min}, l_{max}] \times [\theta_{min}, \theta_{max}]$. The line segment sequences $\{\mathbf{w} = w_i, i = 1, \dots, n\}$ can be considered as a realization of the MPP on the space $S \times M$. For further details on MPP, the interested reader is referred to [12]. The probability density of MPP can be represented by the Gibbs distribution:

$$f(\mathbf{w}) = c\beta^{n(\mathbf{w})} \exp(-U(\mathbf{w})) \text{ where } U(\mathbf{w}) = U_P(\mathbf{w}) + U_D(\mathbf{w}). \quad (2.1)$$

The term c is the normalizing constant, β is the intensity of the point process, $n(\mathbf{w})$ and $U(\mathbf{w})$ are the number of line segments and the total energy of the realization \mathbf{w} , respectively. The terms $U_P(\mathbf{w})$ and $U_D(\mathbf{w})$ correspond to the energies contributed by the prior model and the data likelihood term, respectively.

2.2.1 Prior Model

The prior model determines more likely geometric properties of the line segments and spatial interactions between neighboring line segments. The parameters $\{l_i, \theta_i\}$

are sampled from a uniform distribution where the parameters of the uniform distribution set limits on the geometric properties of the line segments:

$$l_i \sim \mathcal{U}([l_{min}, l_{max}]) \text{ and } \theta_i \sim \mathcal{U}([\theta_{min}, \theta_{max}]). \quad (2.2)$$

Under the prior model, two segments can interact with each other in spatial domain. In other words, the segments can influence each other's probability of being included in a realization of the MPP due to their spatial proximity. In sections 2.3 and 2.4, two different prior models with different interactions between line segments under two MPP models will be discussed in detail.

2.2.2 Data Likelihood Energy

The data likelihood energy determines where line segments are more probable to appear given data. An image is filtered with a Laplacian of Gaussian (LoG) filter to highlight the intensity gradients caused by wrinkles. Figure 2.3 shows the LoG filter response to a low resolution forehead image. We selected the standard deviation ($\sigma = 0.5$) for LoG filters. Let D_i denote the pixels corresponding to the segment w_i in image. Two different likelihood energy models were tested for this work where the likelihood was proportional to the absolute LoG filter responses and the square of the LoG filter responses respectively. Following are the two terms for the data likelihood energy of the segment w_i under the two models.

$$U_D(w_i) = -\alpha \sum_{d \in D_i} (d). \quad (2.3)$$

$$U_D(w_i) = - \left[\alpha \sum_{d \in D_i} (d - d_{min})^2 \right]. \quad (2.4)$$

where d_{min} is the minimum filter response at a site in the image.

2.3 Model 1: Representation as Free and Singly Connected Line Segments

The first MPP model consists of free line segments (of 3-7 pixels long) and *singly* connected line segments where these segments are connected to the free end points of existing free segments. The two individual geometric properties modeled are length and connectivity. The model favors line segments with smaller length in pixels and line segments with larger lengths are penalized as follows.

$$q_{1,l} = \exp - \left(\frac{l - l_{min}}{l_{min}} \right) \quad (2.5)$$

Regarding connectivity, a segment is allowed to be connected on either side by exactly one line segment which can result in singly or doubly connected segments. We want to penalize individual free line segments with no connections at all. Let $c_i \in \{0, 1, 2\}$ denote the number of connections for the line segment w_i . The segments with more connections are favored as follows.

$$q_{1,c}(w_i) = q(c_i + 1) \quad (2.6)$$

where:

$$q = \left[q(1) \quad q(2) \quad q(3) \right]^T \text{ and } q(3) \geq q(2) \geq q(1) \quad (2.7)$$

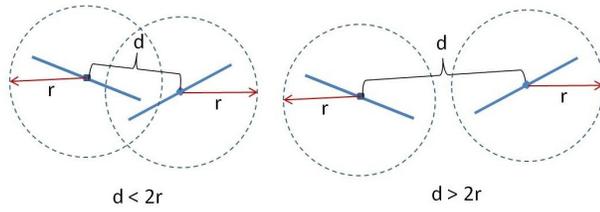


Figure 2.2: (Left) Rejection interaction. (Right) No rejection interaction.

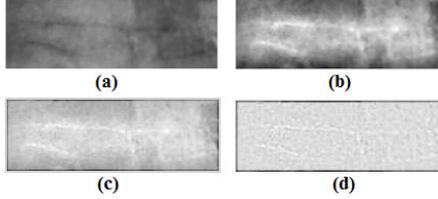


Figure 2.3: LoG filter response to low resolution forehead image. (a) Original image. (b) $\sigma = \frac{1}{2\sqrt{2}}$. (c) $\sigma = \frac{1}{2}$. (d) $\sigma = \frac{1}{\sqrt{2}}$

The interaction between line segments is modeled through the term $q_2(w_i, w_j)$. Two segments have rejection interaction if they are overlapping or lie within a radius r of each other. This penalizes the overlapping or congested line segments:

$$q_2(w_i, w_j) = \gamma^{\mathbb{I}_{(w_i \sim w_j)}} \quad (2.8)$$

The parameter γ is the penalty assigned to segments with rejection interaction and $\mathbb{I}_{(w_i \sim w_j)}$ is the indicator function for interacting segments. Figure 2.2 shows examples of interactions between two line segments.

2.4 Model 2: Representation as Singly and Doubly Connected Line Segments

The second model has the following differences from the first model.

1. The free segments are replaced by *seed* segments. A detailed description of seed segments will be given in the next section. For this reason, there is no need to penalize segments with fewer number of connections and the penalty term $q_{1,c}$ is eliminated.
2. In addition to singly connected line segments, the inclusion of doubly connected line segments is also allowed.
3. The penalization of longer line segments is eliminated so all lengths of line segments in the range $[l_{min}, l_{max}]$ are equally probable.
4. A rectangular interaction region instead of a circular region is proposed. This change allows the inclusion of doubly connected line segments which will be explained in section 2.5.2 later.

2.4.1 Seed Segments

Free line segments in the first model are replaced by seed segments in the second model. The seed segments are essentially free segments with their placement dependent on image intensity gradients in contrast with the random placement of free segments in the first model. This ensures faster convergence of the Markov chain in RJMCMC algorithm as will be explained later. The image sites with the highest filter responses are selected while keeping a minimum distance r_{seed} among them. Seed segments are then placed on these sites in the directions perpendicular to the highest intensity gradients. This results in the placement of seed segments uniformly with inter-seed distance r_{seed} . Let a seed segment be given as

$w(\theta) = \begin{bmatrix} x_{seed} & y_{seed} & l_{seed} & \theta \end{bmatrix}^T$ and the corresponding set of pixels be $D(w(\theta))$.

Then the optimal direction of the seed segment θ^* is found as follows

$$\theta^* = \arg \max_{\theta \in [\theta_{min}, \theta_{max}]} \sum_{d \in D(w(\theta))} d. \quad (2.9)$$

Figure 2.5 includes an example of the seed segments.

We propose a rejection interaction under the prior model which penalizes the birth (addition) of overlapping or congested segments. Let the rejection interaction between two segments $\{w_i, w_j\}$ be denoted by $w_i \overset{r}{\sim} w_j$. A rectangular rejection region of a certain width exists around each existing segment in the configuration. Figure 2.4(b) shows the rectangular rejection interaction region around an existing segment w_1 . The new segment is penalized if it overlaps with the rejection region of any existing segment. The rejection interaction is modeled through the term $q(w_i, w_j)$ as follows. The parameter γ determines the penalty assigned to the interacting new segment.

$$q(w_i, w_j) = \gamma \times \mathbb{I}_{w_i \overset{r}{\sim} w_j} \text{ where } \mathbb{I}_{w_i \overset{r}{\sim} w_j} = \begin{cases} 1 & \text{if } w_i \overset{r}{\sim} w_j \\ 0 & \text{if } w_i \not\overset{r}{\sim} w_j \end{cases} \quad (2.10)$$

The total energy for the prior model is given by:

$$U_P(\mathbf{w}) = \sum_{\substack{(w_i, w_j) \in \mathbf{w} \\ w_i \overset{r}{\sim} w_j}} q(w_i, w_j). \quad (2.11)$$

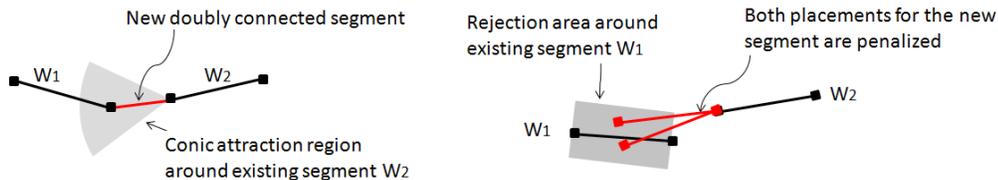


Figure 2.4: (a) Attraction interaction creates doubly connected segment. (b) Rejection interaction.

2.5 The Reversible Jump Markov Chain Monte Carlo (RJCMCMC)

Algorithm

For an MPP, the number of objects is not known a priori and the Metropolis-Hastings algorithm cannot be used to sample from distributions. Green [14] presented the Reversible Jump Markov Chain Monte Carlo (RJCMCMC) algorithm to allow configurations of different dimensions, i.e. different numbers of line segments. The algorithm allows jumps between states in configurations of different dimensions. Given that the state \mathbf{w} is changed to \mathbf{w}' , the algorithm requires the matching of dimensions for two configurations. This is achieved by using an auxiliary random variable ω , sampled from an arbitrary distribution $g(\omega)$, and a bijective transformation \mathcal{T} such that $\mathbf{w}' = \mathcal{T}(\mathbf{w}, \omega)$. The acceptance probability ϕ of a jump modifying configuration \mathbf{w} to the configuration \mathbf{w}' is then modified by the Jacobian of the transformation, $\left| \frac{\partial \mathcal{T}}{\partial (\mathbf{w}, \omega)} \right|$, as follows.

$$\phi(\mathbf{w} \rightarrow \mathbf{w}') = \min\{1, R\} \text{ where } R = \frac{f(\mathbf{w}')}{f(\mathbf{w})g(\omega)} \left| \frac{\partial \mathcal{T}}{\partial (\mathbf{w}, \omega)} \right| \times \frac{\text{prob}(\text{reverse jump})}{\text{prob}(\text{forward jump})}. \quad (2.12)$$

The term R is called the acceptance ratio and needs to be calculated for every

move under RJMCMC. Furthermore, simulated annealing is used to reach the global minimum of the Gibbs energy $U(\mathbf{w})$ in equation (2.1) by replacing the probability density $f(\mathbf{w})$ by $f(\mathbf{w})^{\frac{1}{T}}$ in RJMCMC algorithm. The cooling schedule for T is given by

$$T = \frac{T_0}{\log(1 + i)}. \quad (2.13)$$

where i denotes the current iteration and T_0 is selected to be 10. The simulation of point processes involves birth and death moves, where points are either added to or deleted from the configuration, changing its dimension. The reverse jumps (death) ensure the reversibility of the Markov chain (See Geyer [15] for a proof and further discussion). In the following subsections, the calculations of the acceptance ratios for different moves under the two models are presented in detail.

2.5.1 Calculations of RJMCMC Acceptance Ratios for Model 1

The model 1 represents wrinkles as free line segments and the line segments which are connected to these existing free line segments. We introduce the following moves for this model and calculate their corresponding acceptance ratios.

1. Birth or Death of Free Segments
2. Birth or Death of Connected Segments

Each move involves the calculation of $g(u)$, $\mathcal{T}(\mathbf{w}, u)$, $\left| \frac{\partial \mathcal{T}}{\partial(\mathbf{w}, u)} \right|$ and R . Let $\{n, n_{c0}, n_{c1}, n_{c2}\}$ denote the total number of segments and the number of segments having zero, one and two connections respectively. For the birth of a free segment, let ω

denote the new free segment added to the configuration and let u be the random vector sampled from the segment parameter space (S, M) according to uniform distribution. Then u is given by:

$$u = \begin{bmatrix} x_u & y_u & l_u & \theta_u \end{bmatrix}^T \quad (2.14)$$

and the density function for u denoted by $g(u)$ is given as:

$$g(u) = \frac{1}{\nu(S)} \times \frac{1}{l_{max} - l_{min}} \times \frac{1}{\theta_{max} - \theta_{min}} \quad (2.15)$$

where $\nu(S)$ is the total measure on image space S . The bijective transformation for this move is selected to be:

$$\mathbf{w}' = \mathcal{T}(\mathbf{w}, u) = \{w'_1 = w_1, \dots, w'_n = w_n, \omega = u\} \quad (2.16)$$

Then the Jacobian of the transformation is one. Let $P_{F,birth}$ and $P_{F,death}$ be the probabilities of choosing birth and death of free segments respectively, $P_{F,death}/n_{c0}$ is the probability of reverse jump of death of that particular free segment. Then, according to (2.12), the acceptance ratio for the birth of a free segment can be written as:

$$R = \frac{P_{F,death}}{n_{c0}} \times \frac{\nu(S)(l_{max} - l_{min})(\theta_{max} - \theta_{min})}{P_{F,birth}} \times \frac{f(\mathbf{w}')}{f(\mathbf{w})} \quad (2.17)$$

where the ratio $\frac{f(\mathbf{w}')}{f(\mathbf{w})} = \frac{f(\mathbf{w} \cup \omega)}{f(\mathbf{w})}$ is given as follows:

$$\frac{f(\mathbf{w}')}{f(\mathbf{w})} = \beta \exp \left(q_1(\omega) \sum_{\substack{w_i \in \mathbf{w} \\ w_i \sim \omega}} q_2(w_i, \omega) + \alpha \sum_{d \in D(\omega)} d \right) \quad (2.18)$$

For the birth of a connected segment, a new segment is sampled and connected to a randomly selected segment, with at least one free end, from the configuration. This

move can also result in doubly connected segments. For the death move, a singly connected segment is randomly selected which eventually can result in the deletion of a doubly connected segment. Let (x_e, y_e) be the coordinates of the free end of an existing segment to connect the new segment to. Let $u = (\theta_u, l_u)$ be the random vector sampled from mark space M with density function:

$$g(u) = \frac{1}{l_{max} - l_{min}} \times \frac{1}{\theta_{max} - \theta_{min}} \quad (2.19)$$

Then the new segment ω is given by:

$$\omega(u) = \begin{bmatrix} x_\omega(x_e, \theta_u, l_u) & y_\omega(y_e, \theta_u, l_u) & \theta_u & l_u \end{bmatrix}^T \quad (2.20)$$

$$x_\omega = x_e \pm \frac{l_u}{2} \cos(\theta_u) \text{ and } y_\omega = y_e \pm \frac{l_u}{2} \sin(\theta_u) \quad (2.21)$$

Then it can be shown that the Jacobian of the bijective transformation is given as

$$\left| \frac{\partial \mathcal{T}}{\partial(\mathbf{w}, u)} \right| = \left| \frac{\mathbf{w}'}{\partial(\mathbf{w}, u)} \right| = \left| \frac{\begin{bmatrix} \mathbf{w} & \omega(u) \end{bmatrix}^T}{\partial(\mathbf{w}, u)} \right| = 1 \quad (2.22)$$

Let $P_{C,birth}$ and $P_{C,death}$ be the probabilities of choosing the birth and death of connected segments respectively. Then $P_{C,birth}/(2n_{c0} + n_{c1})$ is the probability of the selection of a particular free end point (x_e, y_e) for the birth of a singly connected segment, and $P_{C,death}/n_{c1}$ is the probability of the reverse jump of death (deletion) of a connected segment. Then the acceptance ratio for the birth of a singly connected segment can be written as:

$$R = \frac{2n_{c0} + n_{c1}}{P_{C,birth}} \times \frac{P_{C,death}}{n_{c1}} \times (l_{max} - l_{min})(\theta_{max} - \theta_{min}) \times \frac{f(\mathbf{w}')}{f(\mathbf{w})}. \quad (2.23)$$

where the ratio $\frac{f(\mathbf{w}')}{f(\mathbf{w})}$ is given in equation 2.18.

2.5.2 Calculations of RJMCMC Acceptance Ratios for Model 2

The model 2 represents wrinkles as single or doubly connected line segments. We introduce the following moves for this model and calculate their corresponding acceptance ratios.

1. Birth or Death of Singly Connected Segments
2. Birth or Death of Doubly Connected Segments

The birth and death of a singly connected segment is the same as in Model 1 with the corresponding acceptance ratio given in equation 2.23. However, in this case the likelihood under Model 2 is different and the ratio $\frac{f(\mathbf{w}')}{f(\mathbf{w})}$ is different from Model 1 as follows.

$$\frac{f(\mathbf{w}')}{f(\mathbf{w})} = \frac{f(\mathbf{w} \cup \omega)}{f(\mathbf{w})} = \beta \exp \left(\alpha \sum_{d \in \omega} (d - d_{min})^2 - \sum_{\substack{w_i \in \mathbf{w} \\ w_i \sim \omega}} q(w_i, \omega) \right). \quad (2.24)$$

The birth or death of a doubly connected segment is actually a result of spatial interactions among segments. The attraction interaction is important when a new line segment is added to the configuration (w) determining if the new segment is going to be connected at one or both ends. Figure 2.4(a) shows an example of birth of a doubly connected segment as a result of attraction interaction. The left end of the existing segment w_2 is selected to be connected to the new segment. An attraction interaction region is defined to be a conic region connected to that end point. An existing segment in that region (w_1 in this case) results in the new segment

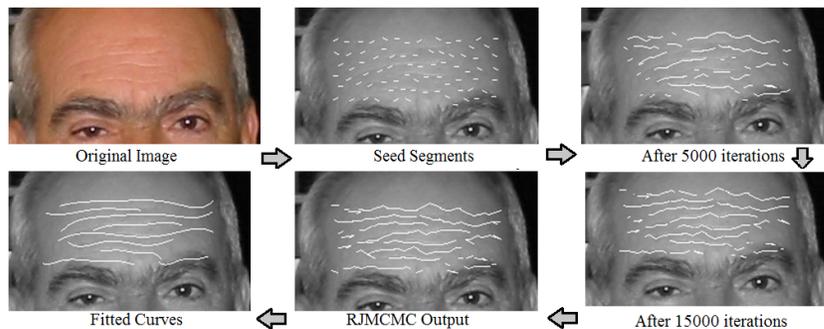


Figure 2.5: Step-by-step process for the detection of wrinkles in an image.

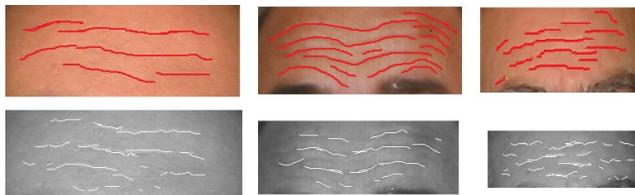


Figure 2.6: (Top) Hand-drawn wrinkles. (Bottom) localization results.

being connected at both sides between the two existing segments. A segment is not allowed to be connected to more than one segment at either end point.

In the case of the birth of a doubly connected segment, the parameter vector u for the new line segment ω is simply a function of the neighboring two segments and is not sampled randomly. In this case $g(u) = 1$ and $u = \omega$ which reduces the Jacobian of the bijective transformation to one. Then the acceptance ratio for the birth of a doubly connected segment is given as:

$$R = \frac{2n_{c0} + n_{c1}}{P_{C,birth}} \times \frac{P_{C,death}}{n_{c2}} \times \frac{f(\mathbf{w}')}{f(\mathbf{w})}$$

2.6 Experimental Results

For assessing the performance of the detection algorithm in uncontrolled image settings, images were selected from the Internet and the FG-NET database [16]. Aging results in wrinkles at several regions of a face i.e. forehead, eye, mouth and nose corners as well as at sagging contours of the skin. For this work, we restricted to the detection of wrinkles on forehead for two reasons (a) generating the ground truth by hand-drawing of wrinkles on the foreheads was easier and (b) wrinkles were more obvious on the forehead in most of the images. For each image, the forehead was hand cropped to a rectangle and input to the detection algorithm. It should be mentioned here that the detection of forehead area was not considered under the scope of this work and user-provided cropping was considered sufficient. In total we used 36 images with hand-drawn wrinkles as the baseline. Where most of the wrinkles were obvious, some were less apparent and posed difficulty even for humans. The image size varied from 86 x 37 as minimum to 290 x 110 as maximum. Resizing of images was not considered as we wanted to assess the performance of the detection algorithm on images of different size and resolution.

2.6.1 Localization Results for Model 1

Four different types of moves are allowed under model 1 where each move was selected with equal probability i.e. $1/4$. Under RJMCMC, one iteration performed one move only. Table 2.1 shows selected values for different parameters of the MPP model. Figure 2.7 shows localization results for different subjects and Figure 2.6

α	β	γ	$\log q_{c0}$	$\log q_{c1}$	$\log q_{c2}$
4000	0.005	5000	-10^{10}	10^6	10^{10}

Table 2.1: Parameter Values



Figure 2.7: Localization results.

shows the comparison of ground truth vs. localization results for some images. It can be observed that the localized wrinkles closely resemble the ground truth. However, the line sequences are broken at some places and do not cover the original wrinkle completely. At some locations, some overlapping of segments can also be observed.

2.6.2 Localization Results for Model 2

2.6.2.1 Evaluation Setup

Since localization of wrinkles is a new application in the area of image-based analysis of aging skin, an evaluation setup is required to assess the performance of detection algorithms. The terminology of ‘detected’, ‘original’ and ‘well-localized’ wrinkles is

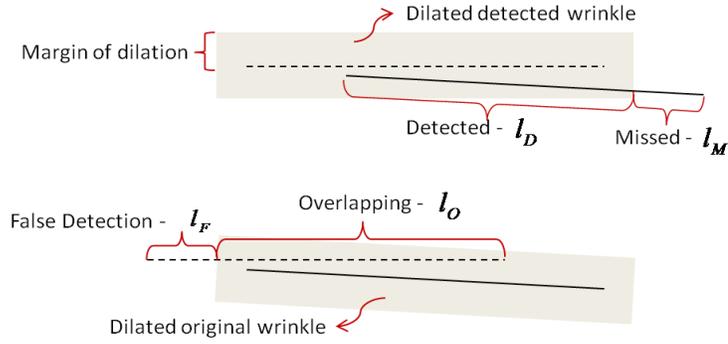


Figure 2.8: Evaluation setup. (top) Calculation for r_{detect} . (bottom) Calculation for r_{false} .

used. The detected wrinkles are the output of the algorithm. The original wrinkles are those hand-drawn by a user and the well-localized wrinkles are wrinkles detected at correct locations. A margin of $m = 3$ pixels is allowed in localization i.e. a detected wrinkle is considered well-localized if it is within the distance of m pixels from the hand drawn wrinkle. This way the margin accounts for small inaccuracies in the hand-drawing of wrinkles or small displacements in maximum gradient locations. Morphological dilation with the margin m is used to define the overlap area. A detected wrinkle is considered well-localized if it lies in the overlap area.

Let n_W and n_D be the total number of hand drawn and detected wrinkles respectively. We propose the following ratios for evaluation. Figure 2.8(a) shows the detailed procedure for determining the ratios.

1. Detection Ratio (\mathbf{r}_{detect}): The ratio of the total length of original wrinkles within the overlap region of detected wrinkles to the total length of the original wrinkles.

$$r_{detect} = \frac{\sum_{n_W} l_{overlap}}{\sum_{n_W} (l_{overlap} + l_{miss})} \quad (2.25)$$

2. False Alarm Ratio (\mathbf{r}_{false}): The ratio of the total length of falsely detected wrinkles to the background area with no wrinkles ($\nu(S)$ represents the measure on image space).

$$r_{false} = \frac{\sum_{n_D} l_{false}}{\nu(S) - \sum_{n_W} l_{original}} \quad (2.26)$$

3. Miss Ratio (\mathbf{r}_{miss}): The ratio of the total length of missed original wrinkles to the total length of original wrinkles.

$$r_{miss} = \frac{\sum_{n_W} l_{miss}}{\sum_{n_W} (l_{overlap} + l_{miss})} \text{ where } r_{miss} = 1 - r_{detect} \quad (2.27)$$

The cropped forehead images were input to the RJMCMC algorithm and 2×10^4 iterations were used for every image. The probabilities $P_{C,birth}$ and $P_{C,death}$ were set equal to 0.5 each. Seed parameters $\{r_{seed}, l_{seed}\}$ were fixed at 9 and 4 pixels respectively. The two parameters γ and α were selected heuristically. A large value of γ , 150 in this case, suffices to penalize overlapping or congested segments. The parameter α favors the birth of new segments. Figure 2.11(b) shows a plot of results with α varying over a scale of 10^6 for an image (Note that the x-axis has logarithmic scale). The number of segments increases with α increasing both detection and false alarm ratios. The optimal values for α were in the range of 0.001-0.03 for smaller images and in the range of 0.00008-0.0005 for larger images. An unsupervised estimation technique for the parameters, however, is required. The



Figure 2.9: Detection of wrinkles as line segments and fitted curves with hand-drawn wrinkles for few subjects.

dynamics of the Markov Chain were similar for every image where the total number of segments increased sharply in the beginning and then became stable with equal proportion of birth and death moves. Figure 2.5 shows the different steps in the algorithm. The increased number of line segments from the initial state of seed segments can be seen after 5000 iterations.

Figure 2.10 shows the results for some of the 36 test images both as line segments and curves. Figure 2.11(a) shows the ROC points for the images. Each plot point is represented by a square whose size is proportional to the size of the image. Most of the images have detection rates greater than 80% and false alarm rates around 2.5% on an average. It can be observed from the plot that the results are independent of the image size. The images on the right end of the plot have

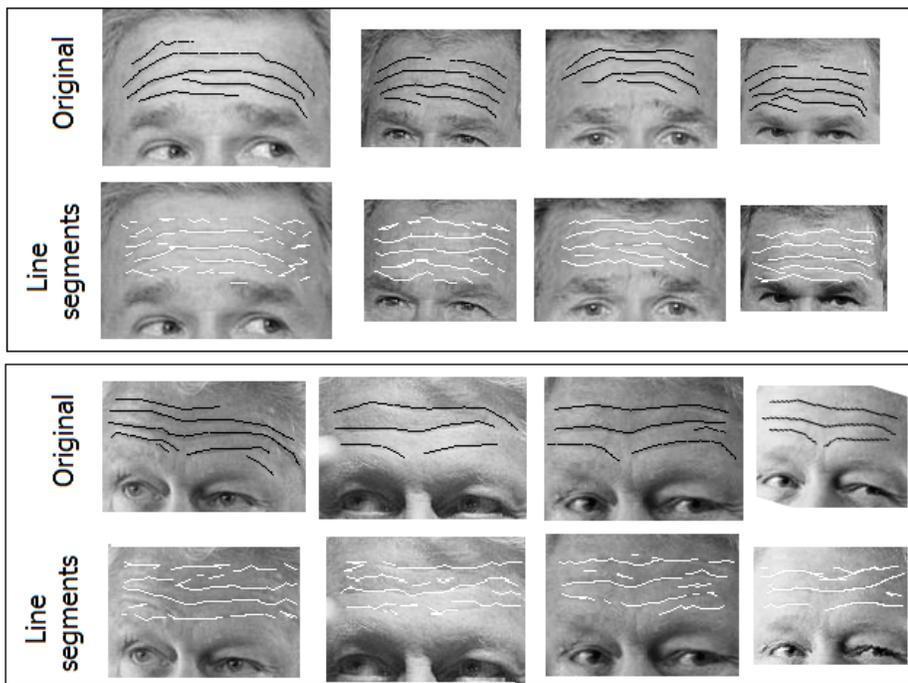


Figure 2.10: Detection of wrinkles as line segments and fitted curves with hand-drawn wrinkles for few subjects.

relatively lower detection and higher false alarm rates. These are the cases where the algorithm performed relatively poorly because of three reasons (a) some of the wrinkles were not obvious and had lower intensity gradients and, as a result, data likelihood energy values (b) wrinkles were mixed with other skin color irregularities and (c) the subject had relatively discontinuous wrinkle curves. Overall, the detection rates were observed to be better in subjects with deeper and continuous wrinkle curves.

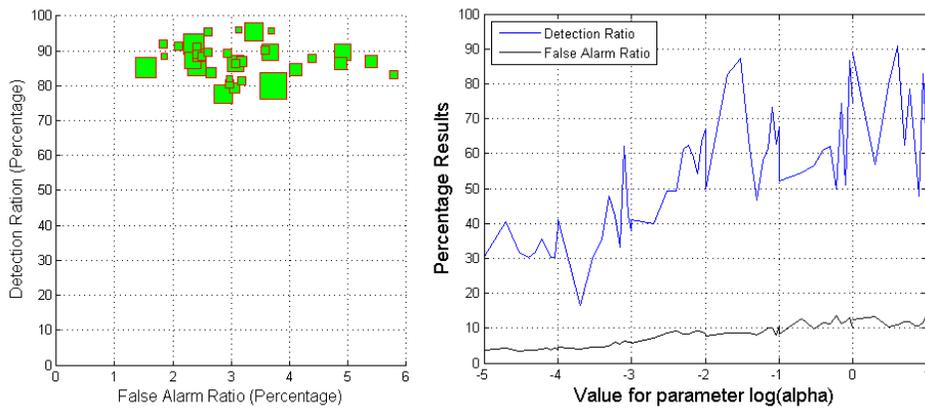


Figure 2.11: Detection and false alarm ratios for 36 images of different sizes. (Left) Wrinkles as line segments. (Right) Wrinkles as curves.

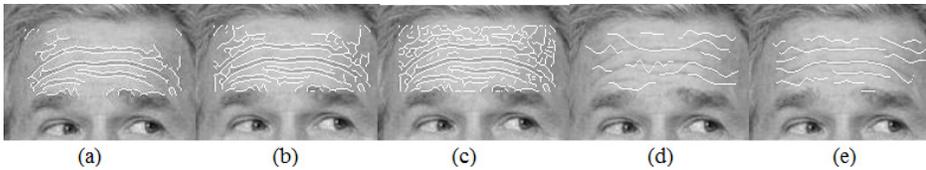


Figure 2.12: Canny edge detector. (a) threshold=0.3 (b) threshold=0.2. (c) threshold = 0.1. (d) Curves fitted to seed segments only. (e) Curves fitted to line segments.

Chapter 3: Fast Detection of Facial Wrinkles based on Gabor Features using Image Morphology and Geometric Constraints

3.1 Motivation

Previously, we proposed a model based on Marked Point Processes to localize wrinkles as line segments/curves [1,2] and observed some limitations. First, the value of the parameter α varied with image resolution. Second, the simulated annealing procedure used for optimization warranted a slower cooling schedule and a large number of iterations of RJMCMC which required considerable computation time. These factors led us to adopt a deterministic approach based on Gabor features and image morphology as compared to the stochastic approach of MPP modeling.

3.2 Related Work

Curvilinear structures of interest appear in several types of medical and non-medical images. These structures can either constitute the main objects of an image or can appear as some sort of anomaly/discontinuity in the main objects of an image. Few examples of the former are finger prints in finger print images, retinal blood vessels in fundus camera images and brain sulcal lines in MRI scan images. Few

examples of the latter are structures like buildings, roads and rivers in aerial images and cracks in manufactured steel slabs and railway tracks. An extensive survey of all the related work regarding the detection of curvilinear structures is out of scope of this work. Here, we review only the methods based on filtering and mathematical morphology.

Normally, the targeted curvilinear structures have image features very similar to those of background and pose challenges to the extraction of accurate shapes of the structures. A thorough survey of detection of cracks in roads can be found in [17] and detection of blood vessels in retinal images can be found in [18]. In cases where enough data is available, one set of approaches is to apply machine learning techniques to learn an appropriate data model. The other set of approaches is to use filtering (e.g. Gabor filters, Wavelets, Contourlets, Gaussian filters) and/or morphological processing of data. Several oriented feature detectors have been developed including steerable Gaussian second-derivative filters, line operators and Gabor filters. A comparative study can be found in [19] where real Gabor filters were assessed to be the best detector of oriented features. Figure 3.1 depicts a forehead image and the corresponding maximum amplitude filtered response from a Gabor filter banks as a grayscale image. It can be observed that the wrinkle discontinuities are well highlighted by Gabor filter bank. However, as seen in Figures 3.1(c) and (d), a simple thresholding of the Gabor filter response is unable to yield well localized wrinkle curves. Hence, we apply morphological techniques to retrieve wrinkle curves from thresholded Gabor response images.

In cases where prior knowledge about the geometric properties of the objects of

interest is available, mathematical morphology has been a powerful tool to enhance and/or identify such objects. Consequently, morphology has been used as segmentation and post-processing technique in detection of curvilinear structures in wide range of applications. An early piece of work on detection of (blood-vessel like) curvilinear structures in medical images using morphology was presented by Zana and Klein in [20]. The so-called vessel-like patterns were defined to have certain geometric properties of shape, width, connectivity and separation. These properties were manipulated by the morphological operations of top-hat, opening and closing to highlight such patterns to be analyzed further for their curvature. Landstrom and Thurley proposed an algorithm based on several morphological operations to detect cracks in steel slabs from 3D range data [21] while focusing on minimizing false positives. Lam and Yan used divergence of vector fields to detect vessel like objects in retinal images and then applied skeletization to remove artifacts [22]. Chambon and Moliard used locally variant thresholding, closing, median/mean filtering and histogram equalization to detect cracks in road pavements [17]. Lelore and Bouchara used dilation and analysis of pixel-level neighborhoods to reduce false positives in document image restoration [23].

3.3 Approach

The MPP modeling provided a way to incorporate prior knowledge about wrinkles in spatial interaction of line segments. For example, overlapping and congestion of line segments were penalized. Since such spatial interactive rules were a key to

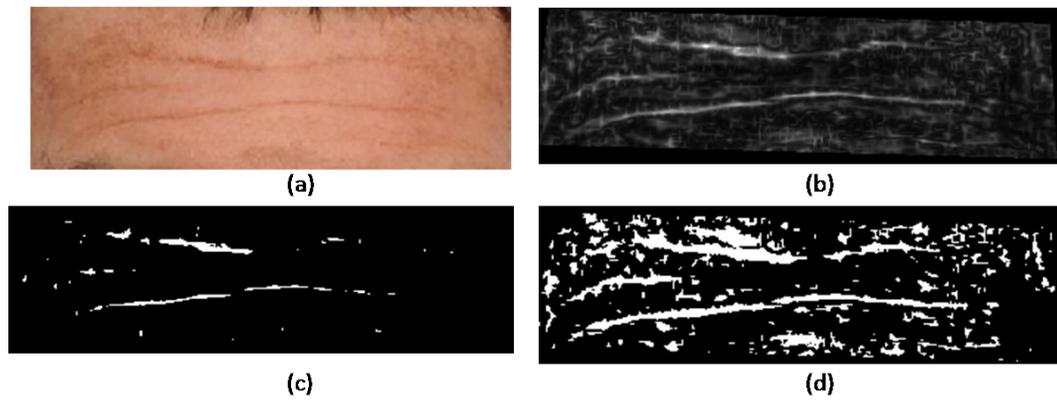


Figure 3.1: A simple thresholding of Gabor image features. (a) Original forehead image. (b) Gabor image features scaled to grayscale. (c) Resulting binary image with threshold value of 0.5. (d) Resulting binary image with threshold value of 0.3.

distinguishing edges due to wrinkles from those due to other factors, a deterministic approach incorporating similar rules is expected to perform better for the detection of wrinkles. In this work, we apply successive thresholding and morphological operations to incorporate geometric characteristics of facial wrinkles. The resulting geometric constraints can then discriminate between image intensity gradients due to wrinkles and other factors. Our approach is based on the following assumptions regarding geometrical and image characteristics of wrinkles.

A1: Wrinkles create intensity gradients which are highlighted by Gabor filter banks.

A2: Wrinkles appear as curvilinear features instead of blob like features.

A3: Wrinkles are continuous and not a series of disconnected curvilinear segments.

A4: Wrinkles are less probable to intersect each other or to be congested in small areas.

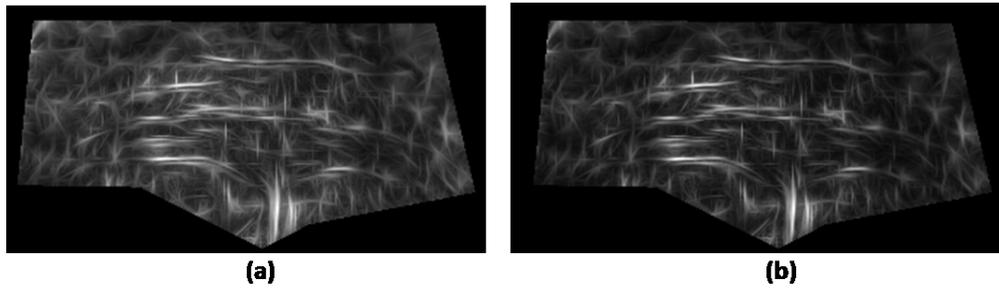


Figure 3.2: Values of $I^N(x_1, x_2)$ (equation 3.4) in grayscale. (a) $p = 1$. (b) $p = 1.25$. Highlighting of image sites with larger Gabor responses can be observed in (b).

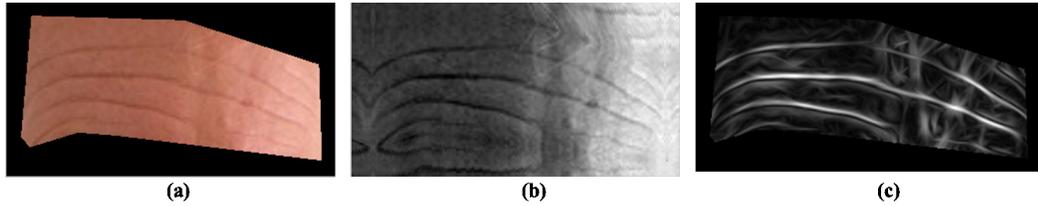


Figure 3.3: Boundary conditions to calculate Gabor features. (a) User-cropped forehead image embedded in a rectangular region with a margin. (b) Grayscale image with mirror boundaries. (c) Gabor image features scaled to grayscale.

3.3.1 Gabor Filters and Image Features

Several oriented feature detectors have been developed including steerable Gaussian second-derivative filters, line operators and Gabor filters. A comparative study can be found in [19] where the real Gabor filters were shown to be the best detector of oriented features. We use Gabor filter responses as image features to highlight the curvilinear features of wrinkles in this work. The real Gabor filter kernel oriented at the angle α is given by

$$g(x_1, x_2) = \frac{1}{2\pi\sigma_{x_1}\sigma_{x_2}} \exp \left[\frac{-1}{2} \left(\frac{x_1'^2}{\sigma_{x_1}^2} + \frac{\gamma^2 x_2'^2}{\sigma_{x_2}^2} \right) \right] \cos(2\pi f x_1') \quad (3.1)$$

where the parameters $\sigma_{x_1}, \sigma_{x_2}$ denote the scale of the 2D Gaussian envelope, f denotes the frequency of the sinusoid and γ denotes the spatial aspect ratio which defines the ellipticity or the elongation of support of Gabor function. The values for x_1, x_2 are given as follows:

$$\begin{bmatrix} x_1' \\ x_2' \end{bmatrix} = \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (3.2)$$

Let $\{g_k(x_1, x_2), k = 0, \dots, K - 1\}$ denote the set of real Gabor filters oriented at angles $\alpha_k = -\frac{\pi}{2} + \frac{\pi k}{K}$ where K is the total number of equally spaced filters over the angular range $[-\frac{\pi}{2}, \frac{\pi}{2}]$. Let $\{I(x_1, x_2); x_1 = 1 \dots N_1, x_2 = 1 \dots N_2\}$ denote the input image in gray scale and $I_k^F(x_1, x_2)$ denote the image filtered by the filter $g_k(x_1, x_2)$.

The corresponding maximum amplitude among the filtered responses is given as:

$$I'(x_1, x_2) = \max_k I_k^P(x_1, x_2) \quad (3.3)$$

The maximum amplitude response is normalized to the range $[0, 1]$ according to the following equations:

$$I''(x_1, x_2) = \left(I'(x_1, x_2) - \min_{(x_1, x_2)} I'(x_1, x_2) \right)^p \quad (3.4)$$

$$I^N(x_1, x_2) = \frac{I''(x_1, x_2)}{\max_{(x_1, x_2)} I''(x_1, x_2)} \quad (3.5)$$

where the value of the parameter p is set to be greater than 1 to highlight the image sites with larger Gabor filter responses. Figure 3.2 includes two images of

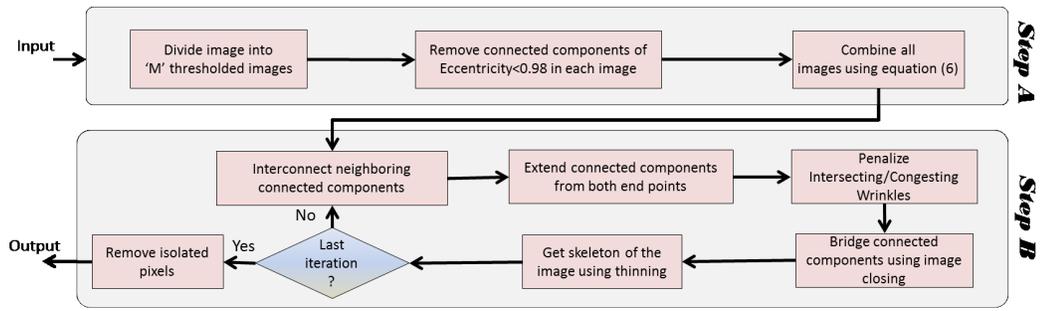


Figure 3.4: Block diagram of wrinkle localization algorithm.

$I^N(x_1, x_2)$ with $p = 1$ and $p = 1.25$ as an example. The set of the normalized maximum filter response, $\{I^N(x_1, x_2)\}$, constitutes image features for automatic detection of wrinkles. Figure 3.3(a) shows a user-cropped forehead image embedded in a rectangular image. Figure 3.3(b) depicts the mirror boundary conditions used to fill the black border around the forehead image in Figure 3.3(a) to eliminate the boundary artifacts in Gabor response images. Figure 3.3(c) shows the corresponding maximum Gabor responses in the original forehead image. It can be observed that high filter responses are caused by both illumination variation, appearing mostly vertically in the middle of the forehead, and wrinkles.

3.3.2 The Algorithm

Figure 3.4 shows a block diagram for our two-step algorithm. The first step is based on assumptions ‘A1’ and ‘A2’ and aims at detecting key image sites with high gradients more probable to be caused by wrinkle than by other factors. This is done by gradual thresholding and discarding of the thresholded blob-like regions. The second step is based on assumptions ‘A3’ and ‘A4’. The purpose of this step is

to trace wrinkle curves out from key image sites detected in the first step. Wrinkle curves are extended from key image sites while penalizing congestion or intersection of wrinkle curves. Following, we present details of the incorporation of geometric constraints using our two-step algorithm.

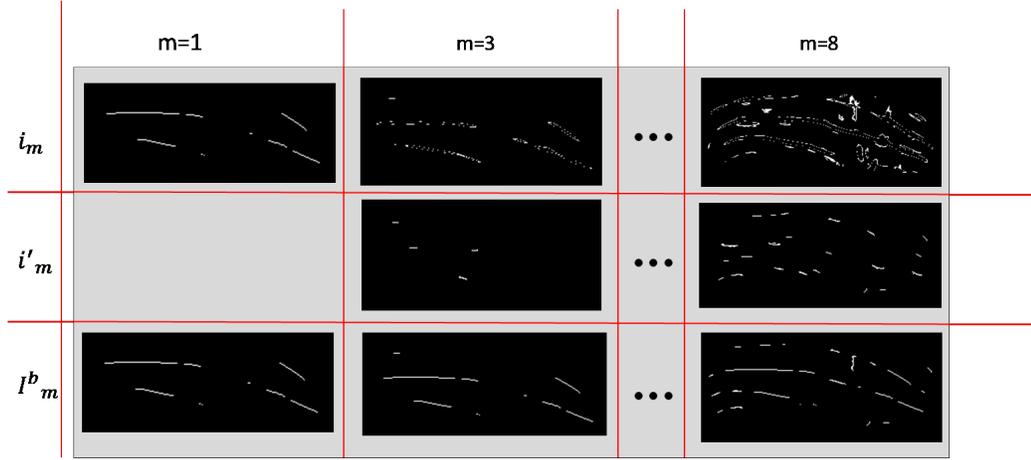


Figure 3.5: Detection of key wrinkle sites in step ‘A’ of the algorithm.

3.3.3 Step ‘A’: Detection of Key Wrinkle Sites

This step is based on first two assumptions that wrinkles are curvilinear structures and cause high intensity gradients in the image. In this step, key image sites with high Gabor filter response, non-overlapping and in curvilinear shapes (instead of blob-like shapes) are detected. Such sites have higher probability of being part of a wrinkle. The process is similar to the placement of seed segments under MPP modeling where a favorable initial state in the RJMCMC algorithm was provided by placing ‘seed’ segments on image sites with the highest filter responses. For the detection of key wrinkle sites, successive thresholding of the image $\{I^N(x_1, x_2)\}$ and

morphological processing of the resulting binary images are applied.

Let $T = \{t_m | t_m \in (t_{\min}, 1); t_m = 1 - m\Delta_t, m = 1, 2, \dots, M\}$ denote a set of thresholds. The image $\{I^N(x_1, x_2)\}$ is quantized in threshold increments of Δ_t resulting in M binary images where the number M is determined by the selection of $t_{\min} \in (0, 1)$. Let each binary image be denoted as i_m where $t_{m-1} \leq i_m < t_m$. Each binary image is processed to discard the connected components having blob-like and non-curvilinear shapes using the ‘eccentricity’ property of connected components. Let i'_m denote the resulting binary images having only curvilinear portions. The binary images are then combined while keeping a certain distance among selected connected components. The reason for maintaining a distance is to prevent consecutively thresholded connected components from aggregating into non-linear shapes which would normally result from a simple thresholding of Gabor response image. Let us assume that we want to select connected components r^d pixels apart. Then the images are dilated by a disk, denoted by D_{r^d} , of radius r^d and combined using ‘OR’ operation recursively to obtain final key wrinkle sits as follows:

$$\begin{aligned}
 I_1^b &= i'_1 \\
 I_2^b &= (I_1^b \oplus_b D_{r^d})^c \vee i'_2 \\
 &\vdots \\
 I_M^b &= (I_{M-1}^b \oplus_b D_{r^d})^c \vee i'_M
 \end{aligned} \tag{3.6}$$

where symbols ‘ \oplus_b ’ and ‘ \vee ’ denote the operations of binary dilation and ‘OR’ respec-

tively. Figure 3.5 illustrates this process where the top row shows simple thresholded binary images with several connected components at three different iterations. The middle row shows images after discarding connected components of high eccentricity values. The bottom row shows the combined binary image upto that iteration. The final image, I_M^b , consists of key wrinkle sites and is processed in the second step of the algorithm.

3.3.4 Step ‘B’: Tracing Wrinkle Curves from Key Wrinkle Sites

After the key sites have been detected, the next step is to trace the wrinkle curves from those sites. This step is similar to the birth and death moves of line segments under MPP model where new line segments are connected to the existing ones. The processing in this step is based on the last three assumptions regarding the geometric characteristics of wrinkles as explained follows.

3.3.4.1 Continuity of Wrinkles

The third geometrical property of wrinkles, continuity, is incorporated in two steps. First, a search area of a certain shape, based on the orientation of connect component, is searched around each end point of the connected component. If there exists an end point of a neighboring connected component, the two components are joined. Figure 3.6(a) demonstrates this step. The component ‘A’ has an end point of a nearby component ‘B’ in the search area of its left end point. Hence, the two components are joined by placing the component ‘D’ between their end points.

Regarding search area of the right end point of the component ‘A’, since there is no end point of any other component, this end point will be extended in the next step. The shape of the search area depends on the orientation of the connected component to ensure smoothly varying wrinkle curves. The search area is a circular region in case of a single point component (Figure 3.6(b)), vertically oriented conic regions in case of a component as vertical line (Figure 3.6(c)) and horizontally oriented conic regions otherwise (Figure 3.6(a)).

If no neighboring components are present, for continuity, the end points of a connected component are extended to nearby Gabor response maxima. As an example, in Figure 3.6(b), the component ‘A’ is extended from the right end point P_2 . First a conic region of the radius r_E and angle θ_E is searched for the maximum $I^N(x_1, x_2)$. If this maximum is greater than t_{min} , the corresponding point is considered for extension. Let this point be denoted as P_3 in Figure 3.6(d). The component ‘A’ is then extended by adding line segment $\overline{P_2P_3}$. This process can be considered similar to the active contours (or ‘snake’) algorithms where an initial curvilinear segment is drifted to the local maxima along the directions of local gradients. However, in case of wrinkles, a simpler extension suffices because of two reasons, (a) the newer extended segments are only a few pixels long and a greedy algorithm is efficient and (b) the initial location of the line segment is close to the maximum as it was extended from an existing connected component.

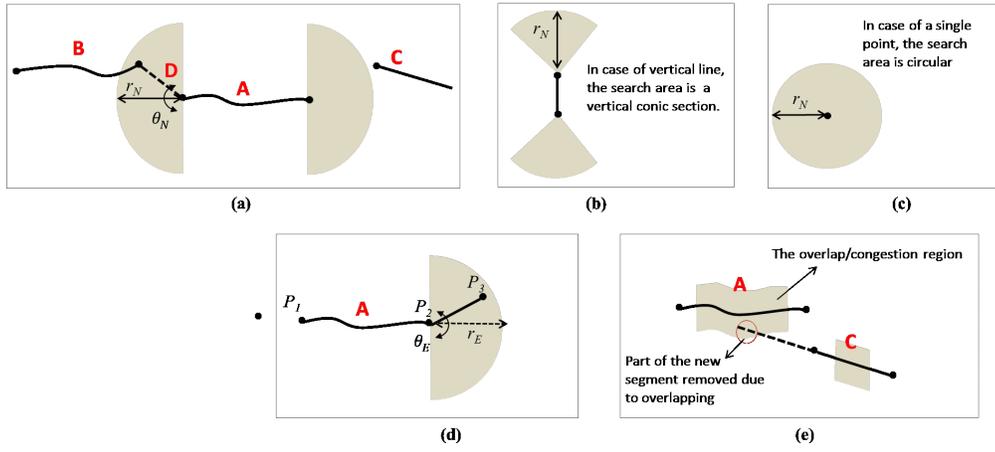


Figure 3.6: Graphical illustration of the geometric constraints of the connected components. (a) 'A' is connected to 'B' through 'D' as 'B' lies in the conic search area of 'A'. (c) 'A' is extended beyond site P_2 by including $\overline{P_2P_3}$. Site P_3 has the maximum Gabor response in the conic search area. (e) Penalizing congestion/interseption of connected components.

3.3.4.2 Penalizing Intersection/Congestion of Wrinkles

In the process of tracing wrinkle curves from key wrinkle sites, assumption 'A4' is incorporated to penalize intersection and congestion of wrinkle curves. This is achieved using binary dilation. A non-overlapping area for every connected component is obtained by dilating it with a rectangular region. During extension of wrinkle curves, any portions overlapping with these areas are discarded. As an example, Figure 3.6(c) shows two wrinkle curves with their non-overlapping neighborhood. When wrinkle curve 'C' is extended, its portion overlapping with the neighborhood of wrinkle curve 'A' is discarded.

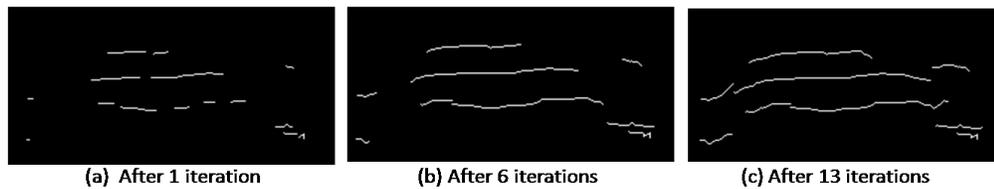


Figure 3.7: Step ‘B’: Extension of wrinkle curves from key wrinkle sites.



Figure 3.8: Three subjects of the same age have visible differences in the appearance of their wrinkles.

3.4 Results and Discussion

3.4.1 Experiments and Parameter Settings

We conducted experiments on two sets of images. The first set consisted of high resolution images of public figures downloaded from the Internet. The second set consisted of the low resolution images used in our previous work [1] for comparison. Facial wrinkles create a wide variety of image gradients and can vary from being very light to being very deep across individuals even of the same age. For example, Figure 3.8 contains images of three public figures taken at the age of 52 illustrating the differences in wrinkle length and depth. At the same time, depth of wrinkles can vary significantly within one individual’s face as well. Figure 3.9 depicts an image

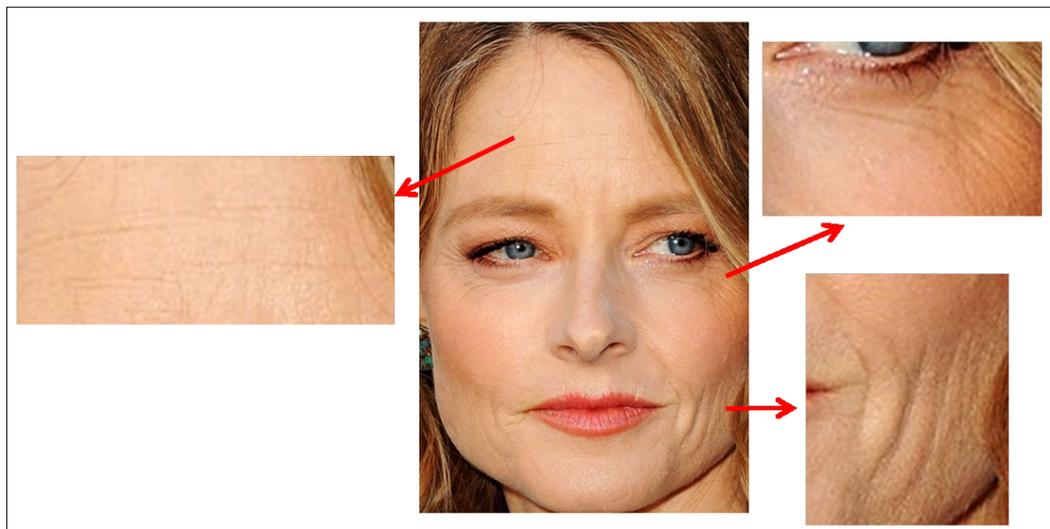


Figure 3.9: Image of a subject having very deep wrinkles around eyes and mouth and light wrinkles on forehead.

of a subject where wrinkles around eyes and corners of mouth are much deeper than those on forehead. For this reason we used a larger Gabor filter bank consisting of sets of frequencies and scales which can highlight wrinkles of varying length and depth across individuals.

We used images of public figures, mostly celebrities, downloaded from the Internet as our data set for the reason that medium to high resolution images of celebrities were more easily available. We also made the interesting observation that it was relatively more difficult to find images of female celebrities with facial wrinkles because of the masking of wrinkles due to skin treatments, photo retouching or occluding hair styles. We did not conduct experiments on images from currently available aging databases e.g. FG-Net, FERET, MORPH, due to the low resolution and/or quality of images. Table 3.1 shows parameter values for the two sets of im-

ages.

Parameter	High Resolution Images	Low Resolution Images
Gabor filter frequency, f	{0.3, 0.4}	0.05
Gabor filter scale $\sigma_{x_1} = \sigma_{x_2}$	1	{10, 15}
Gabor filter aspect ratio γ	15	10
t_{\min}	0.1	0.1
r^d (pixels)	10	7
r_N (pixels)	6	4
r_E (pixels)	$r_E - 2$	$r_E - 2$

Table 3.1: Parameter values for experiments on two sets of images.

3.4.2 Evaluation Setup and Quantitative vs. Qualitative Results

We presented an evaluation set up to assess the detection results quantitatively in the last chapter. The evaluation framework calculated quantitative results in the form of detection, false alarm and miss ratios. However, the following observations were made regarding the limitations of this evaluation setup.

- (A): High detection rate did not ensure results of visually high quality where the term ‘quality’ can be described as the continuity of a wrinkle curve as well as its smooth curvature looking more similar to real wrinkles.

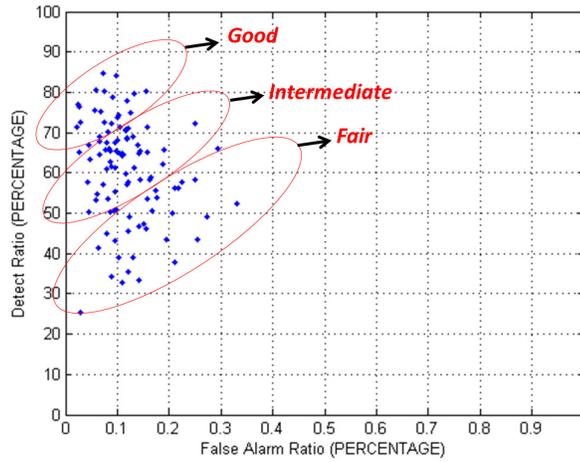


Figure 3.10: A plot of false alarm vs. detection rates in percentage for the 105 images of medium resolution.

(B): The hand-drawn wrinkles can be subjective depending on the user perception of wrinkles. We observed that one user tended to draw more wrinkles than others. This introduced subjectivity in the quantitative results calculated by our evaluation setup.

Therefore, although we follow the same evaluation set up for comparison with the previous results of MPP modeling, we make additional remarks about the quality of results where appropriate. In general, we observed that a false alarm rate lower than 0.2% contributed to visually better detection results.

3.4.3 Detection Results in Medium Resolution Images

In this experiment we downloaded images of high resolution, more than ‘1024 x 768’ pixels, of public figures. There were 123 images of 60 subjects of both genders and different ethnicities and ages. There were one or more images of a subject

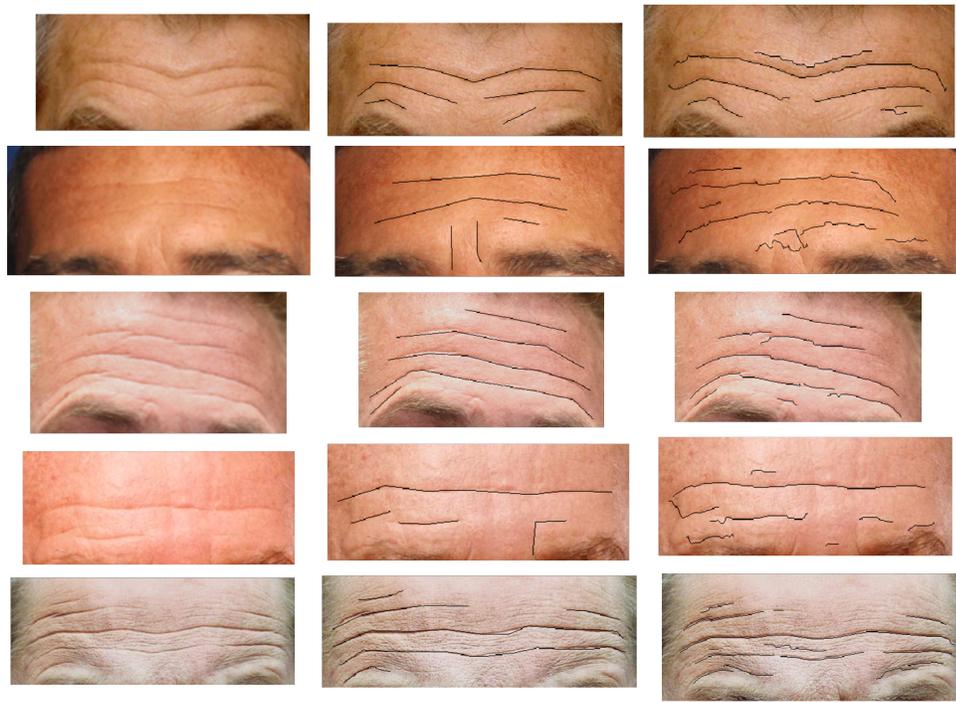


Figure 3.11: A few typical examples of images with a detection rate of $> 70\%$.

(Left) Original (Middle) Hand-drawn (Right) Automatically Localized

depending on quality of the images available on the Internet for that particular subject. In each image, the face was hand cropped and resized so that the maximum dimension was equal to 700 pixels. These images presented a variety of appearances of wrinkles. After running experiments, we separated 18 images with low detection results to highlight the typical challenges faced by a detection algorithm. We plotted the detection and false alarm ratios for the rest of 105 images shown in Figure 3.10. The detection rate varied from 25% to 85% with the typical rate in the range of 60–75% whereas the false alarm rate varied from 0.01% to 0.33% with the typical rate in the range of 0.05 – 0.15%. Based on visual observation of images, qualitatively, we sorted images in three different categories of having good, intermediate or fair



Figure 3.12: A few typical examples of images with a detection rate in the range of 50 – 65%. (Left) Original (Middle) Hand-drawn (Right) Automatically Localized visual results where low false alarm rate is as important as high detection rate in determining the visual quality of results. The qualitative categories of images can loosely be represented as regions in the plot in Figure 3.10.

Figure 3.11 includes a few examples of images with high detection rates. Figure 3.12 shows some examples of medium detection rate and Figure 3.13 examples of low detection rates.

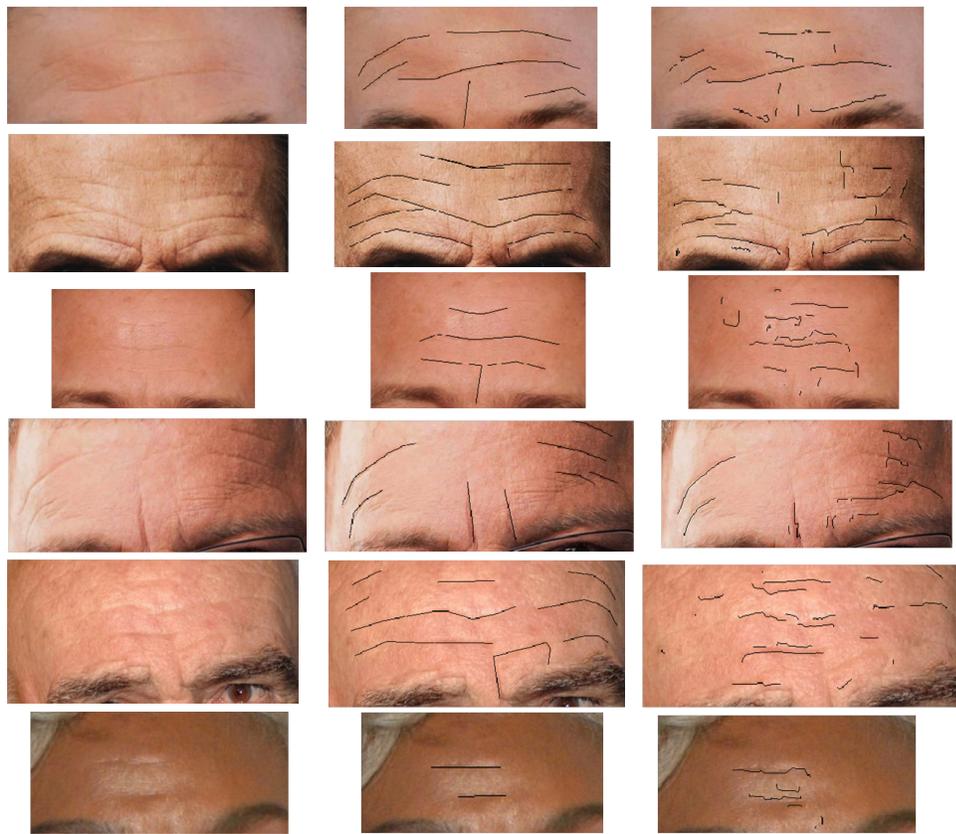


Figure 3.13: A few typical examples of images with a detection rate of $< 50\%$.

(Left) Original (Middle) Hand-drawn (Right) Automatically Localized

3.4.3.1 Challenges

We observed that the following factors contributed to the degradation of localization results.

Very light wrinkles: The most common reason for low detection rates was wrinkles' being very light. Figure 3.14 illustrates some typical examples. It can be seen in the left column of the figure that in such cases, the intensity gradients caused by the granular skin texture are comparable to those caused by wrinkles. Consequently, the algorithm picks non-wrinkle sites as wrinkles as well.



Figure 3.14: A few typical examples of images with very light wrinkles. (Left) Gabor features in grayscale. (Middle) Hand-drawn wrinkles. (Right) Automatically localized wrinkles.

Skin discolorations: This problem was more pronounced in subjects of lighter skin color which is more prone to having moles and brown/dark spots. The intensity gradients due to the 2D skin features of discoloration are mixed with those due to the 3D skin features of wrinkles and cause erroneous maxima locations in Gabor features. Some examples are shown in Figure 3.15. Although our algorithm discards most of such blob-like maxima using eccentricity property of connected components, a few linear maxima are wrongly classified as wrinkles.

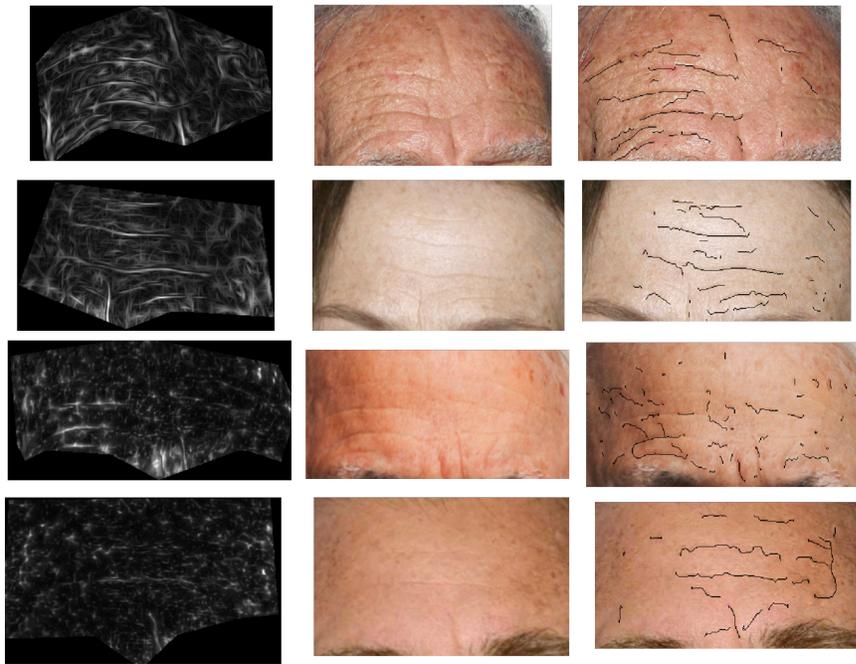


Figure 3.15: A few typical examples of images with skin discolorations masking wrinkles. (Left) Gabor features in grayscale. (Middle) Hand-drawn wrinkles. (Right) Automatically localized wrinkles.

Illumination and bright spots: Although we selected images with minimum illumination variation from the Internet, due to the uncontrolled acquisition settings, most of the images had some illumination variation. The variation caused ‘bright spots’ in images where light is reflected from skin surface due to oily skin or skin texture. In some cases, such bright spots caused false intensity gradients. Figure 3.16 illustrates two examples where illumination highlights irregular skin texture and causes bright spots highlighting erroneous intensity gradients in Gabor images.



Figure 3.16: Two examples of images with variant illumination. (Left) Gabor features in grayscale. (Middle) Hand-drawn wrinkles. (Right) Automatically localized wrinkles.

3.4.4 Comparison with Detection Results using MPP Modeling in Low Resolution Images

Next we conducted experiments on the dataset used in our previous work of detection of wrinkles using MPP modeling [1]. The dataset consisted of 36 images with hand-drawn wrinkles as ground truth. The image size varied from 86 x 37 as minimum to 290 x 110 as maximum. Images were not resized in this experiment for a better comparison with MPP modeling results. There was a significant improvement in runtime from 65 seconds on average in case of MPP modeling to 9 seconds on average in case of current algorithm. Figure 3.17 shows plots of detection/false alarm rates for the two methods where two distinct clusters of data points can be observed. MPP modeling classifies more image sites as wrinkles increasing both detection and false alarm rates whereas the current algorithm has consistent false alarm rate around 0.01% with varying detection rate. Qualitatively, this results in a visually better localization results with less false alarm rate. Figure 3.18 includes

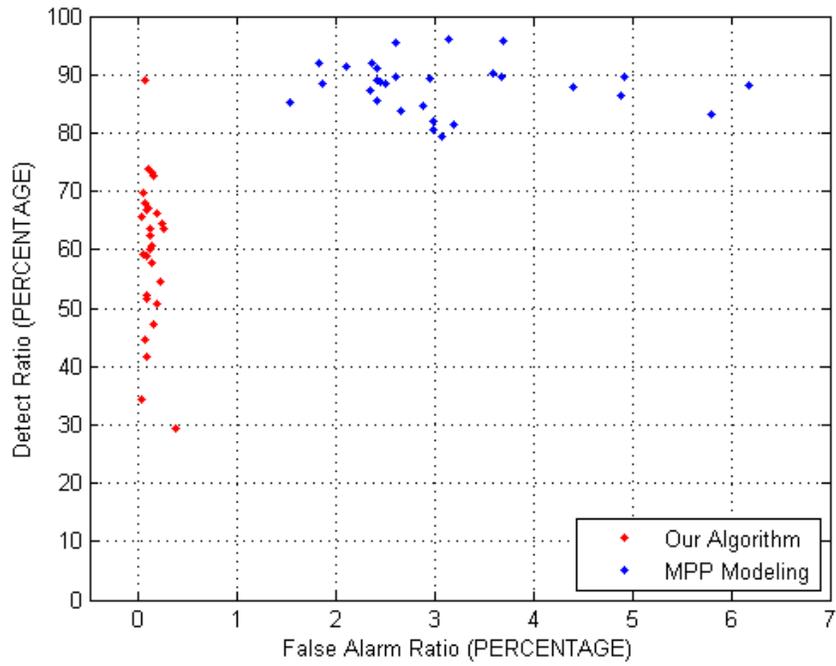


Figure 3.17: Plots of detection results using current algorithm and MPP modeling for low resolution images used in [1].

example images of five subjects with localization results by both approaches where the superior performance of the current algorithm can be observed.

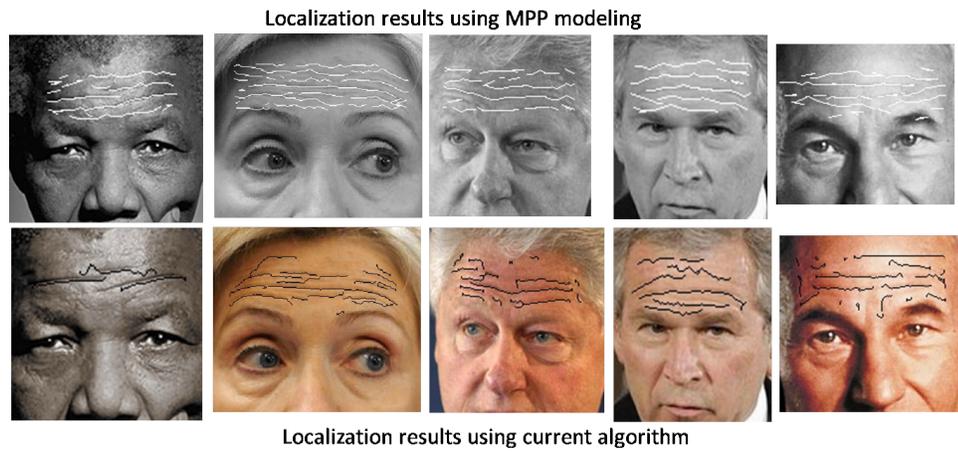


Figure 3.18: Visual comparison of detection results using our algorithm and MPP modeling for low resolution images used in [1].

Chapter 4: Assessment of Discriminative Power of Facial Wrinkles as Curve Patterns

4.1 Motivation

Recently a new area of research in face recognition has focused on analysis of facial features other than typical features (e.g. eyes, nose, mouth, chin, ears, texture, geometry). These new features, called facial micro-features, facial marks or facial soft biometrics, include but are not limited to scars, freckles, moles, facial shape, skin color, hair color, facial hair, tattoos, eye color, shape of nose, beard, mustache and wrinkles [24–29]. Detection and analysis of these features have become possible owing to the availability of high resolution, real-time capturing devices. For example, facial freckles, moles and scars have been used in conjunction with commercial face recognition software for face recognition in cases of occlusion and pose [24, 26]. Another interesting application is presented in [28] where recognition between identical twins was done using proximity analysis of manually annotated facial marks along with other typical facial features. Miller et al. evaluated the discriminative power of local texture of periocular (around eye) region vs. full facial texture as a soft biometric trait [25]. A combination of traits of skin, hair, eye color and presence

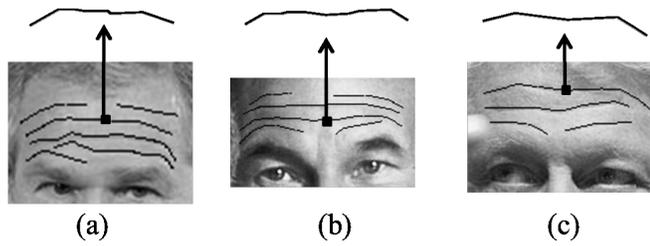


Figure 4.1: Different wrinkle patterns with the similar looking curve for three subjects: The curve has been highlighted.

of glasses, beard and mustache were used by Ouaret et. al [29] to reduce the search size in a database for face recognition. Although, the above is not an extensive list of the recent work, it shows the trend and potential in using other facial features for different applications.

The focus of this work is the evaluation of the discriminative power of wrinkles in human faces and specifically in forehead areas, as soft biometrics. Where the uniqueness of the location of facial marks e.g. moles and scars is very obvious, the same uniqueness of wrinkles is not that obvious and has been an unaddressed question so far. The assumption of similarity of wrinkles on the forehead and in areas around the eyes and nose have widely been used in facial aging simulation. For example, few general sets of wrinkles were used in [8] to simulate wrinkles on aging faces. An interesting and contradictory observation comes from a very different area of portrait drawings, sketching, caricatures, etc. of human figures which, most of the time, include sets of wrinkles very specific to that person. This motivates us to ask the question if a set of wrinkles has discriminative power enough to be used as a soft biometric. Usually the uniqueness of the facial wrinkles is not very

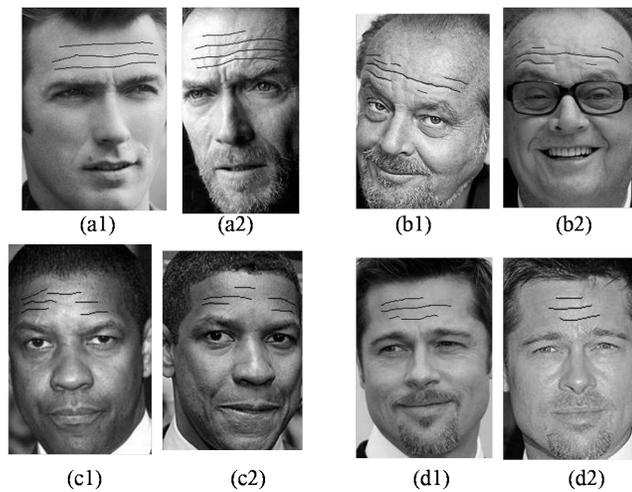


Figure 4.2: Images of four subjects with variations in wrinkle patterns due to age, expression and/or image acquisition.

obvious because of two reasons (a) facial wrinkles tend to appear in similar areas of the face i.e. forehead, eye, mouth for most people and (b) the curvature of wrinkle curves is similar in these locations because of factors such as deformations of similar facial muscles for expressions, etc. Hence different subjects are probable to have similar wrinkles in similar facial areas. We propose the hypothesis that, although individual wrinkles may be similar in different people (e.g. forehead wrinkles, crow feet), *a set* of several wrinkles as a pattern can be unique to an individual. For example, in Figure 4.1, three subjects are shown to have one very similar wrinkle curve but quite different overall wrinkle patterns. This motivates us to exploit the relative locations of wrinkle curves as a discriminative feature for wrinkle patterns.

The problem of wrinkle pattern recognition is challenging due to the large intra class variability caused by several factors. Image acquisition settings play an important role in the appearance of skin texture and, as a result, visibility of wrinkles. The

presence of expressions and/or pose increases the intra class variability even further by causing spatial displacements of wrinkles as well as changes in their curvatures. Figure 4.2 shows examples of differences in wrinkle patterns for four subjects. We believe that any attempt at recognizing wrinkle curve patterns has to address the following variations in curve patterns for a single subject.

1. Missing Curves
2. Discontinuous/Broken Curves
3. Deformed Curves

4.1.1 Related Work

The wrinkle pattern matching problem can be posed as one of matching two sets of spatially oriented curves. An active area of research is curve and shape matching in the presence of distortions and affine transformations, However, the focus in this research is the recognition of a single open/closed curve and not a set of curves as in our case. A more related, and relatively recent, area of research in computer vision community is object recognition/localization using a set of curves or lines [30–32]. For example, Yu and Leung extended the idea of matching points using Hausdorff distance to matching sets of lines/curves to recognize logos, palm prints and stationery characters. Sets of line segments were also used by Guerra and Pasucci to recognize 3D objects using Hausdorff distance between line segment sets. The method was used to extract specific 3D shapes/line patterns from a given image. In this work, we define and investigate several metrics, mostly based on Hausdorff

distance, to match the wrinkle patterns. Our main contribution is the assessment of the discriminative power of the wrinkles on images with uncontrolled acquisition settings. The recognition rate with no facial information other than wrinkle patterns is quite promising. We also present experiments on wrinkles *detected automatically* in some images instead of hand drawn wrinkles. The analysis of person specific wrinkle patterns can also be used in further applications regarding individual aging patterns in the future.

4.2 Wrinkle Pattern Matching

In this section, we present the methodology used for wrinkle pattern matching. We take a two step approach to evaluating the similarity between two wrinkle patterns:

1. Find one-to-many curve correspondences between two curve patterns.
2. Given curve correspondences, calculate the overall distance/similarity between two patterns by combining the distance of each individual correspondence. We call it the *Wrinkle Pattern Distance*($\mathbf{d}_{\mathbf{WPD}}$).

Let the two input binary images representing two wrinkle curve patterns be denoted by $\{\mathbf{I}(x, y); 1 \leq x \leq M_1, 1 \leq y \leq N_1\}$ and $\{\mathbf{J}(x, y); 1 \leq x \leq M_2, 1 \leq y \leq N_2\}$. In binary images the sites corresponding to the wrinkles have value 1 and 0 otherwise. The set of wrinkle curves is represented by \mathbf{V} where each $v_i \in \mathbf{V}$ represents one curve and the coordinates for the image sites belonging to the curve v_i are given by $S_i^v = \{s = (x, y), s \in \mathbb{R}^2\}$. For the rest of the chapter, let us denote the Euclidean

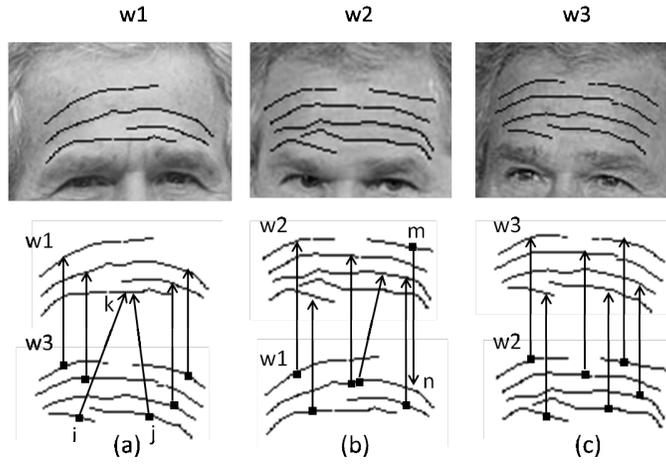


Figure 4.3: Three different patterns of wrinkles for the same subject and right correspondences between curves based on spatial proximity.

distance between two points $a \in \mathbb{R}^2$ and $b \in \mathbb{R}^2$ by $\mathbf{d}_E(a, b)$. We present the above mentioned two steps in detail in following sections.

4.2.1 Resolving Node/Curve Correspondences

The first task in wrinkle pattern matching is to determine the correspondences between curves in two wrinkle patterns. The correspondences are decided upon some similarity metric from the three metrics presented in the next section. Figure 4.3 shows an example of three wrinkle patterns for the same subject. Two of the patterns have 6 curves each whereas one pattern has four curves only. Ideally we want our algorithm to achieve right correspondences as shown in Figure 4.3 which requires many-to-one matching. In Figure 4.3(a) the wrinkles ‘i’ and ‘j’ are matched to the wrinkle ‘k’ due to spatial proximity. But in Figure 4.3(b) the extra wrinkle ‘m’ is matched to ‘n’ which is already matched to two other wrinkles. This eventually

increases the mismatch between two patterns. We resolve the curve correspondences as a bipartite graph matching problem by building a fully connected bipartite graph, $G_B^{(1,2)}$. The two sets of curves from two patterns represent two sets of nodes and the calculated distances between curves represent edge weights between nodes. Here we would like to mention that the node correspondence problem is different from the typical bipartite graph matching as follows:

1. One popular problem in bipartite graph matching is called ‘minimum weight perfect matching’ where, given equal number of nodes in each partition, every node is matched to exactly one other node while minimizing global edge weights or some other cost function. Our problem does not require the minimization of the overall edge weight functions.
2. When the number of nodes in two partitions is different or when the edges are not enough inherently for one-to-one matching, the problem is posed as minimum weight constricted (non-perfect) matching. In this case the graph is fully connected and many to one matching is allowed also to provide for the presumable situations where a wrinkle is detected as one curve in one image vs. more than one curves in the other.

This leads us to the following statement for the node correspondence problem.

Given a bipartite graph, not necessarily having the same number of nodes in each partition, find an edge for every node with minimum weight until all nodes have been covered.

Algorithm 1 Finding node-to-node correspondences

```
1: procedure ALGO2( $\mathbf{d}_v(i, j), \mathbf{C}$ )
2:    $\mathbf{d}_{sorted} \leftarrow$  sorted array of  $\mathbf{d}_v(i, j)$  for all nodes  $i \in \mathbf{V}^{(1)}$  and  $j \in \mathbf{V}^{(2)}$  in
   ascending order;
3:    $\mathbf{C} \leftarrow []$ ;
4:   Step 1:
5:   while  $k \leq \text{length}(d_{sorted})$  do
6:     if still some nodes in  $(\mathbf{V}^{(1)}, \mathbf{V}^{(2)})$  to be visited then
7:        $(i, j) \leftarrow$  nodes corresponding to the edge weight  $\mathbf{d}_v(i, j) = \mathbf{d}_{sorted}(k)$ ;
8:       if  $i$  not visited OR  $j$  not visited then
9:         include  $(i, j)$  to  $\mathbf{C}$ ;
10:      end if
11:     else
12:       break;
13:     end if
14:   end while
15:
16:   Step 2: Erase extra node associations
17:   for  $k = \text{length}(\mathbf{C})$  to 1 do
18:      $(i, j) \leftarrow \mathbf{C}(k)$ ;
19:     if both  $i, j$  are included in  $> 1$  correspondences then
20:        $\mathbf{C}(k) \leftarrow []$ ;
21:     end if
22:   end for
```

Since the number of wrinkles in every pattern is low, we use a greedy approach to solve the problem. Figure 1 presents our algorithm for finding node correspondences. The algorithm has two main steps. At step 1, the edges having at least one node not visited, are included in the increasing order the of edge weight. At step 2, the redundant edges are discarded by erasing the edges having both nodes visited more than once in the decreasing order of the edge weight. Figure 4.4 shows the step-by-step illustration of our algorithm and Figure 4.5 show the result of a typical matching algorithm. Our algorithm is different from a typical algorithms at steps (c) and (d) specifically. At step (c) the edge between $v_2^{(1)}$ and $v_1^{(2)}$ is selected, despite

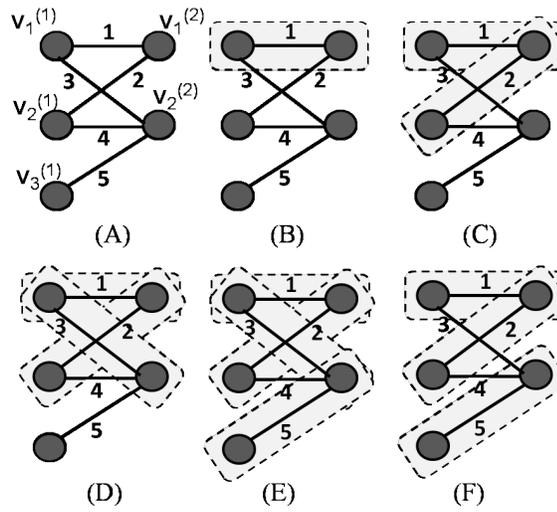


Figure 4.4: Step-by-step illustration of our algorithm for finding node correspondences.

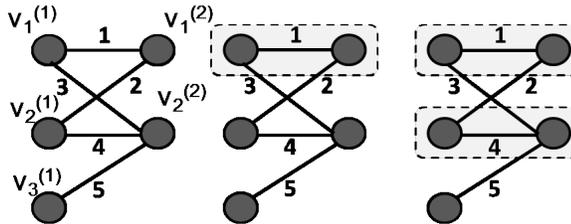


Figure 4.5: Step-by-step illustration of a generic bipartite graph matching algorithm with global optimization.

node $v_1^{(2)}$ having been included, over the edge between $v_2^{(1)}$ and $v_2^{(2)}$ due to the lowest edge weight of 2 among the edges connected to $v_2^{(1)}$. At step (f) the redundant edge between $v_1^{(1)}$ and $v_2^{(2)}$ is discarded. Although our algorithm results in larger global weight of 8 vs. 5 for the typical algorithm, our algorithm results in right spatial correspondences between curves as can be seen in Figure 4.6.

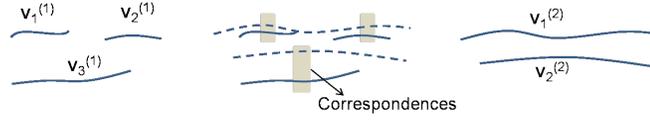


Figure 4.6: The curve correspondences achieved using our algorithm.

4.2.2 Wrinkles Pattern Similarity Metrics

In this section, we present four different metrics to compare two curve patterns. We start with modified Hausdorff distance [33] which has been widely used as a metric for object recognition based on lines/contours/curves [30–32, 34]. Then we present three more metrics based on comparison of pairs of curves in curve correspondences. These metrics compare curve to curve differences in spatial location or shape instead of binary images as a whole as is done in calculation of modified Hausdorff distance. Following are the detailed descriptions of the four metrics:

4.2.2.1 Modified Hausdorff Distance \mathbf{d}_{MHD}

Given two binary images $\mathbf{I}(x, y)$ and $\mathbf{J}(x, y)$, let $S_{\mathbf{I}} = \{(x, y); \mathbf{I}(x, y) = 1\}$ and $S_{\mathbf{J}} = \{(x, y); \mathbf{J}(x, y) = 1\}$. Then the modified Hausdorff distance $\mathbf{d}_{MHD}(\mathbf{I}, \mathbf{J})$ between two images is given as:

$$\mathbf{d}_{MHD}(\mathbf{I}, \mathbf{J}) = \max(\mathbf{d}_D(\mathbf{I}, \mathbf{J}), \mathbf{d}_D(\mathbf{J}, \mathbf{I})) \quad (4.1)$$

where the directed distance $\mathbf{d}_D(A, B)$ is given as follows.

$$\mathbf{d}_D(A, B) = \frac{1}{|A|} \sum_{a \in A} \min_{b \in B} \mathbf{d}_E(a, b). \quad (4.2)$$

4.2.2.2 Curve Proximity Distance \mathbf{d}_{CPD}

We introduce the metric *Curve Proximity Distance* to quantify the spatial proximity of two curves. The main difference between \mathbf{d}_{CPD} and \mathbf{d}_{MHD} is the inclusion of structure of curves. Given curve correspondences, \mathbf{d}_{CPD} is the sum of individual distances for the two curves in a curve correspondence instead of max-min distance for the binary images as a whole in \mathbf{d}_{MHD} . As we will see in experiments section, the inclusion of curve structure generally improves recognition rates.

Let l_i and l_j be the lengths of two curves and let $l_{min} = \min(l_i, l_j)$. Then the Curve Proximity Distance, $d_{CPD}(i, j)$, between the two curves is defined as

$$\mathbf{d}_{CPD}(i, j) = \max(\mathbf{d}_D(S_i^{(1)v}, S_j^{(2)v}), \mathbf{d}_D(S_j^{(1)v}, S_i^{(2)v})). \quad (4.3)$$

4.2.2.3 Directed Curve Proximity Distance \mathbf{d}_{DCPD}

Figure 4.7 shows the motivation behind introducing this metric by highlighting the difference between the distance \mathbf{d}_D calculated from shorter curve to longer curve (Figure 4.7(a)) and vice versa (Figure 4.7(b)). This metric allows us to investigate situations where one wrinkle may be represented as one curve in one image vs. more in the other by restricting the metric from being unnecessarily large as can be seen in (Figure 4.7(c,d)). The $\mathbf{d}_{DCPD}(i, j)$ is defined as follows where $l_{min} = \min(l_i, l_j)$.

$$\begin{aligned} \mathbf{d}_{DCPD}(i, j) &= \mathbb{I}(l_{min} = l_i) \mathbf{d}_D(S_i^{(1)v}, S_j^{(2)v}) \\ &\quad + \mathbb{I}(l_{min} = l_j) \mathbf{d}_D(S_j^{(1)v}, S_i^{(2)v}). \end{aligned} \quad (4.4)$$

4.2.2.4 Curve Shape Distance \mathbf{d}_{CSD}

This metric compares the two curves for their curvature similarity while ignoring their spatial location in the two curve patterns. The distance is calculated in the same way as of calculating curve proximity distance, however, the curves are aligned with each other. For example, Figure 4.8 shows calculation of \mathbf{d}_{CSD} for two curves after aligning them as compared to calculation of \mathbf{d}_{DCPD} without any alignment. The 2D correlation function is used for alignment. For any two nodes $v_i^{(1)} \in \mathbf{V}^{(1)}$ and $v_j^{(2)} \in \mathbf{V}^{(2)}$, the distance $\mathbf{d}_{CSD}(v_i^{(1)}, v_j^{(2)})$ is calculated in the following steps.

1. Construct two images \mathbf{I}^* and \mathbf{J}^* from input images such that they have the curves $v_i^{(1)}$ and $v_j^{(2)}$ only respectively.

$$\mathbf{I}^*(x, y) = \begin{cases} 1 & \text{if } (x, y) \in S_i^{(1)v} \\ 0 & \text{otherwise} \end{cases} \quad (4.5)$$

$$\mathbf{J}^*(x, y) = \begin{cases} 1 & \text{if } (x, y) \in S_j^{(2)v} \\ 0 & \text{otherwise} \end{cases} \quad (4.6)$$

2. Use 2D correlation function to find $s^* = (u^*, v^*)$ as follows

$$s^* = \arg \max_{(u,v)} \sum_{x=1}^{M_1} \sum_{y=1}^{N_1} \mathbf{I}^*(x, y) \mathbf{J}^*(x + u, y + j);$$

$$1 \leq u \leq M_1 + M_2, 1 \leq v \leq N_1 + N_2 \quad (4.7)$$

3. Translate the image \mathbf{J}^* by s^* i.e. $\mathbf{J}_t^*(x, y) = \mathbf{J}^*(x + u^*, y + v^*)$. Thus the curve

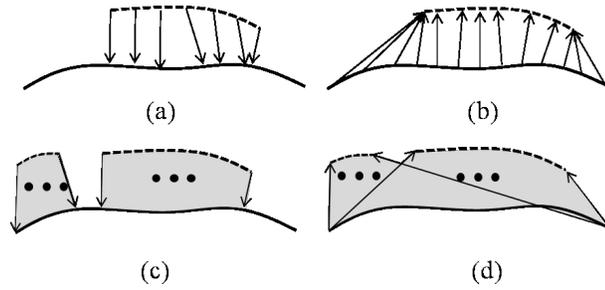


Figure 4.7: Difference between shorter to longer vs. longer to shorter distances in \mathbf{d}_{DCPD} .



Figure 4.8: (Left) Two registered curve patterns (Middle) Calculation of \mathbf{d}_{CSD} (Right) Calculation of \mathbf{d}_{DCPD} .

$v_j^{(2)}$ will also be translated in \mathbf{J}^* . Let us denote the translated curve in \mathbf{J}_t^* as $v_{j^*}^{(2)}$ and the curve in \mathbf{I}^* as $v_{i^*}^{(1)}$.

4. Then the $\mathbf{d}_{CSD(i,j)}$ is equal to the the curve proximity distance \mathbf{d}_{CPD} between $v_{i^*}^{(1)}$ and $v_{j^*}^{(2)}$.

The overall similarity metric between a probe and a gallery wrinkle pattern, Wrinkle Pattern Distance, \mathbf{d}_{WPD} , is the sum of the distance metric for of all curve correspondences as follows.

$$\mathbf{d}_{WPD}(\mathbf{V}^{(1)}, \mathbf{V}^{(2)}) = \sum_{c=(i,j) \in \mathbf{C}} \left(\mathbf{d}_v(v_i^{(1)}, v_j^{(2)}) \right). \quad (4.8)$$

where $\mathbf{d}'(\cdot, \cdot)$ can be any or a combination of \mathbf{d}_{CSD} , \mathbf{d}_{CPD} and \mathbf{d}_{DCPD} .

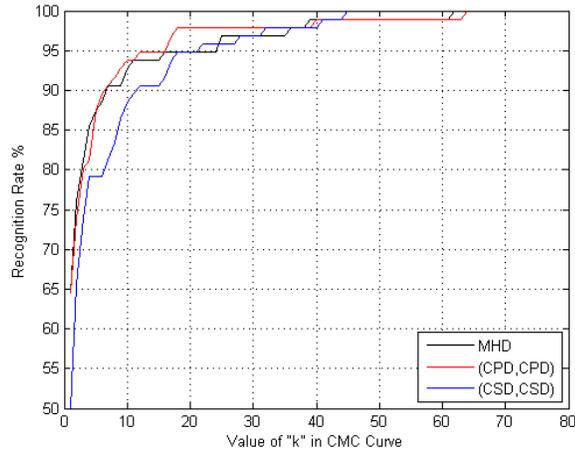


Figure 4.9: Comparison of recognition rates for hand-drawn wrinkles with different methods.

4.3 Experiments and Discussion

To the best of our knowledge, no data set is available in the computer vision community for faces with marked wrinkles. For this work, the data were gathered by selecting images of medium resolution of well-known people from the Internet and consisted of variations of illumination, acquisition setup, pose, expressions and age. The data set comprised of 96 images of 16 subjects with 6 images per subjects. The face images of resolution greater than 200x200 were preferred. The wrinkles were hand drawn by 4 different users. This also included ‘subjective’ variation in perception of the wrinkles. The images were registered by the selection of five land marks on faces i.e. two corners of both eyes and the nose tip. As a next step, the wrinkle curves were separated from a binary image by finding connected components and some morphological processing.

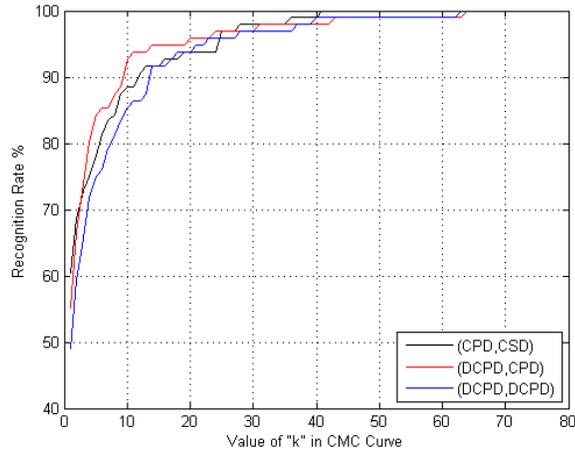


Figure 4.10: Comparison of recognition rates for hand-drawn wrinkles with different methods.

4.3.1 Experiments on Hand-drawn Wrinkles

Several experiments were conducted on the data set of hand drawn wrinkles with different combinations of the metric used to find correspondences and the distance between curve pairs in the correspondences. The cumulative match curves for these experiments are shown in Figures 4.9, 4.10 and 4.11. In the plot legend ‘(a,b)’ means that the metric ‘a’ has been used to find curve correspondences in algorithm 1 and ‘b’ has been used to calculate \mathbf{d}_{MPD} . Table 4.1 shows the percentage recognition rates for top 5 ranks. We can see that the methods MHD and (CPD,CPD) have comparable results and better than the rest of the methods. For further investigation the images where MHD performed better than (CPD,CPD) and vice versa were examined. Figure 4.12 shows four pairs of wrinkle patterns where (CPD,CPD) was able to recognize correctly in contrast with MHD and Figure 4.13 shows four pairs

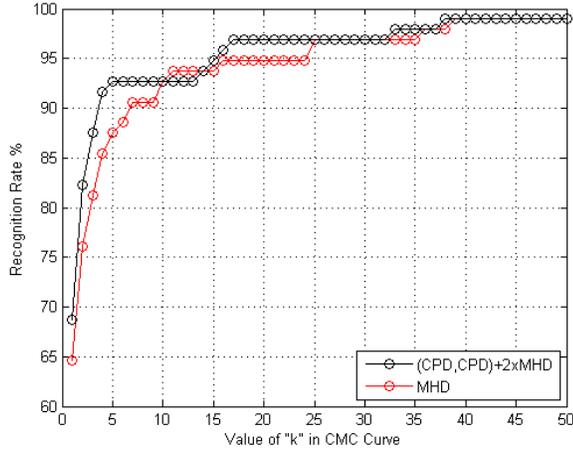


Figure 4.11: Comparison of recognition rates for top two methods.

of wrinkle patterns where MHD was able to recognize correctly in contrast with (CPD,CPD). As a next step, we combined both MHD and (CPD,CPD) i.e. \mathbf{d}_{CPD} was used for finding curve correspondences and $\mathbf{d}_{CPD} + 2 * MHD$ was used to calculate \mathbf{d}_{MPD} . The combination improved the recognition rate by 5-6% as can be seen in Figure 4.11 and in the last row of Table 4.1.

4.3.2 Experiments on Automatically Detected Wrinkles

As a next step, we repeated the experiments on automatically detected wrinkles. Recently the work in [2] reported automatic detection of wrinkles as line segments with average detection rate of 80%. We wanted to investigate if the wrinkle patterns recovered with this detection rate still retain enough discriminative power. However, the work in [2] detects wrinkles as line segments and the sequences of line sequences are broken at times. We fit curves through line segments to create wrinkle curves. The inclusion of line segments in a curve, however, is not trivial, and requires analysis

Algorithm 2 Fitting Curves to Line Segments representing Detected Wrinkles

```
1: procedure ALGO1 (LineSegmentsVector, WrinkleCurveVector)
2:   WrinkleCurveVector  $\leftarrow$  [];
3:   while  $k \leq \text{length}(\textit{LineSegmentsVector})$  do
4:     seg  $\leftarrow$  LineSegmentsVector( $k$ );
5:     CurvePoints  $\leftarrow$  [];
6:     if seg not visited then
7:       newSeg  $\leftarrow$  seg;
8:       go to 18;
9:       repeat
10:        newSeg  $\leftarrow$  The segment connected at RIGHT to newSeg OR the
segment present in RIGHT neighboring conic region of newSeg;
11:        go to 18;
12:        until border of image reached
13:        repeat
14:         newSeg  $\leftarrow$  The segment connected at LEFT to newSeg OR the
segment present in LEFT neighboring conic region of newSeg;
15:         go to 18;
16:         until border of image reached
17:        end if
18:        Add pixels of newSeg to curvePoints;
19:        Mark newSeg as visited; return
20:        Mark seg as visited;
21:        Append curvePoints to WrinkleCurveVector;
22:     end while
23: end procedure=0
```

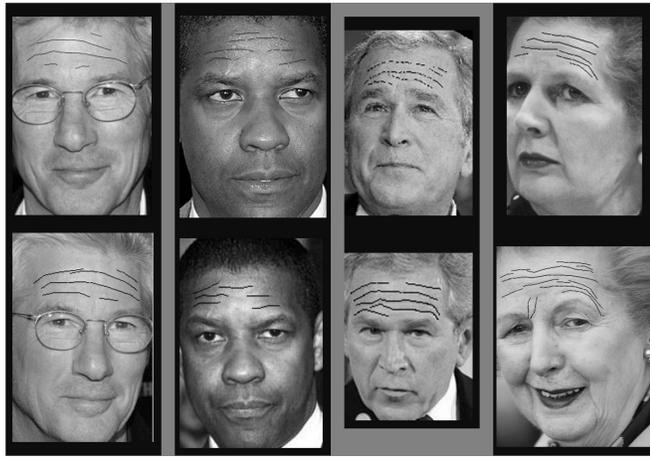


Figure 4.12: Four pairs of images where (CPD,CPD) performed better than MHD.

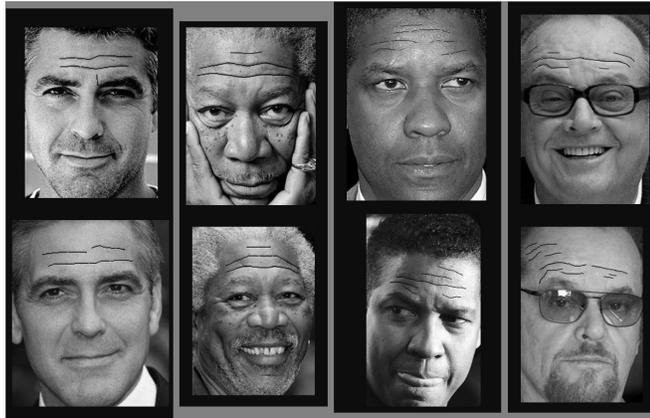


Figure 4.13: Four pairs of images where MHD performed better than (CPD,CPD).

of the proximity of the line segments. Figure 2 presents our algorithm to include line segments to a curve. The vector of line segments is scanned and a new curve is added whenever an unvisited segment is reached. A sequence of connected line segments is included to a single curve by default. Then a conic region of a certain length and angle is searched around each free end point of a sequence of line segments for any neighboring segments to be included to the same curve. The algorithm outputs the data structure where each node represents a group of line segments to be included



Figure 4.14: (Curves fitted to detected wrinkles as line segments (images were taken from [2]).

in a single curve. Figure 4.14 shows some examples of the curves fitted to the line segments.

For detected wrinkles the experiments were conducted on a data set of 12 images with 3 images per subject. Figures 4.15, 4.16 and 4.17 show plots of the results with different combinations. In case of detected wrinkles, we can observe that the metric combination based on shape, (CSD,CSD), is performing better than MHD and the metric combination of (CPD,CPD) both of which had the best performance rates for hand drawn wrinkles. However the detection rate in this case is lower, 50%, as compared to 64% for hand drawn wrinkles. When we combine both (CSD,CSD) and MHD the recognition rate improves to 90% for top 3 rank positions as can be seen in Figure 4.17.

Recognition Rate					
Method	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5
MHD	64%	76%	81%	85%	87%
(CPD,CPD)	64%	74%	80%	81%	88%
(CSD,CSD)	50%	66%	74%	79%	79%
(CPD,CSD)	60%	69%	73%	75%	78%
(DCPD,CPD)	55%	67%	74%	80%	84%
(DCPD,DCPD)	49%	59%	65%	72%	75%
(CPD,CPD)+2xMHD	69 %	82%	87%	92%	93%

Table 4.1:

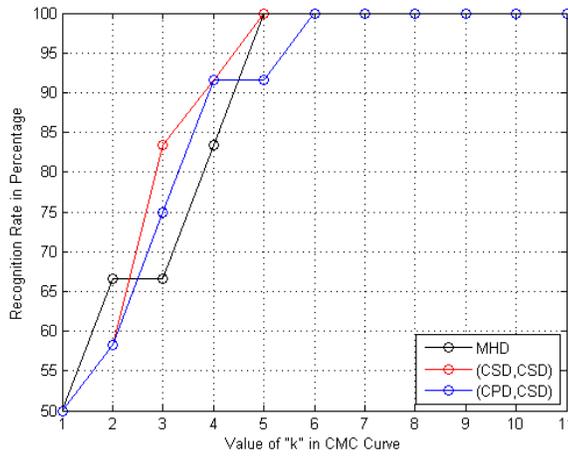


Figure 4.15: Comparison of recognition rates for detected wrinkles with different methods.

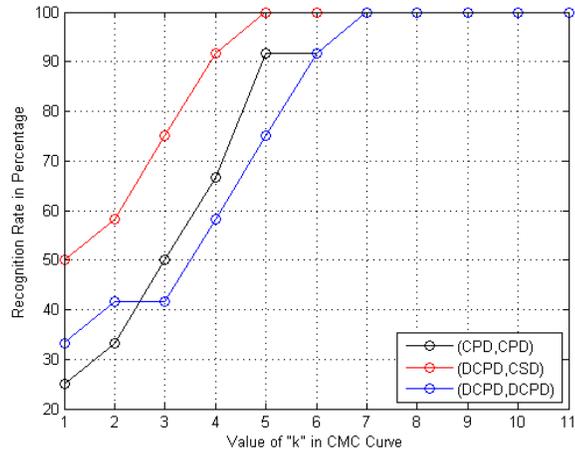


Figure 4.16: Comparison of Recognition Rates for detected wrinkles with different methods

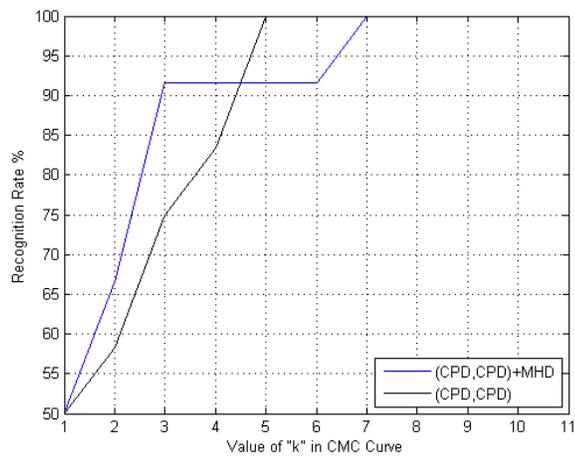


Figure 4.17: Comparison of recognition rates for top 2 methods for detected wrinkles.

Chapter 5: Simultaneous Detection and Removal of Wrinkles from Facial Images using Image Inpainting

5.1 Motivation

Digital image inpainting refers to the filling of the gaps of arbitrary shapes in an image so that they seem to be parts of the original image. Several applications of digital inpainting have been reported in the last decade. Here we propose a specific application of digital inpainting to remove facial wrinkles and imperfections. Traditionally, beautification of skin or facial re-touching in images has been done by professionals using high-end software e.g. Adobe PhotoshopTM. However, the application has become more popular in recent years due to the popularity of smart phones and social networking. As a result, several user friendly smart phone applications (e.g. Visage LabTM [4], BeautifyTM [35], Perfect365TM [36]) have been introduced which provide minimum user interaction for facial touch ups. However, both professional and user-friendly software have limitations. Professional software require significant user interactions where results are subjective, depending on user's expertise. Whereas user-friendly applications targeted for smart phones, while performing an overall beautification or making up of skin with minimum user



Figure 5.1: Typical results of facial retouching (reproduced from [3]).



Figure 5.2: Removal of wrinkles by facial retouching (reproduced from [3]).

interaction, do not target specific skin imperfections e.g. deep wrinkles, acne, scars etc. An example is shown Fig. 5.5, where an overall beautification of skin fades wrinkles and moles but does not remove them completely. The reason may be that these applications seem to process all the skin region equally and do not distinguish between skin vs. skin imperfections. The results can be improved if skin imperfections are detected as a pre-processing step and then processed differently from the surrounding skin.

The current state-of-the-art approach for the removal of wrinkles is an image *painting* algorithm proposed by Georgiev [37]. The algorithm is based on the widely used Poisson image editing tool [38] and provides improved seamless image cloning



Figure 5.3: Reduction of wrinkles by facial retouching (reproduced from [3]).



Figure 5.4: Reduction of wrinkles by facial retouching (reproduced from [3]).

through better handling of lighting variations. The algorithm works behind the *Healing Tool* in Adobe PhotoshopTM. Image painting is slightly a different application from image inpainting. The former deals with inclusion (painting) of a smaller image region in a larger image where both source and destination image regions are provided by the user. The latter deals with the automatic filling of a gap/occlusion, mostly provided by the user, in an image based on local and/or global image characteristics and does not require a source image. However, both applications share the requirement of seamless boundaries. Our work is closer to image inpainting than image painting because both source and destination image areas are selected auto-

matically. We make the following observations about the current facial retouching software as a motivation for our proposed work.

1. Significant user interaction is required with the Adobe Healing Tool for the selection of source and destination skin patches resulting in subjective results depending on user expertise.
2. In the case of more user-friendly applications, facial retouching results in the so-called *flawless* skin. The processing of skin in an image smoothes wrinkles and skin imperfections but does not remove them completely.
3. Regarding image inpainting techniques, both structure and texture inpainting techniques are not applicable directly to skin. Wrinkles and skin imperfections do not appear as edges/boundaries and, hence, structural inpainting is not appropriate. Also, as wrinkles are not homogeneous texture patterns, texture inpainting is not effective.

The main contributions of this work are as follows:

1. An algorithm based on the fusion of Gabor features and texture orientation fields in the framework of Markov field modeling (MRF) is proposed to detect wrinkles and other imperfections in the surrounding skin.
2. A variation of exemplar-based texture synthesis is proposed to fill the gaps of irregular shapes.
3. Both detection and inpainting of wrinkles are unsupervised with minimum user interaction thus minimizing the subjectivity introduced by the user's expertise.

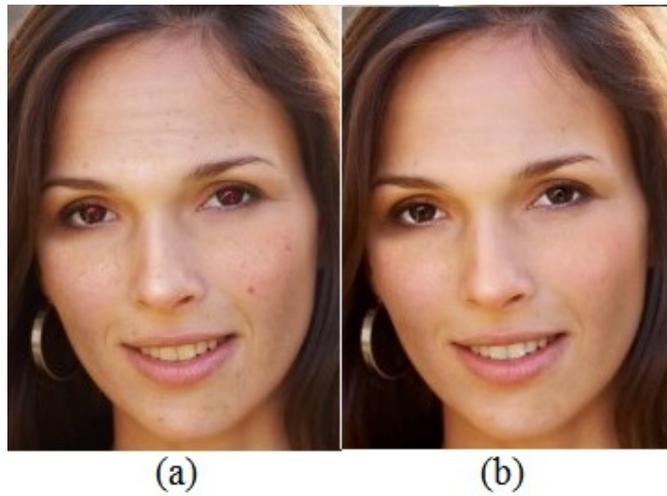


Figure 5.5: Typical results of facial retouching for a smart phone application [4]. (a) Original Image. (b) Image after retouching. Note that wrinkles on forehead and brown spots on cheeks are deemphasized due to blending but still visible.

4. No ‘retouching’ or ‘beautification’ of the rest of the facial skin is done while inpainting skin wrinkles/imperfections.

The organization of this chapter is as follows. In section 2, we present an overview of the related work. In section 3, we present the details of our detection and inpainting algorithms. Experiments and discussion are presented in section 4. Finally, we conclude the chapter in section 5.

5.2 Problem Statement

Any image inpainting technique for textures has three main steps, (a) finding a suitable texture template in the image (b) calculating the warping between the template and the gap (c) Inpainting the object via texture synthesis. For the particular application of wrinkle inpainting, finding the proper correspondences boils down to

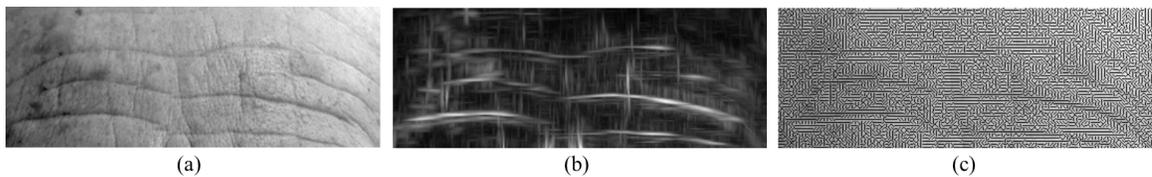


Figure 5.6: Image features used for segmentation. (a) Forehead image in gray scale. (b) Maximum Gabor amplitude response (values $[4.8, 132]$ scaled to the gray scale values $[0,255]$). (c) Texture orientation field.

automatic detection of regions having wrinkles. Then, based on the fact that skin texture can vary significantly within a face, the region has to be filled with the skin texture surrounding wrinkles while compensating for the global lighting variations. In the next section we present a brief review of recent work in image painting and inpainting domains related to our goal.

5.3 Related Work

Image inpainting methods target one or both of the *structure* and *texture* of an image. The difference between the image attributes of structure and texture of an image requires different inpainting methods. A detailed survey of image inpainting methods can be found in [39–41]. Most texture inpainting methods require user input or some masking function to highlight the gap/occlusion to be filled (e.g. the work by Criminisi [42]). Some examples of automatic filling of scratches, rectangular blocks or random noise can be found in [39, 43, 44]. Shi and Chang introduced a patch-based multi-resolution/multi-layer approach to restore the paintings damaged by red scratches [39]. Their approach involved a mechanism to detect the damaged

areas first where the variance in the color of a patch at a specific resolution was used to determine if a patch had damaged pixels. In contrast, the inpainting methods in [43, 44] do not have any explicit detection of gaps to be filled. These techniques are based on the analysis of different layers containing low vs. high frequency details. The low frequency layer determines the piecewise smooth regions of the image and the high frequency layer determines the texture. The recovery of these layers automatically fills the gaps without their being detected explicitly. In case where more than one texture is surrounding the gap, sophisticated techniques are used for combining different textures [42, 45]. Once a suitable combination of different textures has been found, the gap is filled by existing texture synthesis techniques. For example, Grossauer [45] used the exemplar-based texture synthesis technique given in [46] and Criminisi et al. [42] used a synthesis method similar to [47].

The specific application of wrinkle removal is different as wrinkles are not artifacts or separate objects to be removed. Wrinkles are an inherent part of the skin and are visible only due to their discontinuous nature in surrounding skin texture. Recently, the detection of wrinkles as sequences of line segments/curves was reported by Batool and Chellappa ([1, 2]). However, this method is not applicable here because of two reasons. First, wrinkles are localized as curves and the surrounding folds of skin due to a wrinkle are not detected. Second, the method reported in [1, 2] is based on line segments and cannot be used to detect other oval like skin imperfections.

Our wrinkle inpainting approach is based on Poisson editing and a variation of exemplar-based texture synthesis. However, we use a novel approach to detect

wrinkles and skin imperfections. In the following section we present our approach in detail.

5.4 Approach

An image inpainting technique for textures has three main steps, (a) finding a suitable texture template in the image to fill in the gap with, (b) calculating the seamless warping between the template and the gap and (c) filling the gap via texture synthesis. Since we are proposing unsupervised image inpainting, an additional step is required to detect wrinkles automatically. The process of wrinkling creates deep creases and causes curvature in the surrounding skin. The resulting skin curvature causes specific intensity gradients in skin images which look like discontinuities in surrounding skin textures. An accurate inpainting of wrinkles will require both the wrinkle crease and the surrounding curved skin to be removed. In section 5.4.1, we present our approach for detection. Regarding step (a), we select skin patches surrounding the detected wrinkles. This is due to the fact that the skin texture can vary significantly within a small region of face. The skin patches closest to the wrinkles have the most similar looking skin texture. Regarding steps (b) and (c), we use an exemplar-based texture synthesis method based on the work of Efros and Freeman [48]. The details of our texture synthesis method are presented in section 5.4.2.

5.4.1 Automatic Detection of Regions with Wrinkles

We use texture orientation fields proposed by Rao and Schunk [49] and Gabor filter responses as image features. The orientation fields highlight the discontinuity in the normal flow of skin texture whereas Gabor filter responses highlight the intensity gradients in any directions. The two types of features are fused using Gaussian Mixture Models (GMM) and Markov random field representation. The GMM classifies filter responses as a bimodal distribution for skin vs. skin imperfections. The MRF representation allows us to incorporate spatial relationship among GMM distributions of neighboring pixels and to fuse the orientation fields to reshape the class probabilities.

5.4.1.1 Computation of Orientation Fields using Gabor Filters

Several oriented feature detectors have been developed including steerable Gaussian second-derivative filters, line operators and Gabor filters. A comparative study can be found in [19] where the real Gabor filters were assessed to be the best detector of oriented features. The real Gabor filter kernel oriented at angle α is given by

$$g(x_1, x_2) = \frac{1}{2\pi\sigma_{x_1}\sigma_{x_2}} \exp \left[\frac{-1}{2} \left(\frac{x_1'^2}{\sigma_{x_1}^2} + \frac{x_2'^2}{\sigma_{x_2}^2} \right) \right] \cos(2\pi f x_1') \quad (5.1)$$

where

$$\begin{bmatrix} x'_1 \\ x'_2 \end{bmatrix} = \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (5.2)$$

Let $\{g_k(x_1, x_2), k = 0, \dots, K - 1\}$ denote the set of real Gabor filters oriented at angles $\alpha_k = -\frac{\pi}{2} + \frac{\pi k}{K}$ where K is the total number of equally spaced filters over the angular range $[-\frac{\pi}{2}, \frac{\pi}{2}]$. Let $\{I(x_1, x_2); x_1 = 1 \dots N_1, x_2 = 1 \dots N_2\}$ denote the input image in gray scale and $I_k^P(x_1, x_2)$ denote the image filtered by the filter $g_k(x_1, x_2)$. Then the orientation field, $\theta_I(x_1, x_2)$ for the image is computed as follows:

$$\theta_I(x_1, x_2) = \arg \max_k I_k^P(x_1, x_2) \quad (5.3)$$

i.e. at every pixel, the orientation field is equal to the orientation angle of the filter resulting in the maximum filtered response at that pixel. The corresponding maximum amplitude among the filtered responses is given as:

$$I'(x_1, x_2) = \max_k I_k^P(x_1, x_2) \quad (5.4)$$

The set of the maximum filter response and the orientation angle at every pixel, $\{I'(x_1, x_2), \theta_I(x_1, x_2)\}$, constitutes image features for automatic detection of wrinkle regions. Fig. 5.6 shows a forehead image with the corresponding maximum responses and the orientation field. The orientation angle is calculated at every pixel, however, the orientation field in Fig. 5.6(c) is drawn by placing needles at every 3rd pixel. Every needle is of length of 3 pixels and is placed in the direction of the orientation angle.

At high resolution, skin texture appears to be granular resulting in random orientation angles. However, the skin creases or wrinkles and the skin pigments related

to other imperfections (e.g. brown spot, moles) smooth out the granular skin texture. As a result, the orientation field depicts two significant properties in wrinkled regions, (a) a dominant angle of zero degrees and (b) pixels with zero orientation angle appear in clusters. Fig. 5.10 depicts these two properties of orientation field due to wrinkles. We exploit these observations to formulate the GMM-MRF model based, two-class labeling of images into wrinkles and non-wrinkle regions. The next section describes the model in detail.

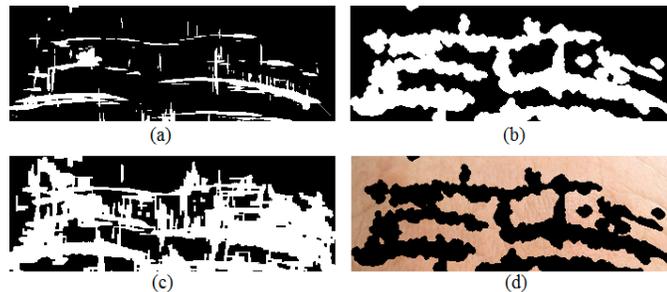


Figure 5.7: Binary labeling results. (a) Initial labels obtained after using GMM functions in MatlabTM. (b) labeling results after 15 iterations of our GMM-MRF method. (c) labeling results using GMM-MRF method in [5]. (d) Resulting gaps in the skin image.

5.4.1.2 Gaussian Mixture Model based on Markov Random Field (GMM-MRF)

The motivation behind using the GMM-MRF model is the fact that the Gabor filter responses or the texture orientation field, when used exclusively, are important but insufficient features to detect the wrinkled regions. For example, Fig. 5.8(a)

shows the result of thresholding Gabor amplitude responses in the range [4.8, 132] with the threshold value of 35 and Fig. 5.8(b) shows image sites with orientation angles lying in the range $[-5, +5]$. Fig. 5.8(c) shows the product of both results and resembles more closely to the actual wrinkles by reducing false positive in either of the Fig. 5.8(a) and Fig. 5.8(b). We make the following observations to justify GMM-MRF modeling.

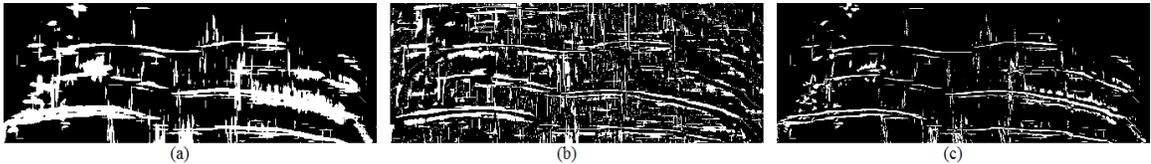


Figure 5.8: Results of thresholding. (a) Thresholding maximum Gabor amplitude at value 35. (b) Thresholding orientation field at absolute angle values of less than 5 degrees. (c) Product of images in (a) and (b).

1. GMM: Histograms of Gabor response amplitude $I'(x_1, x_2)$ typically follow the Beta distribution with heavy tails. For example Fig. 5.9 shows the histogram of the Gabor amplitude response for the image in Fig. 5.6(a). An intelligent thresholding of Beta distribution can provide a good starting point for any segmentation technique. Modeling of Gabor responses as Beta distribution may seem an obvious choice. However, we take the simpler approach of Gaussian mixture models for its more developed theory. A similar approach can be found in [50] where the authors used the GMM to model Beta distribution for segmentation of SAR images.
2. MRF: Since class labels do not depend solely on the Gabor response ampli-

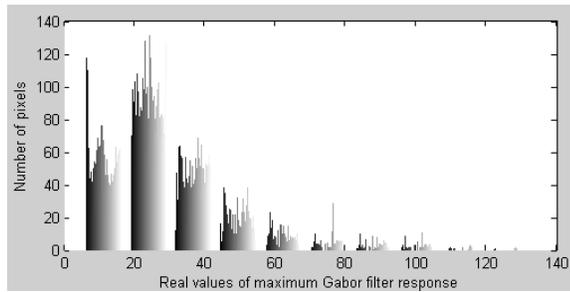


Figure 5.9: Histogram of the Gabor features in Fig. 5.6(b).

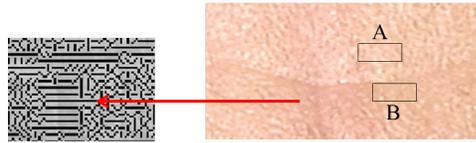


Figure 5.10: (Right) Rectangle 'A' shows the skin texture used as template whereas rectangle 'B' shows change in skin texture due to a wrinkle. (Left) Orientation field at high resolution, note that sites corresponding to the wrinkle have orientation angle of zero degrees.

tude, a simple thresholding of Beta distribution does not work. There is always some under segmentation or over segmentation present. Texture orientation field has to be incorporated to aid thresholding by reshaping the probability of each class. An MRF framework enables not only the incorporation of spatial dependencies among neighboring pixels but also the fusion of texture orientation fields and Gabor amplitude responses.

We first present the GMM and MRF models and then discuss how the MRF model is used to fuse the orientation field with Gabor amplitude responses.

Under GMM, the density function for the observation at pixel (x_1, x_2) is given

as:

$$f(I'(x_1, x_2)|\Pi, \Theta) = \sum_{j=1}^J \pi_{x_1, x_2}^j \Phi(I'(x_1, x_2)|\mu_j, \sigma_j) \quad (5.5)$$

where $\Phi(I'(x_1, x_2)|\mu_j, \sigma_j)$ is the standard Gaussian distribution with mean μ_j and variance σ_j^2 and $\Theta = \{(\mu_j, \sigma_j); j = 1, \dots, J\}$ is the parameter set of Gaussian mixture distributions. The set of mixing proportions, $\Pi = \{\pi_{x_1, x_2}^j; x_1 = 1, \dots, N_1; x_2 = 1, \dots, N_2; j = 1, \dots, J\}$ satisfies the following constraints:

$$0 \leq \pi_{x_1, x_2}^j \leq 1 \text{ and } \sum_j \pi_{x_1, x_2}^j = 1 \quad (5.6)$$

Let \mathcal{I}' denote the ensemble of random variables $I'(x_1, x_2)$ as follows:

$$\mathcal{I}' = \{I'(x_1, x_2); x_1 = 1, \dots, N_1; x_2 = 1, \dots, N_2\} \quad (5.7)$$

Then, assuming statistical independence of individual pixel sites, the joint conditional density function of the whole image can be written as:

$$p(\mathcal{I}'|\Pi, \Theta) = \prod_{x_1=1}^{N_1} \prod_{x_2=1}^{N_2} \sum_{j=1}^J \pi_{x_1, x_2}^j \Phi(I'(x_1, x_2)|\mu_j, \sigma_j) \quad (5.8)$$

According to Bayes' theorem, the posterior probability can be written as follows:

$$p(\Pi|\mathcal{I}', \Theta) \propto p(\mathcal{I}'|\Pi, \Theta) \times p(\Pi) \quad (5.9)$$

Gaussian mixture models based on MRF (GMM-MRF) are proposed to impose spatial smoothness constraints between neighboring pixels [5]. Under MRF models, the prior distribution of the mixing proportion of a pixel π_{x_1, x_2}^j depends on those

of its neighboring pixels. The prior joint distribution of π_{x_1, x_2}^j for all pixels is given by the Gibbs distribution:

$$p(\Pi) = \frac{1}{\mathbf{Z}} \exp\left(-\frac{\mathbf{U}(\Pi)}{T}\right) \quad (5.10)$$

where \mathbf{Z} is the normalization constant, $\mathbf{U}(\Pi)$ is the Gibbs energy function and T is a constant called temperature. According to (5.8), (5.9) and (5.10), the posteriori log-density function can be derived as:

$$\begin{aligned} L(\Pi|\mathcal{I}', \Theta) &= \log p(\Pi|\mathcal{I}', \Theta) \\ &= \sum_{x_1=1}^{N_1} \sum_{x_2=1}^{N_2} \log \left\{ \sum_{j=1}^J \pi_{x_1, x_2}^j \Phi(I'(x_1, x_2)|\mu_j, \sigma_j) \right\} \\ &\quad + \log p(\Pi) \\ &= \sum_{x_1=1}^{N_1} \sum_{x_2=1}^{N_2} \log \left\{ \sum_{j=1}^J \pi_{x_1, x_2}^j \Phi(I'(x_1, x_2)|\mu_j, \sigma_j) \right\} \\ &\quad - \log \mathbf{Z} - \frac{\mathbf{U}(\Pi)}{T} \end{aligned} \quad (5.11)$$

The expectation maximization (EM) algorithm is usually used to estimate the parameters of a GMM distribution. However, the inclusion of prior distribution to GMM via an MRF introduces additional complexity and the M-step of the EM algorithm cannot be directly applied to estimate the model parameters from the observations. Various approximations have been introduced in order to tackle this problem. Recently, Nguyen et. al. [5] introduced a novel way of incorporating spatial correlations in MRF model which allows a close form solution in the Maximization

step. They introduced a factor G_{x_1, x_2}^j as follows:

$$G_{x_1, x_2}^j = \exp \left\{ \frac{\beta}{2|\mathcal{N}_{x_1, x_2}|} \sum_{i \in \{\mathcal{N}_{x_1, x_2}, (x_1, x_2)\}} (z_i^j + \pi_i^j) \right\} \quad (5.12)$$

where \mathcal{N}_{x_1, x_2} is the neighborhood of the pixel (x_1, x_2) . The factor G_{x_1, x_2}^j is proportional to the product of both posterior probabilities and prior distributions of the neighboring pixels and causes smoothing of prior probabilities due to the averaging process. Nguyen et. al. [5] proposed a new Gibbs energy function based on G_{x_1, x_2}^j as follows:

$$\mathbf{U}^{(t+1)}(\Pi|\Theta) = - \sum_{x_1=1}^{N_1} \sum_{x_2=1}^{N_2} \sum_{j=1}^J G_{x_1, x_2}^{j, (t)} \log \pi_{x_1, x_2}^{j, (t+1)} \quad (5.13)$$

The factor G_{x_1, x_2}^j is dependent only on the value of the priors and posteriors at the previous step in the EM algorithm. This allows a simpler, closed-form solution for the update of mixing proportions π_{x_1, x_2}^j as will be shown in section 5.4.1.4 under detailed description of the EM algorithm.

5.4.1.3 Fusion of Gabor Features and Texture Orientation Field

Under the GMM-MRF model, each pixel shares a global set of parameters of Gaussian distributions denoted by Θ . However, each pixel has a different set of mixing proportions, π_{x_1, x_2}^j , instead of a global set. In this section, we describe our modifications to the GMM-MRF model of Nguyen et. al. [5] for the fusion of texture orientation field and Gabor features. The texture orientation field, $\theta(x_1, x_2)$, is considered a priori field affecting the prior probabilities of mixing proportions. Let $\Omega = \{\theta_{x_1, x_2}; x_1 = 1, \dots, N_1; x_2 = 1, \dots, N_2\}$ denote the set of orientation angles of

all pixels. Then, incorporating Ω , equation (5.9) can be re-written as:

$$\begin{aligned} p(\Pi, \Theta | \mathcal{I}', \Omega) &\propto p(\mathcal{I}' | \Pi, \Theta, \Omega) \times p(\Pi | \Omega) \\ &\propto p(\mathcal{I}' | \Pi, \Theta) \times p(\Pi | \Omega) \end{aligned} \quad (5.14)$$

where:

$$p(\Pi | \Omega) = \frac{1}{\mathbf{Z}} \exp\left(-\frac{\mathbf{U}(\Pi | \Omega)}{T}\right) \quad (5.15)$$

$$\mathbf{U}(\Pi | \Omega) = - \sum_{x_1=1}^{N_1} \sum_{x_2=1}^{N_2} \sum_{j=1}^J G_{x_1, x_2}^j(\Omega) \log \pi_{x_1, x_2}^j \quad (5.16)$$

We introduce the factor $G_{x_1, x_2}^j(\Omega)$ as a function of the orientation field Ω and is given as follows:

$$G_{x_1, x_2}^j(\Omega) = \exp \left\{ \sum_{i \in \{\mathcal{N}_{x_1, x_2}, (x_1, x_2)\}} h(j, \theta_i) (z_i^j + \pi_i^j) \right\} \quad (5.17)$$

The factor $h(j, \theta_i)$ controls the mixing proportions of a pixel based on the orientation field angles in its neighborhood. In the binary case, where $j = 0$ denotes the distribution representing background skin and $j = 1$ denotes the distribution of wrinkled skin, the factor is defined as:

$$h(j, \theta_i) = \begin{cases} 1 & \text{for } j = 0 \\ \beta \cos \theta_i & \text{for } j = 1 \end{cases} \quad (5.18)$$

The parameter β has a value greater than 1. So whenever a pixel and its surrounding neighbors have orientation angles of closer to zero, the value of the factor $h(j, \theta_i)$ increases. This increases the mixing proportion π_i^j and that pixel's probability of being included in the wrinkled area.

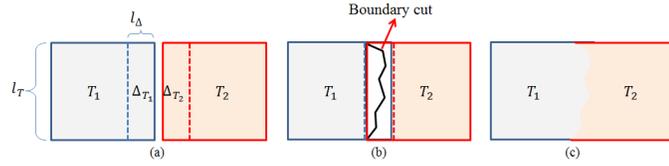


Figure 5.11: Seamless stitching of two patches. (a) Patches with side portions to be overlapped. (b) Minimum square distance boundary cut along the overlapped portions. (c) Resulting stitching of the two patches.

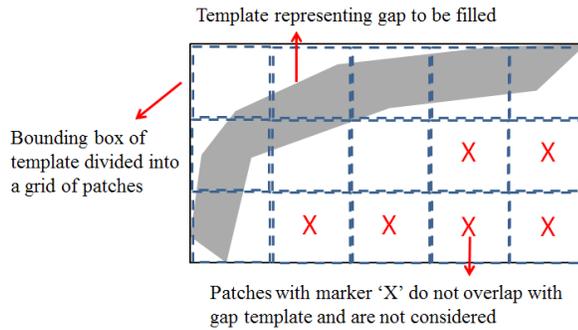


Figure 5.12: The constrained texture synthesis algorithm divides an irregular shaped gap into a regular grid of patches. Each patch is then marked to be painted if it overlaps any pixels of the gap.

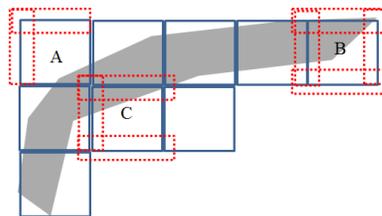


Figure 5.13: The sides of a patch used for seamless stitching are based on two factors; the patch's location in the grid and if the surrounding patches are marked to be painted or not.

5.4.1.4 Expectation Maximization (EM) Algorithm

We use the EM algorithm to find mixing proportions and to maximize the posterior distribution. At iteration t , the E-step of the EM algorithm requires the formulation of the following optimization function [51]:

$$\begin{aligned} \mathbf{Q}_{MAP}(\Theta, \Pi | \Theta^{(t)}, \Pi^{(t)}) \\ = \mathbf{E} [\log p(\mathcal{I}' | \Theta, \Pi) | \mathcal{I}', \Theta^{(t)}, \Pi^{(t)}] + \log p(\Pi) \end{aligned} \quad (5.19)$$

$$\begin{aligned} \mathbf{Q}_{MAP}(\Theta, \Pi | \Theta^{(t)}, \Pi^{(t)}) \\ = \sum_{x_1=1}^{N_1} \sum_{x_2=1}^{N_2} \sum_{j=1}^J z_{x_1, x_2}^{j, (t)} (\log \pi_{x_1, x_2}^{j, (t+1)} \\ + \log \Phi(I'(x_1, x_2) | \Theta^{(t)}, \Pi^{(t)})) \\ - \log \mathbf{Z} + \frac{1}{\mathbf{T}} \sum_{x_1=1}^{N_1} \sum_{x_2=1}^{N_2} \sum_{j=1}^J G_{x_1, x_2}^{j, (t)}(\Omega) \log \pi_{x_1, x_2}^{j, (t+1)} \end{aligned} \quad (5.20)$$

where $z_{x_1, x_2}^{j, (t)}$ is given as follows:

$$z_{x_1, x_2}^{j, (t)} = \frac{\pi_{x_1, x_2}^{j, (t)} \Phi(I'(x_1, x_2) | \mu_j^{(t)}, \sigma_j^{(t)})}{\sum_{k=1}^K \pi_{x_1, x_2}^{k, (t)} \Phi(I'(x_1, x_2) | \mu_k^{(t)}, \sigma_k^{(t)})} \quad (5.21)$$

The constants \mathbf{Z} and \mathbf{T} do not effect the optimization function are are set equal to

1. The modified optimization function is then given as:

$$\begin{aligned}
\mathbf{Q}_{MAP}(\Theta, \Pi | \Theta^{(t)}, \Pi^{(t)}) &= \sum_{x_1=1}^{N_1} \sum_{x_2=1}^{N_2} \sum_{j=1}^J z_{x_1, x_2}^{j, (t)} (\log \pi_{x_1, x_2}^{j, (t+1)} \\
&+ \log \Phi(I'(x_1, x_2) | \Theta^{(t)}, \Pi^{(t)})) \\
&+ \sum_{x_1=1}^{N_1} \sum_{x_2=1}^{N_2} \sum_{j=1}^J G_{x_1, x_2}^{j, (t)}(\Omega) \log \pi_{x_1, x_2}^{j, (t+1)} \tag{5.22}
\end{aligned}$$

$$G_{x_1, x_2}^{j, (t)}(\Omega) = \exp \left\{ \sum_{i \in \{N_{x_1, x_2}, (x_1, x_2)\}} (z_i^{j, (t)} + h(j, \theta_i) \pi_i^{j, (t)}) \right\} \tag{5.23}$$

In the M-step of EM algorithm, the function $\mathbf{Q}_{MAP}(\Theta, \Pi | \Theta^{(t)}, \Pi^{(t)})$ is maximized to obtain the updated values of the parameters $\{\Theta, \Pi\}$. After setting the derivative of $\mathbf{Q}_{MAP}(\Theta, \Pi | \Theta^{(t)}, \Pi^{(t)})$ to zero, we obtain the following expressions for updating parameters Θ :

$$\mu^{j, (t+1)} = \frac{\sum_{x_1=1}^{N_1} \sum_{x_2=1}^{N_2} z_{x_1, x_2}^{k, (t)} I'(x_1, x_2)}{\sum_{x_1=1}^{N_1} \sum_{x_2=1}^{N_2} z_{x_1, x_2}^{j, (t)}} \tag{5.24}$$

$$\sigma^{j, (t+1)} = \left[\frac{\sum_{x_1=1}^{N_1} \sum_{x_2=1}^{N_2} z_{x_1, x_2}^{k, (t)} [I'(x_1, x_2) - \mu^{j, (t+1)}]^2}{\sum_{x_1=1}^{N_1} \sum_{x_2=1}^{N_2} z_{x_1, x_2}^{j, (t)}} \right]^{\frac{1}{2}} \tag{5.25}$$

The values of parameters Π are obtained using the method of Lagrange multiplier and are given as follows:

$$\pi_{x_1, x_2}^{j, (t+1)} = \frac{z_{x_1, x_2}^{j, (t)} + G(\Omega)_{x_1, x_2}^{j, (t)}}{\sum_{k=1}^K [z_{x_1, x_2}^{k, (t)} + G(\Omega)_{x_1, x_2}^{k, (t)}]} \tag{5.26}$$

For the EM algorithm, values of the parameters Π, Θ are initialized by using the MatlabTM functions to train a GMM with two distributions. Then, following the

algorithm given in [5], we proceed with the following steps to complete the automatic detection of wrinkled region.

1. Evaluate the values z_{x_1, x_2}^j according to (5.21) using the current parameter values.
2. Update the factor $G_{x_1, x_2}^j(\Omega)$ using (5.23).
3. Update the means μ^j using (5.24) and variances σ^j using (5.25).
4. Update the prior distributions π_{x_1, x_2}^j using (5.26).
5. Evaluate the log-likelihood in (5.22) and check the convergence of either the log-likelihood function or the parameter values. If the convergence criterion is not satisfied, then go to step 2.

At the end of the EM algorithm, every pixel is assigned the label j with the larger posterior probability value z_{x_1, x_2}^j .

5.4.2 Automatic Removal of Facial Wrinkles

The detected wrinkled regions are inpainted by surrounding skin texture using texture synthesis. Texture synthesis techniques can be categorized as parametric or exemplar-based. In parametric methods, the parameters of a generative texture model are learned from a sample texture. A texture image can then be synthesized by sampling the learned model. The exemplar-based methods focus on sampling patches from a sample texture and then stitching them seamlessly, incorporating neighborhood details, to synthesize larger texture images. The exemplar-

based methods have become popular in recent years to synthesize 2D texture images, [42, 46–48].

Our method is based on the exemplar-based texture synthesis method proposed by Efros and Freeman [48]. In their work, Efros and Freeman introduced a novel method called *image quilting* for seamless stitching of small patches of the exemplar texture. We use their method to stitch skin patches together to fill the gaps left by removal of wrinkled regions. Filling of gaps in images using texture synthesis is also called as ‘constrained texture synthesis’ for the reason that the boundaries between the original texture and the synthesized texture have to be invisible. In section 5.4.2.1, we briefly describe the method of Efros and Freeman for seamless stitching of two patches. Then, in section 5.4.2.2, we present our algorithm of constrained texture synthesis to fill gaps. Finally, in section 5.4.2.3, we present our method of selecting the skin source texture to be inpainted the gaps from.

5.4.2.1 Patch based Image Quilting

Let S_T denote a small source texture sample from which the bigger texture image has to be synthesized. Let T_1 and T_2 denote two square patches to be stitched together and l_T denote the length of a side of a patch. Let Δ_{T_1} and Δ_{T_2} denote portions of the patches T_1 and T_2 from any side as is shown in Fig. 5.11. The portions are set to be of width $l_\Delta < l_T$. The problem of seamless stitching of the two patches then boils down to finding a ragged boundary in the overlapped portions Δ_{T_1} and Δ_{T_2} such that a minimal discontinuity in texture flow is caused across the

boundary. This method is illustrated in Fig. 5.11.

Given T_1 (the first T_1 can be selected randomly from S_T), the first step is to select a patch, T_2 , of the same size from S_T . This is done by finding one or more patches from S_T having cumulative square difference between the overlapping regions, Δ_{T_1} and Δ_{T_2} , within a given tolerance. The patch T_2 is then selected randomly from such patches. The next step is to stitch the two patches seamlessly by founding the best boundary cut.

Let E be a matrix of size $l_T \times l_\Delta$ representing the square difference between Δ_{T_1} and Δ_{T_2} as follows:

$$E^{i,j} = (\Delta_{T_1}^{i,j} - \Delta_{T_2}^{i,j})^2 \quad (5.27)$$

where the superscripts i, j designate the i th row and j th column of the matrix. Then the ragged boundary is found as the cut through E which minimizes the cumulative square difference along the boundary and is found through the following steps.

1. Let E_c be a matrix of size $l_T \times l_\Delta$ denoting the cumulative square difference for a boundary cut.
2. Initialize the first row of E_c to be equal to the first row of E i.e. $E_c^{i,\cdot} = E^{i,\cdot}$.
3. Calculate the rest of the elements of E_c as follows:

$$E_c^{i,j} = E^{i,j} + \min\{E_c^{i-1,j}, E_c^{i-1,j-1}, E_c^{i,j-1}\} \quad (5.28)$$

4. Find the minimum value in the last row of E_c . This is the last element of the cut with the minimum square distance. The best boundary can be found by

tracing back vertically the cut from this element to the first row of E_c .

5.4.2.2 Constrained Texture Synthesis to Fill Image Gaps

The removal of wrinkled skin results in several gaps of irregular shapes as can be seen in Fig. 5.15(b). Filling such gaps requires modifications to the texture synthesis method presented in the last section which was originally used to synthesize rectangular texture samples. Fig. 5.16 presents our algorithm to fill irregular gaps using constrained texture synthesis. The algorithm performs two steps for every gap detected by the GMM-MRF algorithm. The first step consists of finding the bounding box for the current gap and fitting it with a rectangular grid of square patches. Then, each patch in the grid is visited to determine if it overlaps with any pixel(s) in the gap. Fig. 5.12 illustrates this step. The patches which do not overlap with the gap are marked as ‘X’ and are not considered in the second step. In the second step, the patches containing image gap pixels are replaced with the patches of the source skin texture. Each patch is stitched from two (top and left) or more sides depending on its location in the grid. Fig. 5.13 illustrates this step. Patches are visited in a raster scanning manner. Patch ‘A’ is stitched from top and left side with the rest of the skin image. Patch ‘B’ is stitched from all four sides because there is no patch to be visited on its right or bottom side. Patch ‘C’ is stitched from bottom side as there is no patch below it to be visited.

5.4.2.3 Selection of Texture Source Template

Usually a texture sample image, S_T , is provided to a texture synthesis algorithm. However, in our case, to minimize the user interaction, a skin texture source template has to be determined automatically. Since facial skin texture varies greatly, for every patch to be inpainted, we use the skin texture nearest to that patch as a source template. A texture source template is selected so that (a) it is of size $1.3 \times T$, where we choose 1.3 because it is small enough for faster computation time and big enough to provide a compatible texture source, (b) it does not overlap with any of the wrinkle gaps and (c) it is nearest to the current patch to be inpainted. Once an S_T is selected this way, a suitable patch, T , can then be found within this texture source template.

5.4.2.4 Compensation for Skin Tone Variations

This is a post-processing step and is applied specifically to the areas under eyes. This is due to the fact that the skin under eyes is not only wrinkled, but, frequently, has discolorations due to sagging, under-eye bags or dark circles as well. Although image quilting provides seamless stitching of two patches, its main focus is the overlapping areas δ_T of the two patches. In under-eye regions, the interior of such patches may still present a significant skin tone difference. Therefore, a simple stitching method cannot provide the needed adjustment to the overall tone of the inpainted patch. This is illustrated in Fig. 5.17 where boundaries of several patches stitched together are obvious due to the skin tone variation. We use the Poisson

image painting tool [38] to compensate for this tone variation. In Algorithm I, once the patch has been stitched, in case of eyes, the Poisson image editing is used as a post-processing step to compensate for the tone variation. The difference in the inpainting results with and without Poisson image editing is shown in Fig. 5.17.

5.5 Experiments

5.5.1 Experimental Setup

Experiments were conducted on two sets of images downloaded from the Internet. The first set consists of images of public figures, e.g. celebrities and politicians, and was used to remove facial wrinkles. The second set consists of portions of facial images of other people and was used to remove other skin imperfections. We had to crop portions of interest (e.g. cheeks, forehead) from these facial images to hide identities of subjects. All of the images were taken in uncontrolled settings and varied in identity and age of the subjects, illumination and expression. Facial images of public figures consisted of frontal pose whereas other images consisted of side poses as well. All of the images were of high resolution, larger than 1024 pixels x 768 pixels, showing detailed texture of skin. Here we make an interesting observation that facial wrinkles were more prominent in images of male celebrities than those of female celebrities. Hence, most of our experiments were conducted on images of male celebrities. Each image was cropped to have face only and then resized so that the larger dimension was equal to 1100 pixels.

One of the objectives of this work is to minimize user interaction. The only user interaction was to provide polygonal areas of skin to be investigated. Fig. 5.18 shows two polygonal areas provided by a user to be inpainted for a facial image. The only constraint for these patches was to contain skin and no other facial parts. Since GMM-MRF detection is based on Gabor features which depend on image gradients, the inclusion of facial features other than skin, e.g. hair, eyes, would result in erroneous detection results.

Regarding parameters of the Gabor filter bank, the spatial frequency of the sinusoid, f , was set to be $\frac{1}{5}$ cycles per pixels. The variances of the Gaussian envelopes were chosen to be $\sigma_{x_2} = L\sigma_{x_1}$ where σ_{x_1} was set to 0.5 and L was set to 25. The value of K was set to be 180. In patch based image quilting the margin width, l_Δ , was set to be equal to $\frac{1}{3}l_T$ whereas the patch size, l_T , was set to be 11 pixels.

5.5.2 Results and Discussion

5.5.2.1 Removal of Wrinkles

Figures 5.20, 5.31, 5.24, 5.21, 5.30, 5.29, and 5.25 show the results of removal of wrinkles in facial images of public figures. In all the figures, the original images are shown in parts (a), the gaps resulted by GMM-MRF algorithm are shown in parts (b) and inpainted images are shown in parts (c). The user provided areas for these images contained wrinkles due to aging on forehead and around eyes and mouth. Here we would like to mention briefly wrinkles caused by facial expressions. Facial expressions not only highlight aging wrinkles but also cause extra folds on

facial skin. These folds of expressive wrinkles are usually wider than those of aging wrinkles. When inpainting images for the removal of wrinkles, we make a distinction between expressive and aging wrinkles. The removal of expressive wrinkles may result in unnatural looking facial shapes. Fig. 5.24 and Fig. 5.25 show two examples where wrinkles due to smiling were not removed.

Overall, our experiments demonstrate that most of the wrinkles and skin imperfections are detected and inpainted. However, a few with less contrast with the surrounding skin are not detected. For example, small parts of wrinkles on the upper right sides of subjects' foreheads in Fig. 5.21 and Fig. 5.25 remain undetected. Similarly, very few wrinkles around the corners of the eyes in Fig. 5.31 and Fig. 5.24 remain undetected as well. Regarding areas under eyes, the algorithm removes most of the wrinkles while maintaining the skin tone variation due to dark circles. This effect was desirable as the goal was to remove wrinkles without other beautification of the skin. The under eye dark circles remain unaltered due to the skin tone compensation step described in section 5.4.2.4.

5.5.2.2 Removal of Moles/Dark Spots/Scars

We also applied our algorithm on other types of skin imperfections, e.g. moles, dark/brown spots, acne, wound scars and freckles. These imperfections also appear as a disruption in the surrounding texture. Since these images did not contain full facial images to hide the identities of the subjects, these were resized to have the maximum dimension equal to 500 pixels instead of 1100 pixels in case of full facial

images. Figures 5.22 and 5.23 show results of removal of these imperfections for subjects of different skin color. Overall, the detection is better in images where skin imperfections have sharper color contrast with the surrounding skin. The imperfections are removed irrespective of the cause, color, size and shape. We observe that acne or wound scars, moles and darker brown spots are detected and removed. Lighter and smaller brown spots were the most difficult kind of imperfection to be detected. Figures 5.24 and 5.25 show two subjects with darker skin and several moles/dark spots on cheeks which are removed along with wrinkles by the inpainting algorithm.

Fig. 5.27 shows two examples where some of the brown spots/freckles remained undetected. In part (a), the freckles are over segmented due to their less color contrast with the surrounding skin. In part (b), a large brown spot is present along with few smaller spots. A significant color variation is present within the larger spot which creates inhomogeneous orientation texture field within the spot and not equal to zero throughout the spot. Fig. 5.27 (b2) shows the inhomogeneous orientation field. Hence, the areas within the brown spot with nonzero, inhomogeneous orientation field are given lower probability of being an imperfection according to the factor $h(j, \theta_i)$ in equation (5.17) and are left out.

5.5.3 Challenges

5.5.3.1 Effect of Age/Sagging Skin

Advanced aging skin shows signs of overall sagging of skin as well in addition to rough texture and wrinkles. Facial images where subjects had sagging of skin along with wrinkles posed specific challenges to inpainting. In advanced ages, wrinkles are present very closely, and the area of the non-wrinkled skin is limited. This results in the selection of the same patch as a source skin texture for several patches to be inpainted and results in detectable repetitive patterns in the inpainted skin. Then, the overall sagging of skin results in changes of facial muscles and shape which are visible to the eye but not to the GMM-MRF algorithm and hence cannot be detected and removed. Fig. 5.26 shows images of two subjects where most of the wrinkles have been removed but the signs of sagging of skin around mouth, jawline and under eyes remain.

5.5.3.2 Effect of Illumination

The images used in our experiments had a variety of illumination settings. Overall, small illumination variations did not effect the detection results. However, in cases of significant illumination variations e.g. due to pose or bright spots on skin, the intensity changes due to wrinkles were masked by those due to illumination. Two such examples are presented in Fig. 5.28. In Fig. 5.28(1a,1b), wrinkles are masked by the dark areas due to low illumination and in Fig. 5.28(2a,2b), wrinkles are

masked by the bright spots. However, in both figures, the wrinkles not masked by illumination are correctly detected by the GMM-MRF algorithm.

5.5.3.3 Artifacts due to the Repetition of Patches

In few cases, the filling of gaps by the seamless stitching of patches resulted in artifacts that could be detected by a user. For example in Fig. 5.20(c), Fig. 5.25(c) and Fig. 5.26(2c), some of the inpainted areas can be detected due to the slight change in the textures of the inpainted and the surrounding skin. This happens when the same non-wrinkled, texture source image is the nearest to more than two patches to be inpainted. The algorithm then uses the same source skin texture to paint more than one patch. This results in the absence of randomness found naturally in the slowly varying skin texture. The absence of the randomness and the consistency of the painted texture looks like an artifact to the eye. It was observed that a smaller patch size l_{Δ} provided better, natural looking inpainted skin texture as it is easier to find smaller non-wrinkled, source skin texture patches to inpaint from.

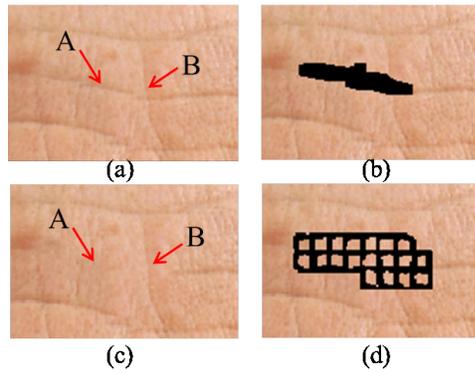


Figure 5.14: Removing a wrinkle with texture synthesis. (a) Original skin image. (b) Template of the gap to be filled. (c) Skin Image after texture synthesis. Comparing with original skin image in (a), note that wrinkle ‘A’ has been removed since it was included in the gap whereas part of wrinkle ‘B’ is re-painted. (d) Stitching of skin patches to fill the gap.

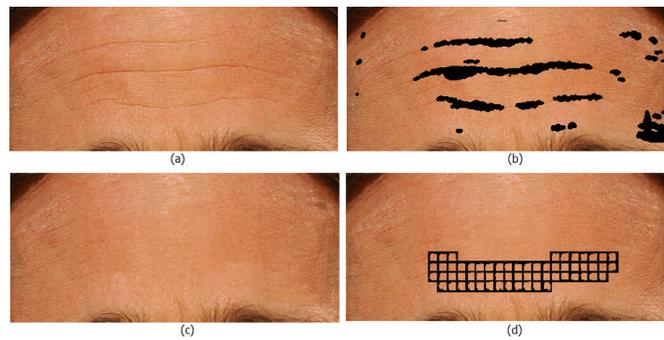


Figure 5.15: A close up view of wrinkle removal. (a) Original Image. (b) Several wrinkled areas detected by GMM-MRF. (c) Inpainted image after removal of wrinkles. (d) Patches from regular grid fitted on one gap which were actually inpainted.

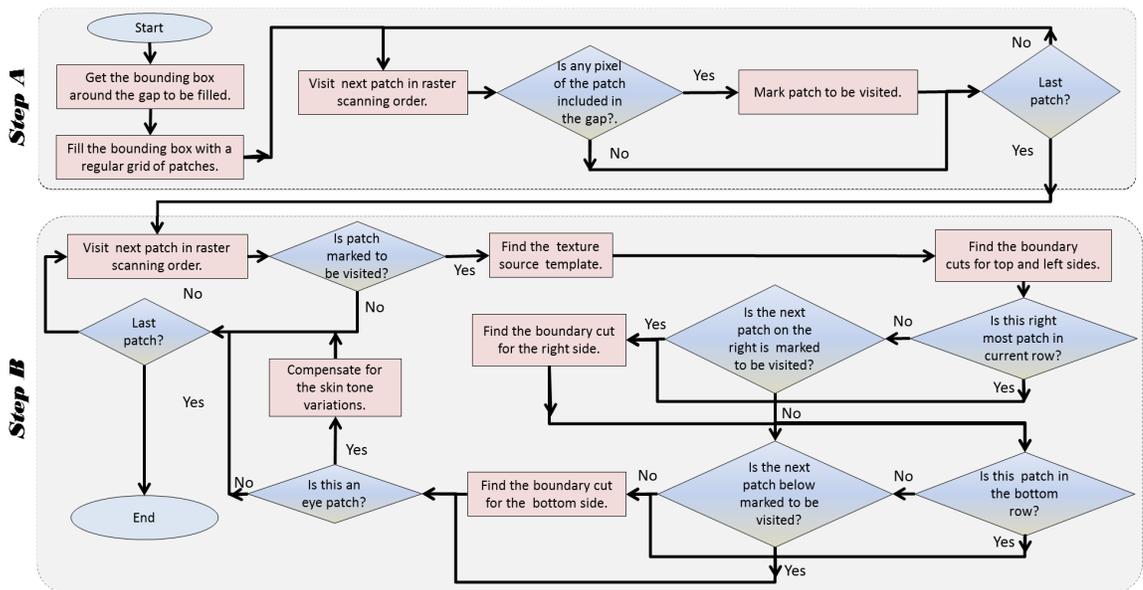


Figure 5.16: Flow chart for the algorithm to fill gaps using constrained texture synthesis.



Figure 5.17: Poisson compensation for color variation under eyes. (a) Inpainted skin without Poisson compensation. (b) Inpainted skin with Poisson compensation.

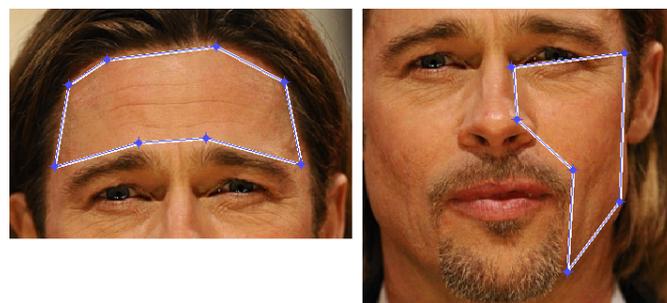


Figure 5.18: Level of user interaction. Users have to select polygonal skin patches to remove wrinkles/skin imperfections from.

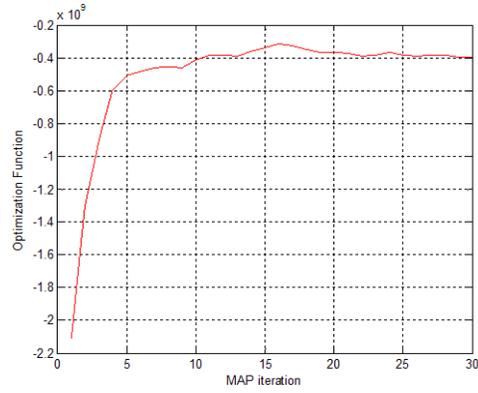


Figure 5.19: A typical graph for the objective function vs. iterations of GMM-MRF function. There is little variation in the function for most images after 10 iterations.

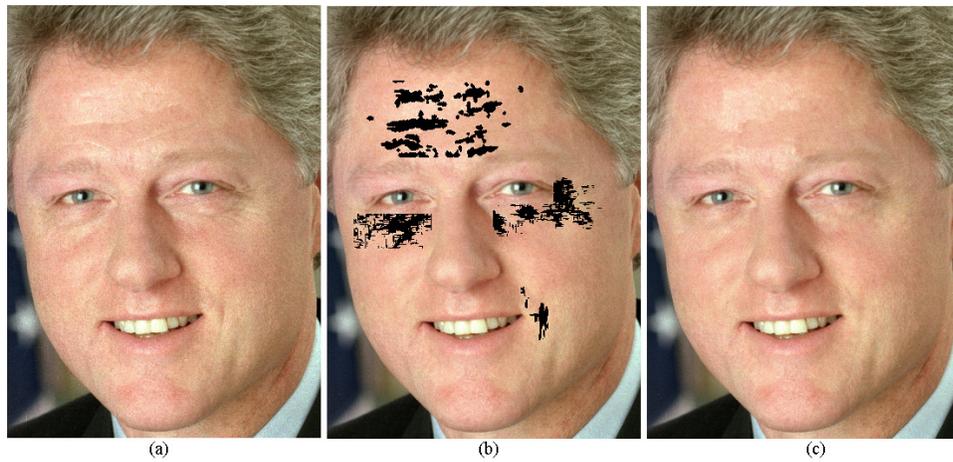


Figure 5.20: Results of wrinkle detection and removal for a subject. (a) Original image. (b) Detected wrinkled areas. (c) Image after wrinkle removal.

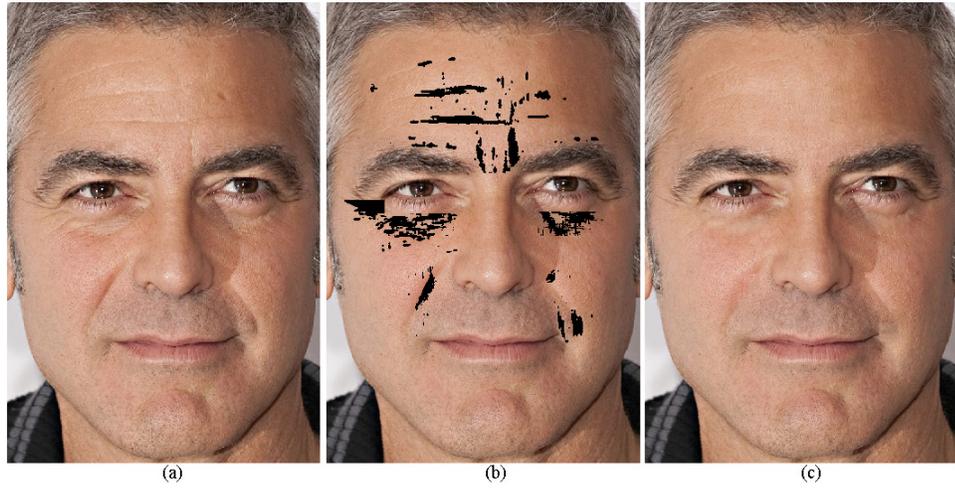


Figure 5.21: Results of wrinkle detection and removal for a subject. (a) Original image. (b) Detected wrinkled areas. (c) Image after wrinkle removal.

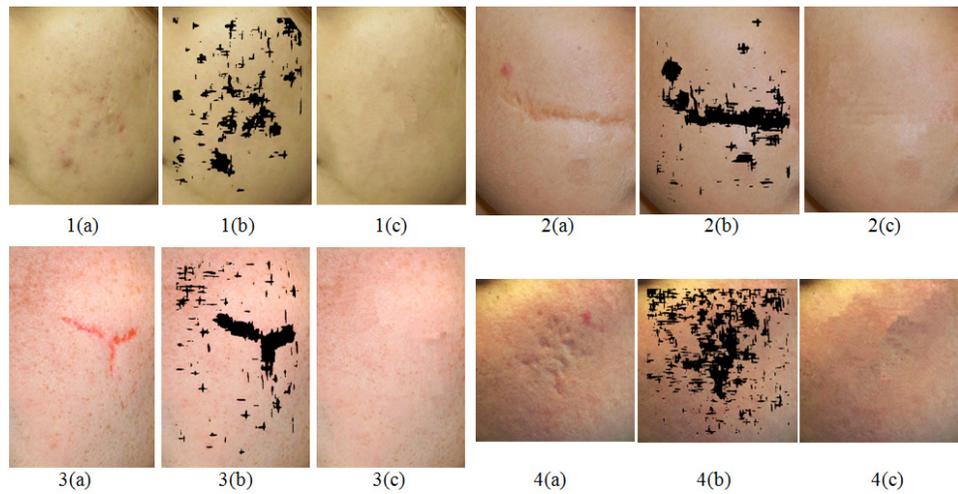


Figure 5.22: Results of detection and removal of skin imperfections including wound scars, acne, brown spots and moles. (a) Original images. (b) Detected imperfections. (c) Images after inpainting.

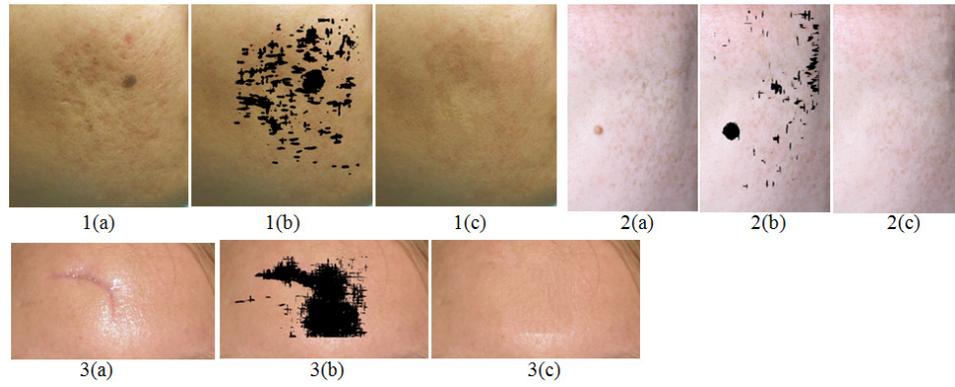


Figure 5.23: Results of detection and removal of skin imperfections including wound scars, acne, brown spots and moles. (a) Original images. (b) Detected imperfections. (c) Images after inpainting.

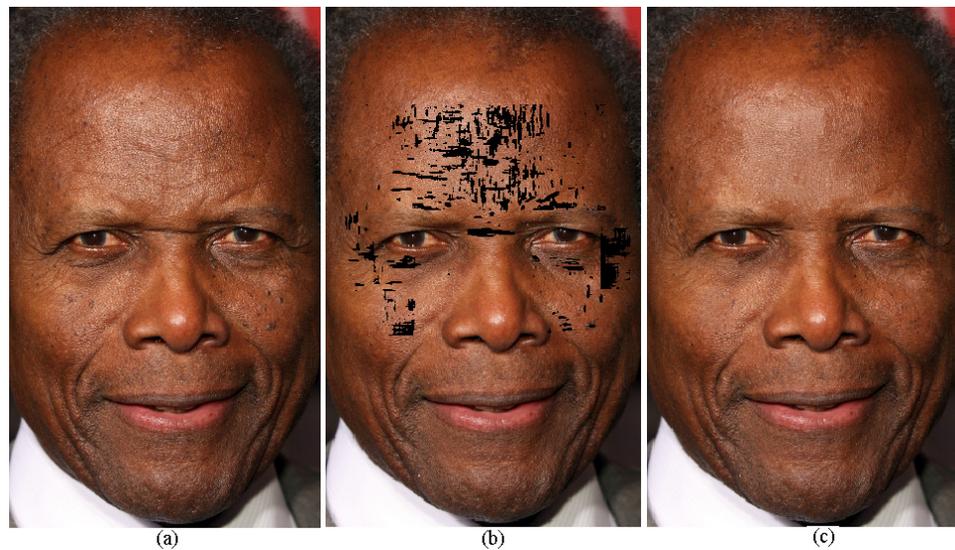


Figure 5.24: Results of wrinkle and dark spot detection and removal for a subject with darker skin. (a) Original image. (b) Detected wrinkled areas and dark spots. (c) Image after inpainting, most of the dark spots are removed as well.

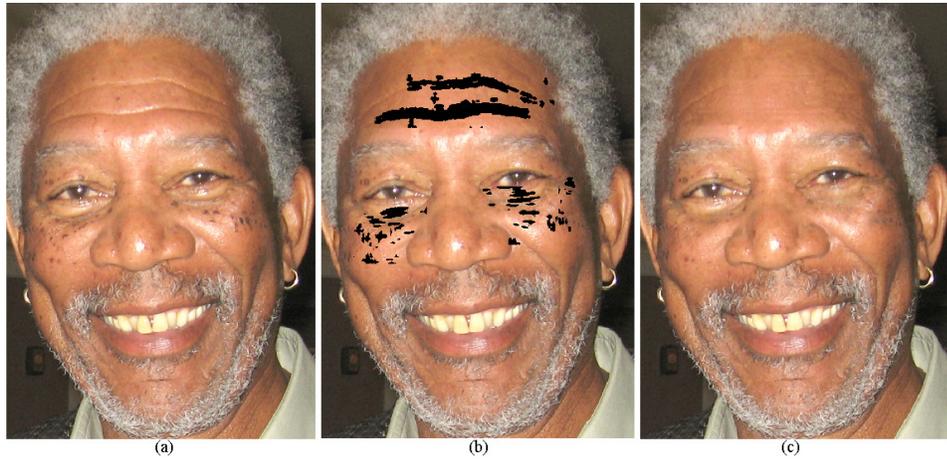


Figure 5.25: Results of wrinkle and dark spot detection and removal for a subject with darker skin. (a) Original image. (b) Detected wrinkled areas and dark spots. (c) Image after inpainting, most of the dark spots are removed as well.

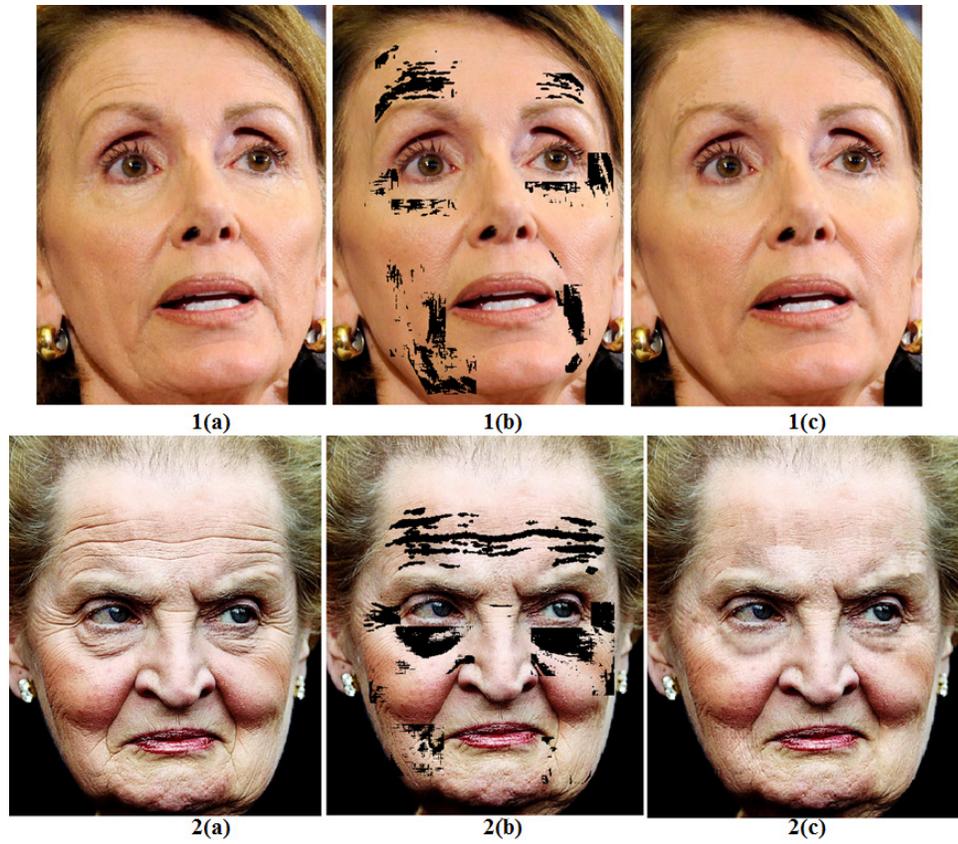


Figure 5.26: Results of wrinkle detection and removal for a subject with sagging skin. (a) Original image. (b) Detected wrinkled areas . (c) Image after inpainting, most wrinkles are removed but skin sagging is visible.

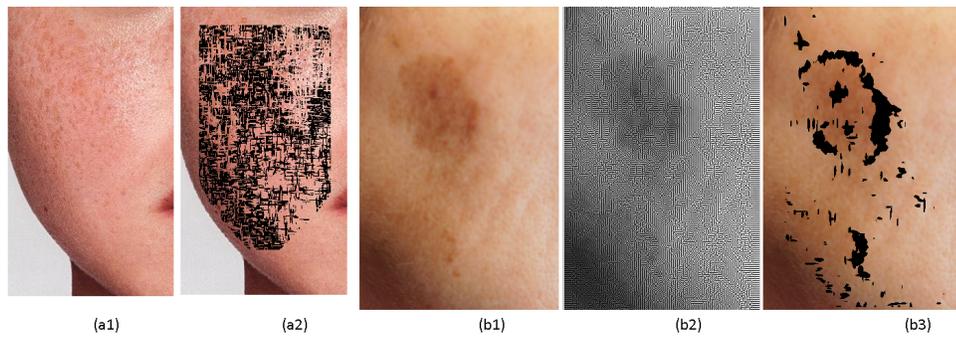


Figure 5.27: Challenges in the detection of brown spots. (a) Brown spots are over segmented due to lower contrast with surrounding skin. (b) Brown spot is not detected due to the inhomogeneity of texture orientation field within the spot as shown in (b2).



Figure 5.28: Effect of illumination on detection of wrinkles. (1a,1b) Wrinkles are masked by bright spots. (2a,2b) Wrinkles are masked by shadows.

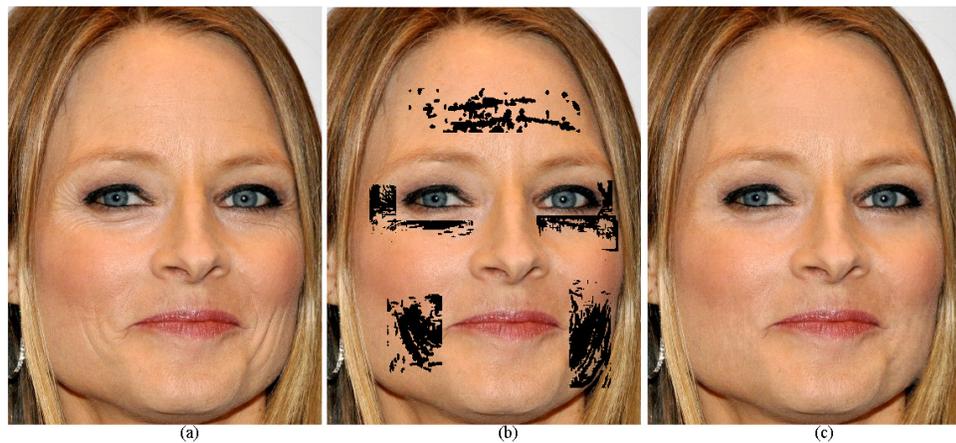


Figure 5.29: Results of wrinkle detection and removal for a subject. (a) Original image. (b) Detected wrinkled areas. (c) Image after wrinkle removal.

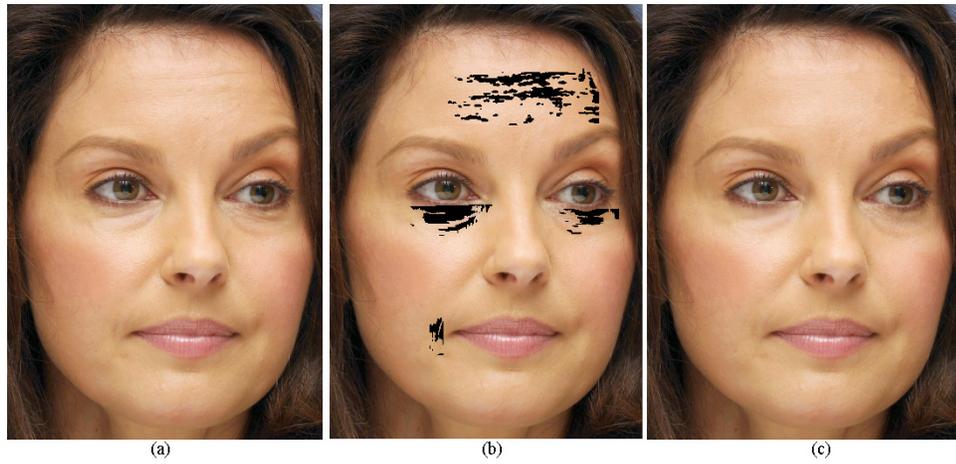


Figure 5.30: Results of wrinkle detection and removal for a subject. (a) Original image. (b) Detected wrinkled areas. (c) Image after wrinkle removal.

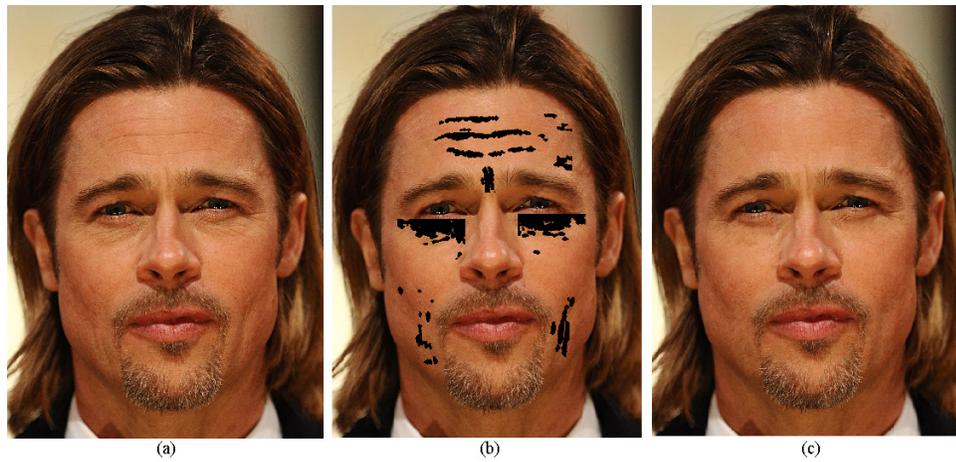


Figure 5.31: Results of wrinkle detection and removal for a subject. (a) Original image. (b) Detected wrinkled areas. (c) Image after wrinkle removal.

Chapter 6: Conclusion

6.1 Summary

In this dissertation, we examined the image properties of wrinkles i.e. intensity gradients and geometric properties and used them for several applications including low-level image processing for detection, soft biometrics and digital inpainting for removal of wrinkles in images. We presented our work on four problems briefly summarized below:

We presented results of detection/localization of wrinkles in images using Marked Point Process (MPP). Wrinkles were modeled as sequences of line segments in a Bayesian framework. Wrinkles were then localized by sampling the posterior probability using a Reversible Jump Markov Chain Monte Carlo (RJMCMC) algorithm. We also presented an evaluation algorithm to quantitatively evaluate the detection and false alarm rate of our algorithm and conducted experiments with images taken in uncontrolled settings. The model can allow the incorporation of wrinkles, otherwise difficult features to capture, explicitly in future applications.

Next we presented a computationally efficient deterministic algorithm based on Ga-

bor filters and image morphology to improve wrinkle localization results. We used image features based on Gabor filter bank to highlight subtle curvilinear discontinuities in skin texture caused by wrinkles and image morphology to incorporate geometric constraints of wrinkle curves. We conducted experiments on a dataset of medium resolution images downloaded from the Internet. Experiments showed that the proposed algorithm was faster as well as provided better visualization results.

We presented our work on the investigation of forehead wrinkles as curve patterns for their discriminative power as a soft biometrics. We experimented with different metrics based on the shape and spatial proximity of the curves. The recognition rate achieved by using only the wrinkle patterns and no other facial features was promising. We also presented the recognition rates on automatically detected wrinkles. This work presented a rudimentary analysis of the wrinkle curves and the structure of a wrinkle pattern with the information of relative positions/orientations of the curves was not included. Also a simple algorithm for finding curve correspondences from two patterns was presented.

We addressed some of the limitations of current facial retouching applications. We presented an algorithm incorporating Gabor features and texture orientation field of facial skin in the framework of GMM and MRF representations to detect wrinkles and other skin imperfections. Then, we presented an algorithm based on exemplar-based texture synthesis to automatically inpaint the irregular gaps left by the removal of skin wrinkles/imperfections. Experiments

on images downloaded from the Internet show the effectiveness of our algorithms. With minimum user interaction, the algorithms were able to detect and remove most of the wrinkles/imperfections.

We propose the following directions for extension of this work in future.

6.2 Future Work

1. Regarding MPP modeling, a more elaborate data likelihood term, based on *shearlet or wavelet transform coefficients*, can be used to highlight the wrinkle intensity gradients more accurately. The elements of the shearlet representation are basically a collection of well-localized waveforms, at various locations, scales and orientations, and with highly anisotropic shapes. This makes the shearlet representation particularly well adapted at representing the edges and intensity gradients at various scales and orientations. In addition, a more elaborate prior model incorporating more spatial interaction rules can be proposed. Finally, a technique to the estimates of MPP parameters can be investigated.
2. Regarding facial biometrics, a global curves pattern matching technique can be proposed by (a) incorporating structural information i.e. positions/orientations of curves relative to each other and (b) matching curves in feature/parameter space (e.g. B-splines) instead of image domain. In addition, the curves pattern matching can be assessed to do recognition in the presence of expressions or pose.
3. Regarding removal of facial wrinkles using digital inpainting, the work can be

extended to address the sagging of skin in more advanced stages of aging as well as to improve inpainting to address the artifacts caused by repetition.

4. Given a facial image of a particular person, the goal of aging/age progression/age synthesis/age simulation is to construct facial image of the same person at a later age. However, most of the age simulation methods model facial shape variations in adults and elderly as an average, composite or prototype face for a specific age group or a specific range of difference in age instead of learning of a person's specific aging pattern. In future, spatial point process modeling can be enhanced to spatio-temporal point process modeling to analyze the characteristic facial aging pattern of an individual's facial wrinkles to develop a facial aging model specific to that individual.

Bibliography

- [1] N. Batool' and R. Chellappa, "A markov point process model for wrinkles in human faces," in *ICIP*, 2012, pp. 1809–1812.
- [2] N. Batool and R. Chellappa, "Modeling and detection of wrinkles in aging human faces using marked point processes," in *ECCV Workshops (2)*, 2012, pp. 178–188.
- [3] [Online]. Available: <http://www.photoshopsupport.com/tutorials/jf/retouching-tutorial/remove-wrinkles-healing-brush.html>
- [4] [Online]. Available: <http://app.lk/visage-lab>
- [5] T. Nguyen and Q. Wu, "Fast and robust spatially constrained Gaussian mixture model for image segmentation," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 23, no. 4, 2013.
- [6] Y. Fu, G. Guo, and T. Huang, "Age synthesis and estimation via faces: A survey," *PAMI*, vol. 32, no. 11, pp. 1955–1976, nov. 2010.
- [7] K. Luu, T. D. Bui, C. Y. Suen, and K. Ricanek, "Combined local and holistic facial features for age-determination," in *ICCARV*, 2010, pp. 900–904.
- [8] J. Suo, F. Min, S. Zhu, S. Shan, and X. Chen, "A multi-resolution dynamic model for face aging simulation," in *CVPR*, 2007, pp. 1–8.
- [9] Y. H. Kwon and N. D. V. Lobo, "Age classification from facial images," in *CVPR*, 1999.
- [10] G. O. Cula, P. R. Bargo, and N. Kollias, "Assessing facial wrinkles: automatic detection and quantification," pp. 71 610J–71 610J–6, 2009.
- [11] O. G. Cula, K. J. Dana, F. P. Murphy, and B. K. Rao, "Skin texture modeling," *IJCV*, vol. 62, pp. 97–119, April 2005.

- [12] M. N. M. Lieshout, *Markov Point Processes and their Applications*. Imperial College Press/World Scientific Publishing: London/Singapore, 2000.
- [13] R. Stoica, X. Descombes, and J. Zerubia, “A Gibbs point process for road extraction from remotely sensed images,” *IJCV*, vol. 57, pp. 121–136, May 2004.
- [14] P. Green, “Reversible jump MCMC computation and Bayesian model determination,” *Biometrika*, vol. 82, pp. 711–732, 1995.
- [15] C. J. Geyer and J. Moller, “Simulation procedures and likelihood inference for spatial point processes,” *Scandinavian Journal of Statistics*, vol. 21, no. 4, pp. 359–373, 1994.
- [16] Available, “The FG-net aging database,” <http://www.fgnet.rsunit.com/>.
- [17] S. Chambon and J.-M. Moliard, “Automatic road pavement assessment with image processing: Review and comparison,” *International Journal of Geophysics*, vol. 2011, p. 20, 2011.
- [18] M. M. Fraz, P. Remagnino, A. Hoppe, B. Uyyanonvara, A. R. Rudnicka, C. G. Owen, and S. A. Barman, “Blood vessel segmentation methodologies in retinal images - a survey,” *Comput. Methods Prog. Biomed.*, vol. 108, no. 1, pp. 407–433, Oct. 2012.
- [19] F. J. Ayres and R. M. Rangayyan, “Design and performance analysis of oriented feature detectors,” *Journal of Electronic Imaging*, vol. 16, no. 2, pp. 023 007–023 007–12, 2007. [Online]. Available: + <http://dx.doi.org/10.1117/1.2728751>
- [20] F. Zana and J.-C. Klein, “Segmentation of vessel-like patterns using mathematical morphology and curvature evaluation,” *Image Processing, IEEE Transactions on*, vol. 10, no. 7, pp. 1010–1019, 2001.
- [21] A. Landstrom and M. Thurley, “Morphology-based crack detection for steel slabs,” *Selected Topics in Signal Processing, IEEE Journal of*, vol. 6, no. 7, pp. 866–875, 2012.
- [22] B. Lam and H. Yan, “A novel vessel segmentation algorithm for pathological retina images based on the divergence of vector fields,” *Medical Imaging, IEEE Transactions on*, vol. 27, no. 2, 2008.
- [23] T. Lelore and F. Bouchara, “Fair: A fast algorithm for document image restoration,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 8, pp. 2039–2048, 2013.
- [24] A. Jain and U. Park, “Facial marks: Soft biometric for face recognition,” in *Image Processing (ICIP), 2009 16th IEEE International Conference on*, nov. 2009, pp. 37–40.

- [25] P. Miller, J. Lyle, S. Pundlik, and D. Woodard, "Performance evaluation of local appearance based periocular recognition," in *Biometrics: Theory Applications and Systems (BTAS), 2010 Fourth IEEE International Conference on*, sept. 2010, pp. 1–6.
- [26] U. Park and A. Jain, "Face matching and retrieval using soft biometrics," *Information Forensics and Security, IEEE Transactions on*, vol. 5, no. 3, pp. 406–415, sept. 2010.
- [27] P. Phillips, P. Flynn, K. Bowyer, R. Bruegge, P. Grother, G. Quinn, and M. Pruitt, "Distinguishing identical twins by face recognition," in *Automatic Face Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, march 2011, pp. 185–192.
- [28] B. Klare, A. A. Paulino, and A. K. Jain, "Analysis of facial features in identical twins," in *Proceedings of the 2011 International Joint Conference on Biometrics*, ser. IJCB '11, 2011, pp. 1–8.
- [29] M. Ouaret, A. Dantcheva, R. Min, L. Daniel, and J. L. Dugelay, "Bioface: a biometric face demonstrator," in *Proceedings of the international conference on Multimedia*, ser. MM '10, 2010, pp. 1613–1616.
- [30] X. Yu and M. Leung, "Shape recognition using curve segment Hausdorff distance," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 3, 0-0 2006, pp. 441–444.
- [31] C. Guerra and V. Pascucci, "Line-based object recognition using Hausdorff distance: from range images to molecular secondary structures," *Image Vision Comput.*, vol. 23, no. 4, pp. 405–415, Apr. 2005.
- [32] X. Yi and O. Camps, "Line-based recognition using a multidimensional Hausdorff distance," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 21, no. 9, pp. 901–916, sep 1999.
- [33] M.-P. Dubuisson and A. Jain, "A modified Hausdorff distance for object matching," in *Pattern Recognition, 1994. Vol. 1 - Conference A: Computer Vision and Image Processing., Proceedings of the 12th IAPR International Conference on*, vol. 1, oct 1994, pp. 566–568 vol.1.
- [34] W. J. Rucklidge, "Efficiently locating objects using the Hausdorff distance," *Int. J. Comput. Vision*, vol. 24, no. 3, pp. 251–270, Sep. 1997.
- [35] [Online]. Available: <http://www.smartience.com/>
- [36] [Online]. Available: <http://perfect365.arcsoft.com/win/features.html>
- [37] T. Georgiev, "Image reconstruction invariant to relighting," *EUROGRAPHICS*, 2005.

- [38] A. B. P. Prez, M. Gangnet, “Poisson image editing,” *Proceedings of ACM SIGGRAPH*, pp. 313–318.
- [39] T. K. Shih and R.-C. Chang, “Digital inpainting - survey and multilayer image inpainting algorithms,” *Third International Conference on Information Technology and Applications, ICITA*, vol. 1, pp. 15–24, 2005.
- [40] V. C. A. Bugeau, M. Bertalmo and G. Sapiro, “A comprehensive framework for image inpainting,” *IEEE Trans. Image Process, PAMI*, vol. 19, no. 10, pp. 2634–2645, 2010.
- [41] Z.-N. L. Zinovi Tauber and M. S. Drew, “Review and preview: Disocclusion by inpainting for image-based rendering,” *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, vol. 37, no. 4, pp. 527–540, 2007.
- [42] A. Criminisi, P. Perez, and K. Toyama, “Object removal by exemplar-based inpainting,” in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 2, june 2003, pp. II–721 – II–728 vol.2.
- [43] M. Elad, J.-L. Starck, P. Querre, and D. Donoho, “Simultaneous cartoon and texture image inpainting using morphological component analysis (mca),” *Applied and Computational Harmonic Analysis*, vol. 19, no. 3, pp. 340 – 358, 2005.
- [44] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera, “Filling-in by joint interpolation of vector fields and gray levels,” *Image Processing, IEEE Transactions on*, vol. 10, no. 8, pp. 1200 –1211, aug 2001.
- [45] H. Grossauer, “A combined pde and texture synthesis approach to inpainting,” in *Computer Vision - ECCV 2004*, ser. Lecture Notes in Computer Science, T. Pajdla and J. Matas, Eds. Springer Berlin Heidelberg, 2004, vol. 3022, pp. 214–224.
- [46] L.-Y. Wei and M. Levoy, “Fast texture synthesis using tree-structured vector quantization,” in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH ’00, 2000, pp. 479–488.
- [47] A. Efros and T. Leung, “Texture synthesis by non-parametric sampling,” in *In International Conference on Computer Vision*, 1999, pp. 1033–1038.
- [48] A. A. Efros and W. T. Freeman, “Image quilting for texture synthesis and transfer,” in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH ’01. New York, NY, USA: ACM, 2001, pp. 341–346. [Online]. Available: <http://doi.acm.org/10.1145/383259.383296>

- [49] A. Rao and R. Jain, “Computerized flow field analysis: oriented texture fields,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 14, no. 7, pp. 693–709, jul 1992.
- [50] P. Orbanz and J. Buhmann, “SAR images as mixtures of Gaussian mixtures,” in *Image Processing, 2005. ICIIP 2005. IEEE International Conference on*, vol. 2, 2005, pp. II–209–12.
- [51] S. Sanjay-Gopal and T. Hebert, “Bayesian pixel classification using spatially variant finite mixtures and the generalized em algorithm,” *Image Processing, IEEE Transactions on*, vol. 7, no. 7, pp. 1014–1028, 1998.