FEATURE EXTRACTION AND IMAGE ANALYSIS WITH THE APPLICATIONS TO PRINT QUALITY ASSESSMENT, STREAK DETECTION, AND PEDESTRIAN DETECTION

A Dissertation
Submitted to the Faculty
of
Purdue University
by
Xing Liu

In Partial Fulfillment of the
Requirements for the Degree
of
Doctor of Philosophy

Dec 2018
Purdue University
West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

Dr. Jan P. Allebach, Chair
    School of Electrical and Computer Engineering
Dr. Kar-Ann Toh
    School of Electrical and Electronic Engineering, Yonsei University
Dr. George Chiu
    School of Mechanical Engineering
Dr. Mary comer
    School of Electrical and Computer Engineering
Dr. Amy Reibman
    School of Electrical and Computer Engineering

Approved by:

Dr. V. Balakrishnan
    Head of the Department Graduate Program

ACKNOWLEDGMENTS

First and foremost I would like to express my sincere gratitude to my advisor Professor Allebach. I appreciate his guidance, patience, and the trust he has placed in me since the very first day. He has taught me how a researcher thinks, works, writes, and presents. I am also thankful for his continuous support and understanding of all my decisions throughout my Ph.D. study regarding the research topic, internship, and so on. During the countless meetings, he not only contributed his time, immense knowledge, and insight, but he also provided substantial encouragement, great humor, and concerns to students. The passion he has for his work always impresses me and motivates me in my career path. I also want to thank him for the opportunities to work with Lexmark, Océ, and Hewlett-Packard.

My sincere thanks also go to Professor Toh, who offered me the opportunity to work as a researcher in his lab at Yonsei University. He provided me an excellent example as an insightful researcher with critical and independent thinking. Without his precious support, it would not be possible to conduct my research while I was working in Yonsei University. I also want to thank Dr. Segovia for her help and support during my internship at Océ.

I would like to thank Professor George Chiu for his help on my research, especially on the work with Lexmark. I am grateful to Professor Comer and Professor Reibman for their insightful comments.

I also appreciate Lexmark, Océ, and HP for their funding support during my Ph.D. program.

The members of the EISL group have been more than helpful and supportive. They have been a source of friendship, advice, and help during both my academic and personal time. I also want to thank my friends from Purdue for offering the care, supports, and good memories.

Last but not least, I would like to thank my family for their unconditional love; for my parents who always support and understand me in all my pursuit; for my parents-in-law who have been caring me like their daughter; for my loving, and supportive husband Jason who believes in me in all situations.

TABLE OF CONTENTS

Page

LIST OF TABLES ix

LIST OF FIGURES x

ABSTRACT xiv

1 INTRODUCTION 1

 1.1 Wavelet-Based Figure of Merit for Macro-Uniformity 2

 1.2 Printing Quality Metric for 2.5D Printing 3

 1.3 Streak Detection in Scanned Images 4

 1.4 Pedestrian Detection using Pixel Difference Matrix Projection (DMP) . 5

2 WAVELET-BASED FIGURE OF MERIT FOR MACROUNIFORMITY . . 6

 2.1 Introduction 6

 2.2 Test Page Design 7

 2.3 Human Vision Model (HVM) and L^* Component 8

 2.4 Wavelet-Based Feature Extraction 8

  2.4.1 Wavelet filtering 8

  2.4.2 Frequency band images 10

  2.4.3 Structural approximation 10

  2.4.4 Feature extraction 13

 2.5 Results 15

  2.5.1 Psychophysical experiment 15

  2.5.2 Experiment setup and design 16

  2.5.3 Experiment result 16

3 CHARACTERIZTION OF 2.5D PRINTING 19

 3.1 Introduction 19

 3.2 2.5D Printing 21

Page
3.3 Scanning Methods 22
3.4 Test Targets 23
  3.4.1 Twisted target 23
  3.4.2 Sinusoidal wave target 24
  3.4.3 Ramp target 26
3.5 Quality Metrics 27
  3.5.1 Overall fidelity 28
  3.5.2 Surface finish 30
3.6 Results 33
  3.6.1 Overall fidelity 33
  3.6.2 Surface finish of relief prints 38
4 FEATURE BASED STREAK DETECTION IN SCANNED IMAGES 46
4.1 Introduction 46
4.2 Related Works 46
4.3 Feature Based Streak Detection 47
  4.3.1 Pre-processing 48
  4.3.2 ΔE 50
4.4 Feature Extraction 51
  4.4.1 Average Peak Location Derivative (APLD) 51
  4.4.2 Modified Local Binary Patterns (M-LBP) 56
  4.4.3 Other features 59
4.5 Ground Truth 60
4.6 Evaluation 60
4.7 Experiment 61
  4.7.1 Dataset 61
  4.7.2 Result 62
5 HIDDEN MARKOV MODEL BASED STREAK DETECTION 64
5.1 Introduction 64

Page

5.2 Related Works 64
5.3 Preliminary 65
5.3.1 HMM 65
5.4 Problem Formulation 66
5.5 Parameter Estimation 67
5.6 Experiment 68
6 PEDESTRIAN DETECTION USING PIXEL DIFFERENCE MATRIX PROJECTION 70
6.1 Introduction 70
6.2 Related Works 72
6.3 Preliminary 73
6.3.1 Histogram of Gradient (HOG) 73
6.3.2 Linear least squares regression 75
6.4 Difference Matrix Projection (DMP) 76
6.4.1 Pixel difference map computation 77
6.4.2 Local pixel difference descriptor 80
6.4.3 Cell-based local descriptors 82
6.4.4 Block normalization 83
6.4.5 The connection between our method and the CNN model 84
6.4.6 Training and testing 86
6.4.7 Feature analysis 87
6.5 Experiments 88
6.5.1 Effect of cell size setting 90
6.5.2 Performance comparison 91
6.5.3 Friedman test 96
6.5.4 CPU processing time performance 97
6.5.5 Summary of results and discussion 98
7 CONCLUSION 100

Page

REFERENCES 103

VITA 107

LIST OF TABLES

Table Page

2.1 Summary of the structural approximation methods at each frequency band. 11

2.2 Summary of the features at each frequency band. There are in total eight scalar-valued features. 18

3.1 Specification of six sinusoidal wave target sets printed by the wet-on-dry printer. 24

3.2 Four metrics based on the light reflection calculation. 33

3.3 Mean absolute difference and RMSE for the twisted target. 38

4.1 Summary of the streak detection result. 62

5.1 Summary of the streak detection result. 68

6.1 A brief summary of selected approaches to pedestrian detection. 74

6.2 Summary of datasets, methods, and parameter settings. 93

6.3 A comparison of CPU time in seconds on the INRIA dataset. 98

LIST OF FIGURES

Figure Page

2.1 (a) A test page with a coverage of 30%. (b) A one-level 2-D Haar wavelet transform of the scanned page of the image in (a). The upper left image is the low-resolution approximation, the upper right, lower left, and horizontal detail, the vertical detail, and the diagonal detail, respectively. . . 7

2.2 An overview of predictor generation pipeline. The feature extraction process consists of wavelet decomposition, frequency band image summation, structural approximation, and feature extraction. 9

2.3 Left: low frequency band image; Middle: medium frequency band image; Right: high frequency band image of the example image in Fig. 2.1(a). . . 11

2.4 Square-wave approximation for the low frequency band projection along vertical direction (left) and the projection along horizontal direction (right). 12

2.5 Piece-wise linear spline approximation for the medium projection along vertical direction (left) and the projection along horizontal direction (right). 13

2.6 DFT analysis for the high frequency band projection along vertical direction (left) and the projection along horizontal direction (right). 14

2.7 Average training error and testing error linear regression from a 5-fold cross-validation. 17

3.1 Height map of twisted target, which is to be printed by the wet-on-dry printer. 24

3.2 Sinusoidal wave target page designed to measure the MTF of the wet-on-dry printer 25

3.3 Sinusoidal wave target page designed to measure the MTF of the wet-on-wet printer 26

3.4 Ramp target designed to characterize the wet-on-dry printer. 27

3.5 (a) Height modulation of the input signal. (b) Height modulation of the output signal. 29

3.6 (a) Surface slices on the ramp target are taken as shown in the dashed line along the direction indicated by the arrow. (b) Definitions of the four metrics: left angle, right angle, peak height, and width. 31

Figure Page

3.7 (a) Surface of interest for the ramp target. The height of the surface is in measurement units. (b) Surface roughness profile in measurement units. . . 32

3.8 MTF for part of sinusoidal wave target set I printed by the wet-on-dry relief printer. (a) MTF as a function of frequency for 10 fixed amplitudes. (b) MTF as a function of amplitude for 5 fixed frequencies. 34

3.9 MTF for sinusoidal wave target set II printed by the wet-on-wet relief printer. (a) MTF as a function of frequency for 3 fixed amplitudes. (b) MTF as a function of amplitude for 3 fixed frequencies. 36

3.10 Difference map for the twisted target printed using the we-on-dry printer. It is calculated by subtracting the scanned height map from the desired height map. 37

3.11 Ramp targets used for calculation of angular accuracy. This is an image of the targets that have been printed using the wet-on-dry printer. The left and right side are defined as noted. 39

3.12 Measurement for the ramp target shown in Fig. 3.11 printed with the wet-on-dry printer. (a) angles, (b) peak height, and (c) width. 41

3.13 Residuals for the ramp target shown in Fig. 3.11 printed with the wet-on-dry printer: (a) RMS value of the residuals, (b) Maximum value of the residuals. 42

3.14 Surface normal maps for the twisted target: (a)From the input height map. (b)From the scanned height map of the twisted target printed by the wet-on-dry printer. 43

3.15 Surface light reflection comparison. (a), (b), and (c) are with different lighting conditions and viewing angles. 44

3.16 Histograms of the four metrics: root mean square error, contrast, density, density contrast. 45

4.1 An example of image patch that contains a streak. From the zoomed-in image on the right, it can be seen that the streak is about two pixels in width, and has a very faint appearance. 47

4.2 A detection window of 13 pixels in width. The image is scanned along the horizontal direction with the detection window. Within each detection window, the streak is detected on a row-basis. The intensity of a row in the detection window is defined as a row profile, based on which the feature are extracted. 48

4.3 A brief overview of the streak detection framework. 49

Figure Page

4.4 Left: a patch of the image that contains a thin streak in the center; Right: the output in the unit of ΔE after applying Gaussian filtering and color space conversion. 51

4.5 Overview of the feature extraction process. 52

4.6 Left: a patch that contains a streak; Middle: pre-processed output; Right: pixel intensity at the 100th row. 53

4.7 Row profile of a streak. The length of the row profile equal to that of the detection window. 53

4.8 Averaged streak extracted from the groundtruth. 54

4.9 Matched filtering output. 54

4.10 Peak locations from the matched filtering output. Left: A patch that contains a streak. Three detection windows are marked in yellow, green, and brown colors, respectively. Right: For each of the detection windows, the peak locations at each row is marked using the corresponding color. The brown detection window contains a streak. Hence the peak locations are consistent. 56

4.11 Peak locations from the matched filtering output. Left: A patch that contains a streak. Three detection windows are marked in red, purple, and yellow colors, respectively. Right: For each of the detection windows, the peak locations at each row is marked using the corresponding color. The yellow detection window contains a streak, hence the peak locations are consistent. At the location where there is an edge that disturbs the streak, the peak location also changes. 57

4.12 Basic LBP operator. The LBP binary pattern is computed in a neighborhood of 3 × 3 pixels. 58

4.13 The order of pattern in the modified LBP operator 58

4.14 GUI of the ground truth labeling tool. 61

4.15 A preview of the annotated streak is provided by clicking on ‘Show annotated streak’. In case it looks wrong, the annotation can be discarded and redone. 62

4.16 Examples of the scanned pages from the dataset. 63

5.1 $\{X_m\}$ represent the states and $\{Y_m\}$ represent the observations (features). The arrows explains the dependency relationships. 66

5.2 Comparison of the detection results using SVM and HMM. It can be seen that HMM rejects the detections with too short length. 69

Figure Page

6.1 Left: A test sample containing pedestrian. Middle: HOG descriptor. Right: Concatenated HOG features of a block containing 2 × 2 cells. . . . 75

6.2 An overview of the proposed DMP feature extraction process. 77

6.3 Pixel difference pattern involving pixel groups within a neighborhood of 5 × 5 pixels. 79

6.4 Pixel difference patterns at different scales. The pixel differences are taken between the cyan cell and the red cell. The blank cells are not involved in the calculation. Only the magnitudes of the differences will be used to construct the features. 81

6.5 Pipeline of the DMP and the HOG feature extraction algorithm. 84

6.6 Understanding DMP from a different point of view. Upper: the DMP algorithm; Lower: one-layer CNN model 85

6.7 DMP-based pedestrian classification process. 86

6.8 An illustrative example of DMP feature vector describing different shapes. The HOG features are provided for comparison. (a) DMP and HOG feature vectors for a 8 × 8 pixels cell with vertical edges. (b) DMP and HOG feature vectors for a 8 × 8 pixels cell with horizontal edges. (c) DMP and HOG feature vectors for a 8 × 8 pixels cell with diagonal edges. 88

6.9 An illustrative example of noise tolerance. (a) DMP and HOG feature vectors for a 8 × 8 pixels cell containing vertical edge (upper) and the same cell with artificial noise added (lower). (b) DMP and HOG feature vectors for a 8 × 8 pixels cell containing vertical edges with noise (upper) and the same cell with noise removed (lower). 89

6.10 Evaluation of different cell sizes. (a) INRIA dataset. (b) Daimler-CB dataset.92

6.11 Performance comparison on the Daimler-CB dataset. 94

6.12 The performances of selected detectors on the INRIA dataset. 95

6.13 The performances of selected detectors on the NICTA dataset. (a) Pedestrian sample size of 20 × 8 pixels. (b) Pedestrian sample size of 20 × 16 pixels. 96

6.14 Friedman test analyzing the critical difference in performances. The x-axis shows the rank, and the blue bar link detectors for which there is no sufficient evidence to declare a statistically significant difference in their performances. 97

ABSTRACT

Liu, Xing PhD, Purdue University, Dec 2018. Feature Extraction and Image Analysis with the Applications to Print Quality Assessment, Streak Detection, and Pedestrian Detection. Major Professor: Jan P. Allebach.

Feature extraction is the main driving force behind the advancement of the image processing techniques in fields such as image quality assessment, object detection, and object recognition. In this work, we perform a comprehensive and in-depth study on feature extraction for the following applications: image macro-uniformity assessment, 2.5D printing quality assessment, streak defect detection, and pedestrian detection. Firstly, a set of multi-scale wavelet-based features is proposed, and a quality predictor is trained to predict the perceived macro-uniformity. Secondly, the 2.5D printing quality is characterized by a set of merits that focus on the surface structure.Thirdly, a set of features is proposed to describe the streaks, based on which two detectors are developed: the first one uses Support Vector Machine (SVM) to train a binary classifier to detect the streak; the second one adopts Hidden Markov Model (HMM) to incorporates the row dependency information within a single streak. Finally, a novel set of pixel-difference features is proposed to develop a computationally efficient feature extraction method for pedestrian detection.

1. INTRODUCTION

Image processing and computer vision techniques allow the machine to analyze and interpret images more efficiently and accurately than human. Their applications can be found in tasks such as object detection, image segmentation, visual surveillance, image quality analysis, and so on. Two of the areas that benefited from image processing techniques are the printing and scanning. One example is the halftone printing. The halftone printing technology enables a printer to produce multi-tonal prints with a limited number of ink colors. It is utilized by all products ranging from desktop printers and multi-function printing devices. The improvement of halftone imaging algorithm over the years has primarily improved the image quality and affected the users of printers all over the world. Another example is the printing quality analysis. The need for high-quality printing images arises from the daily usage of printers. However, due to the device mechanism, the printed image quality may suffer from banding, non-uniformity, streak, jitter, and motte. The analysis of print quality is crucial to the evaluation and diagnosis of printing devices and the guidance for improvements. As new printing technologies such as 2.5D and 3D printing are brought into the industry, the transitional printing quality analysis is not suitable to be used directly. Thus, new print quality standards are needed and have attracted lots of attention. Other than quality assessment, defect detection is also a primary area for printing and scanning image processing. Defect detection aims to locate the particular type of defects in the printed or scanned image and to guide image healing.

Although the study in printing image processing has undergone significant progress over the last decades, no one simple solution is suitable for all types of problems and devices. For example, as we mentioned above, there are many types of printing defects, which, unfortunately, are impossible to be measured by a single merit. On the other hand, the same kind of defect may appear differently depending on the background content and color. Measuring banding defect in a monotone image would be a different task than in a document image. Finally, prints with significantly different structure, such as 2.5D image, require particularly designed quality measurement methods. Image analysis also finds its influence in image understanding and pattern recognition. Features that are extracted from the images project the image to space where critical information is presented, and at the same time, the dimensions are reduced. Feature extraction has been a crucial topic which significantly influenced the recognition accuracy in the field of computer vision. A good application example of this field is the pedestrian detection.

In this thesis, we deal with four types of problems from the image processing and the computer vision areas

- Wavelet-Based Figure of Merit for Macrouniformity.
- Printing qualtiy metric for 2.5D printing.
- Streak detection in scanned images.
- Pedestrian detection.

The details of each problem are introduced below.

1.1 Wavelet-Based Figure of Merit for Macro-Uniformity

Wavelet transform is a powerful tool that can be applied for image processing and analysis. They provide a multi-scale decomposition of an original image into average terms and detail terms that capture the characteristics of the image at different scales. In this project, we develop a figure of merit for macro-uniformity that is based on wavelets. We use the Haar basis to decompose the image of the scanned page into eleven levels. Starting from the lowest frequency level, we group the eleven levels into three non-overlapping separate frequency bands, each containing three levels. Each frequency band image consists of the superposition of the detail images within that band. We next compute 1-D horizontal and vertical projections for each frequency band image. For each frequency band image projection, we develop a structural approximation that summarizes the essential visual characteristics of that projection. For the coarsest band comprising levels 9,10,11, we use a generalized square-wave approximation. For the next coarsest band comprising levels 6,7,8, we use a piecewise linear spline approximation. For the finest bands containing levels 3,4,5, we use a spectral decomposition. For each 1-D approximation signal, we define an appropriate set of scalar-valued features. These features are used to design a predictor based on linear regression, using the data from our image quality ruler experiments with human subjects.

1.2 Printing Quality Metric for 2.5D Printing

2.5D printing technology developed by Océ allows the superposition of several layers of colorant on different types of media which creates a variation of the surface height defined by the input to the printer. Evaluating the reproduction accuracy of distinct surface characteristics is of great importance to the application of the 2.5D printing system. Therefore, it is necessary to develop quality metrics to evaluate the 2.5D process. In this paper, we focus on the evaluation of the shape and surface characteristics of the 2.5D printing. To achieve this goal, we define metrics and develop models that aim to evaluate 2.5D prints in two aspects: overall fidelity and surface finish. To characterize the overall fidelity, three metrics are calculated: Modulation Transfer Function (MTF), difference and root-mean-squared error (RMSE) between the input height map and scanned height map, and print surface angle accuracy. For the surface finish property, we measure the surface roughness, generate surface normal maps and develop a light reflection model that serves as a simulation of the differences between ideal prints and real prints that may be perceived by human observers. Three sets of test targets are designed and printed by the Océ 2.5D printer prototypes for the calculation of the above metrics: (i) twisted target, (ii) sinusoidal wave target, and (iii) ramp target. The results provide quantitative evaluations of the printing quality in the third dimension and demonstrate that the height of 2.5D prints is reproduced accurately to the input design. The factors that affect the printing quality include printing direction, frequency, and amplitude of the input signal, the shape of 2.5D prints. Besides the above factors, two additional aspects influence the viewing experience of 2.5D prints: lighting condition and viewing angle.

1.3 Streak Detection in Scanned Images

We describe a feature-based method for detecting streaks in the scanned image that are caused by dust on the scanner sensor. The proposed approach takes advantage of an "overscan" region that is created by running the scanner before feeding the document to the sensor. Dust on the sensor will cause streaks at the corresponding locations in the overscan region. These streaks can be detected easily as the overscan region has a uniform and light grey background. Two assumptions are made: the dust on the sensor will always cause a streak in the overscan region and will not move after the document is fed to the scanner, and no extra dust will be introduced while the document is being scanned. We can use the streak information in the overscan region as indicators of the streak locations in the scanned document. That is to say, the streak detection is conducted only in narrow vertical windows surroundings the streak locations in the overscan region. Within the cropped window, we extract row-based features including average streak location derivative and modified LBP. Then we treat each row in the window as a node in a Markov chain. The hidden state of each node describes if it contains steak or not. The observations are the calculated features. The state of each row depends on the feature values and the state of the previous row. As a result, we formulated the problem as a Hidden Markov Model and predicts the streak existence by calculating the state of the nodes in the Markov Chain.

1.4 Pedestrian Detection using Pixel Difference Matrix Projection (DMP)

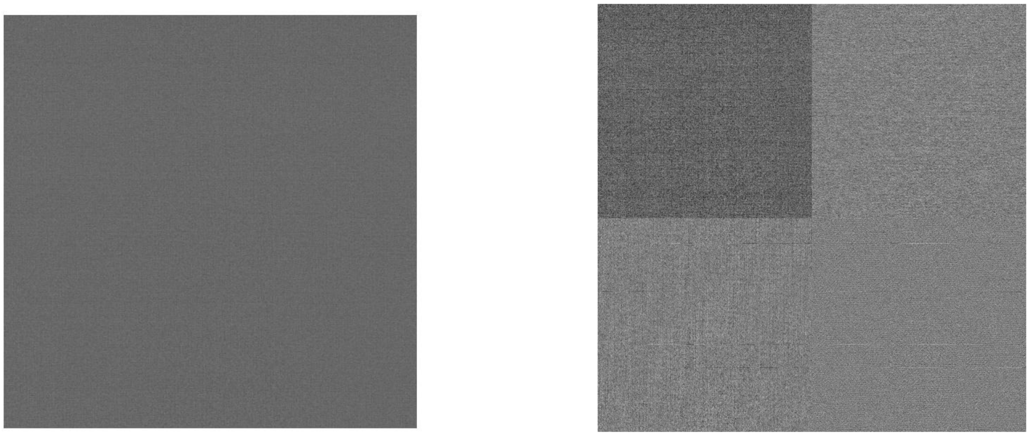
The main challenge in pedestrian detection under video surveillances include having a limited number of pedestrian image samples with low resolution and requires a high detection speed. We address the problem by proposing a set of light-weight features for pedestrian detection when only a small-medium scale of data is available for training. Essentially, a difference matrix projection (DMP) which computes an aggregation of multi-oriented pixel differences using global matrix operations is proposed. The cell-based aggregation reduces the influences of noises in the low-resolution samples, while the global matrix projection avoids the laborious iterative operations. We tested our method on the INRIA, Daimler Chrysler classification (Daimler-CB), and NICTA datasets. The experiments on these benchmark data sets show encouraging results regarding detection performance, particularly for image datasets with low-resolution pedestrians.

2. WAVELET-BASED FIGURE OF MERIT FOR MACROUNIFORMITY

2.1 Introduction

Assessing print quality regarding human perceptions is essential for electrophotographic (EP) printers. The human visual system is often described in the frequency domain, so analyzing the printed pages in the frequency domain has been the main approach in print quality prediction. One of the most widely used multi-scale analysis technique is the wavelet transform. It has been widely applied to noise removal, image compression, and edge detection [1] [2]. Different from the Fourier transform which is based on sinusoids, the wavelet transform uses scaling and wavelet functions as the basis. It provides a multi-scale analysis that allows a decomposition of an image into average terms and detail terms at different resolution levels while maintaining spatial structure [1]. By taking advantage of the multi-scale decomposition, the images can be filtered into different frequency scales, separating the defects in the image at the predefined frequency ranges.

The macro-uniformity in the monotone printed image suffers from the banding defects at different scales in a non-periodical manner. To develop a figure of merit for macro-uniformity, we take advantage of the wavelet analysis for the multi-scale decomposition it provides. Before performing a wavelet analysis, we first apply a human vision model (HVM) and calculate the L^* component. Then we use the Haar wavelet transform to decompose the image into 11 levels. Each level contains one approximation sub-image and three detail sub-images along the horizontal, vertical, and diagonal directions. The eleven levels are then grouped into three non-overlapping frequency bands. Within each frequency band, a single frequency band image is constructed as the superposition of the detail images within that band. Next, a



(a) (b)

Fig. 2.1. (a) A test page with a coverage of 30%. (b) A one-level 2-D Haar wavelet transform of the scanned page of the image in (a). The upper left image is the low-resolution approximation, the upper right, lower left, and horizontal detail, the vertical detail, and the diagonal detail, respectively.

set of horizontal and vertical projections are obtained from each frequency band image, followed by a structural approximation that summarizes the essential visual characteristics of the projections. A set of scalar-valued features is developed for each approximation signal. These features are used to train a predictor based on linear regression.

2.2 Test Page Design

Seven sets of samples are designed as the test pages. Each set is printed by a different printer at 600 dots per inch (dpi) and consists of five samples, ranging from 10% to 90% coverage in a step of 20%. An example test page is shown in Fig. 2.1(a).

2.3 Human Vision Model (HVM) and L^* Component

To analyze the image in terms of human perception, a human vision model (HVM) is applied in the 2-D frequency domain. The Mannos-Sakrison contrast sensitivity function CSF is used at a viewing distance of 15.7 inches. After applying the HVM, the images are converted into the CIE $L^*a^*b^*$ space. The L^* component is used throughout our analysis.

2.4 Wavelet-Based Feature Extraction

The feature extraction process consists of wavelet decomposition, frequency band image summation, structural approximation, and feature extraction. Fig 2.2 provides an overview of the pipeline.

2.4.1 Wavelet filtering

A 2-D discrete wavelet for an image $I(x,y)$ of size $M \times N$ is represented as

$$W_\varphi(j_0, m, n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1}, I(x,y)\varphi_{j_0,m,n}(x,y) \tag{2.1}$$

$$W_\psi^i(j, m, n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1}, I(x,y)\psi_{j,m,n}^i(x,y) \tag{2.2}$$

where $\varphi(x,y)$ is a 2-D scaling function and $\psi^i(x,y), i \in H, V, D$ are 2D wavelets, and H, V, D denote the horizontal, vertical, and diagonal directions, respectively. $W_\varphi(j_0, m, n)$ are the coefficients that define an approximation of the image $I(x,y)$ at scale j_0. $W_\psi^i(j, m, n)$ are the coefficients that add all the details along horizontal, vertical, and diagonal directions at scales $j \geq j_0$.

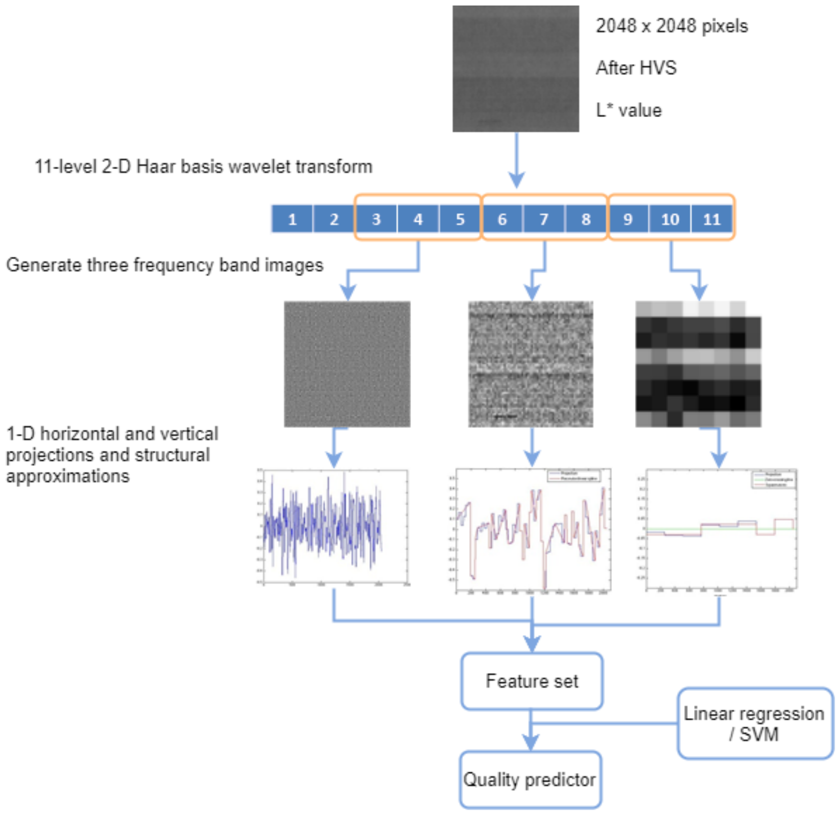

2048 x 2048 pixels
After HVS
L* value
11-level 2-D Haar basis wavelet transform
1
2
3
4
5
6
7
8
9
10
11
Generate three frequency band images
1-D horizontal and vertical projections and structural approximations
Feature set
Linear regression / SVM
Quality predictor

Fig. 2.2. An overview of predictor generation pipeline. The feature extraction process consists of wavelet decomposition, frequency band image summation, structural approximation, and feature extraction.

The Haar wavelet is adopted in our work for its simplicity.The 1-D Haar wavelets and scaling functions are defined as

$$\varphi(x) = (\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}), \tag{2.3}$$

$$\psi(x) = (\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}). \tag{2.4}$$

Fig. 2.1(b) shows an example of a 2-D Haar transform of the example image in Fig. 2.1(a).

2.4.2 Frequency band images

The Haar wavelet transform can be performed at eleven levels for an image of size 2048 × 2048 pixels. The first two levels are dropped out because the highest frequency components are hard to capture by the human being. The remaining nine levels are grouped into three separate frequency bands. The highest frequency band contains levels 3, 4, and 5. The medium frequency band contains levels 6, 7, and 8. The lowest frequency band contains levels 9, 10, and 11. For each frequency band,the horizontal, vertical, and diagonal detail images are summed to form a frequency band image. The frequency band images can be expressed as

$$I^b(x,y) = \frac{1}{\sqrt{MN}} \sum_i \sum_j \sum_m \sum_n W_\psi^i(j,m,n)\psi_{j,m,n}^i(x,y), \tag{2.5}$$

where j denotes the wavelet level, $i \in \{H, V, D\}$, and b denotes the high, medium, and low frequency band. The frequency band images for the example image in Fig. 2.1(a) are shown in Fig. 2.3. It can be seen that the three frequency band images capture the defects at low, medium, and high frequencies. For each frequency band image, we compute the 1-D horizontal and vertical projections. In the ideal case where no defect exists on the monotone printed page, the projections have one single value across the image. However, with the presence of defects, the projections are expected to contain fluctuation at different locations and scales.

2.4.3 Structural approximation

The raw projections from the frequency band images are too noisy to be directly used as features. A structural approximation is applied to extract the most significant characteristic in the projections. It is expected that the projections from difference frequency bands possess different characteristics, hence a particular structural approximation is developed for each frequency band. Table 2.1 gives a summary of the structural approximations.

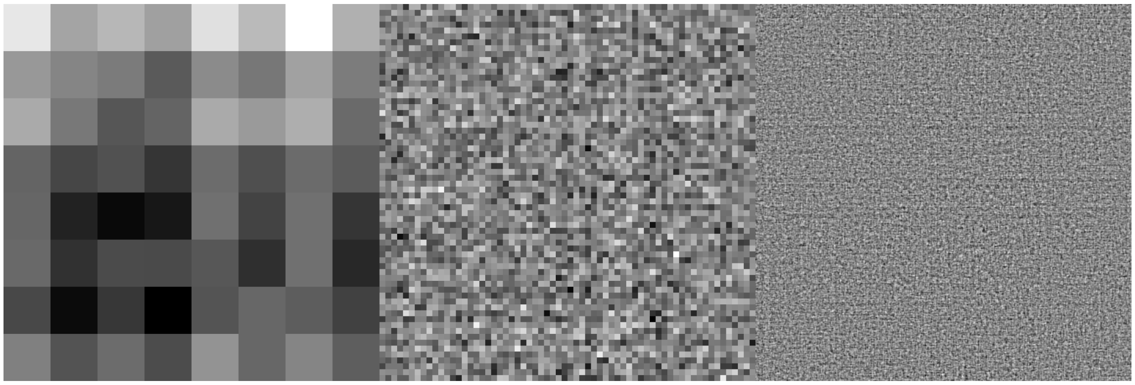


Fig. 2.3. Left: low frequency band image; Middle: medium frequency band image; Right: high frequency band image of the example image in Fig. 2.1(a).

Table 2.1.
Summary of the structural approximation methods at each frequency band.

Frequency range	Wavelet decomposition levels	Structural approximation
Low	9 10 11	Generalized square-wave approximation
Medium	6 7 8	Piecewise linear approximation
High	3 4 5	1D DFT

Low frequency band

The low frequency band image contains large-scale defects by computing. A generalized square-wave approximation is performed on the horizontal and vertical projections as shown in Fig. 2.4. A zero-level threshold is applied to the projection to determine the horizontal edges of the generalized square-wave. Then the height of

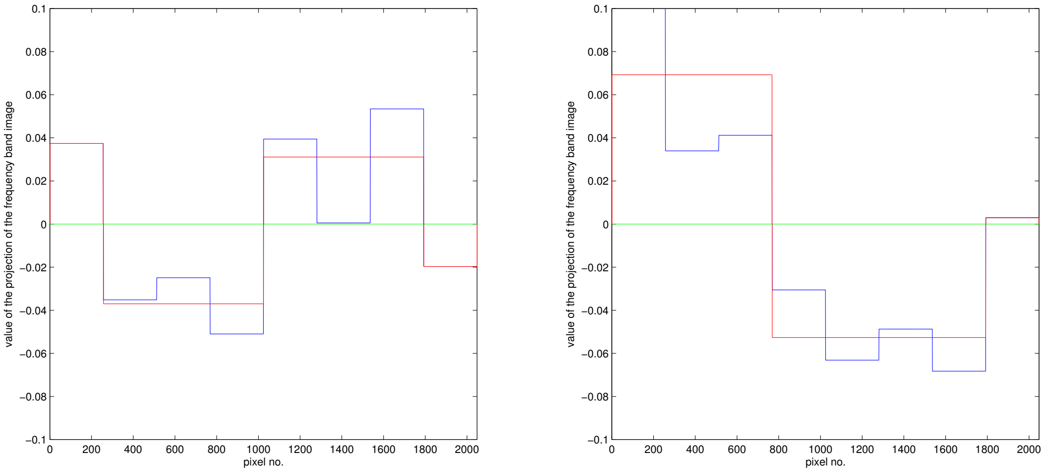


Fig. 2.4. Square-wave approximation for the low frequency band projection along vertical direction (left) and the projection along horizontal direction (right).

each rectangle is the average value of the projection within the extent of the rectangle.

Medium frequency band

For the medium frequency band image, a piecewise linear spline approximation is developed to capture the variation in the projection while ignoring the fine details. We iteratively add knots until the maximum error between the spline and the projection is less than twice the RMS value of the projection. Figure 2.5 shows the result of piecewise linear approximation for the example image in Fig. 2.1(a). The line segments characterize the fluctuations in this frequency range.

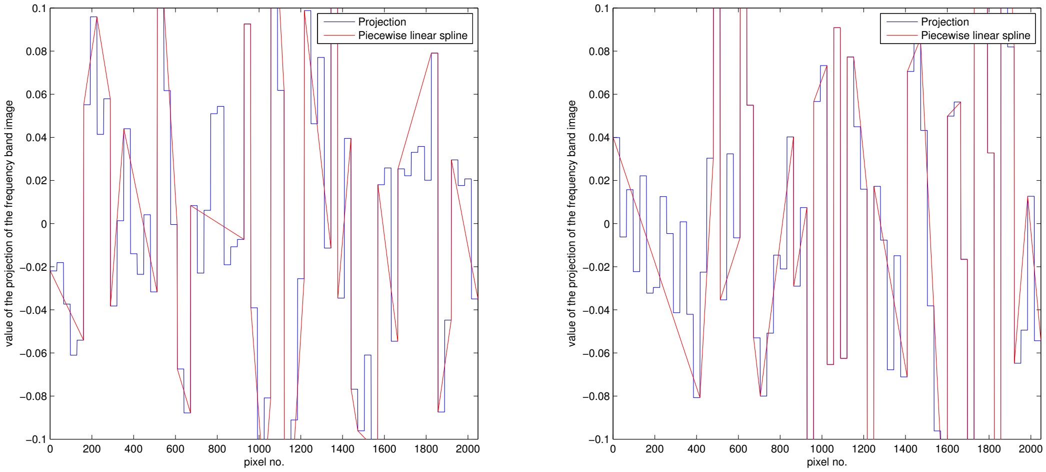


Fig. 2.5. Piece-wise linear spline approximation for the medium projection along vertical direction (left) and the projection along horizontal direction (right).

High frequency band

To capture the high frequency defects, a spectral decomposition is used for the high frequency band image. After obtaining a 1-D DFT of the projection, we define the 'neighborhood of the peak' as the largest interval containing the peak of the DFT such that all values in the interval are above the threshold. Figure 2.6 gives an example of the neighborhood of the peak marked in red.

2.4.4 Feature extraction

To detect and characterize the defect within each frequency range, an appropriate set of scalar-valued features is defined for each structural approximations.

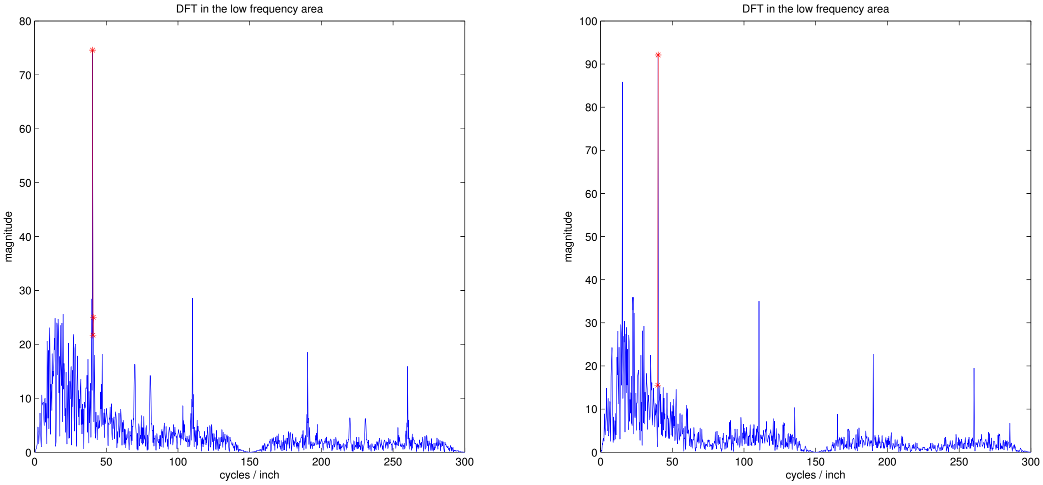


Fig. 2.6. DFT analysis for the high frequency band projection along vertical direction (left) and the projection along horizontal direction (right).

Low frequency band

For the generalized square-wave approximation, the feature set consists of $\overline{|\Delta h|}$, $\max(|\Delta h|)$, $(\overline{|h|} + const)/(\Delta(|h|) + const)$, and $(\overline{|w|} + const)/(\Delta(|w|) + const)$, where h and w denote the height and width of the rectangular pluses in the generalized square wave. The constant term is added in order to avoid instability when the change in $|\Delta h|$ or $|w|$ approaches zero. $\overline{|\Delta h|}$ is the average difference in the heights of two adjacent rectangular pulses and $\max(|\Delta h|)$ is the maximum difference in the heights between two adjacent rectangular pulses. These two features characterizes the variance in the pixel values of the defect image. On the other hand, $(\overline{|h|} + const)/(\Delta(|h|) + const)$ and $(\overline{|w|} + const)/(\Delta(|w|) + const)$ measure the uniformities in the heights and widths of the rectangles, respectively.

Medium frequency band

For piecewise linear spline approximation, we first calculate $\Delta y \times \frac{\Delta y}{\Delta x}$ for each line segment, where Δy is the range of change and $\frac{\Delta y}{\Delta x}$ is the rate of change. With the goal of capturing some sense of the non-uniformities from the structure of the line segments, we compute $\max\left(\Delta y \times \frac{\Delta y}{\Delta x}\right)$ and $\sum\left(\Delta y \times \frac{\Delta y}{\Delta x}\right)$ as the features.

High frequency band

For spectral decomposition, the feature set consists of the peaking factor, defined as the ratio of the total energy in the neighborhood of the peak to the total energy in all other areas, and the energy concentration factor, defined as the peaking factor divided by the width of the neighborhood of the peak.

Table 2.2 summarizes the feature designed for each frequency band.

2.5 Results

The goal of the experiment is to develop a metric that predicts macro-uniformity as judged by human subjects. The metric is based on two parts. The first part is the features set based on the image analysis that we introduced in Sec. 2.3. The second part is the subject evaluation scores set provided by a psychophysical experiment [3].

We conducted the experiment on seven printers, each providing five samples scanned at 600 dpi with nominally uniform coverage ranging from 10% to 90%. Throughout our experiment, all the scanned print samples were de-screened using the human vision model followed by L^* calculation before wavelet processing and feature calculation.

2.5.1 Psychophysical experiment

32 subjects, most of whom are not working in the image quality related area, take part in the psychophysical experiment. Each subject provides a subject score

for each print sample by comparing it to the image quality ruler. The image quality ruler method, proposed by W1.1 working group [4], contains seven image quality ruler (IQR) samples with scores ranging from 3 to 21, with 3 denoting the best print quality with the lowest level of defects. Each unit stands for one just noticeable difference (JND). The scores are averaged over the subjects to give a final evaluation score for each print sample [3].

2.5.2 Experiment setup and design

As discussed in Sec. 2.3, the algorithm generates 16 features for each of the 35 samples. This results in a feature matrix with a size of 35×16. The ground truth is a 35×1 vector with average subject scores as the elements. Due to the limited size of the data set, a 5-fold cross-validation is performed to test the predictor. All the training features are normalized to [0,1] and the testing features are normalized according to the normalization of the training set. The predictor is trained using linear regression.

2.5.3 Experiment result

Fig.2.7 compares the average training error and testing error for each print sample using the linear regression method. The average test error is 1.33 Image Quality Ruler (IQR) units. It can be seen that the predictor provides a good prediction of the macro-uniformity score.

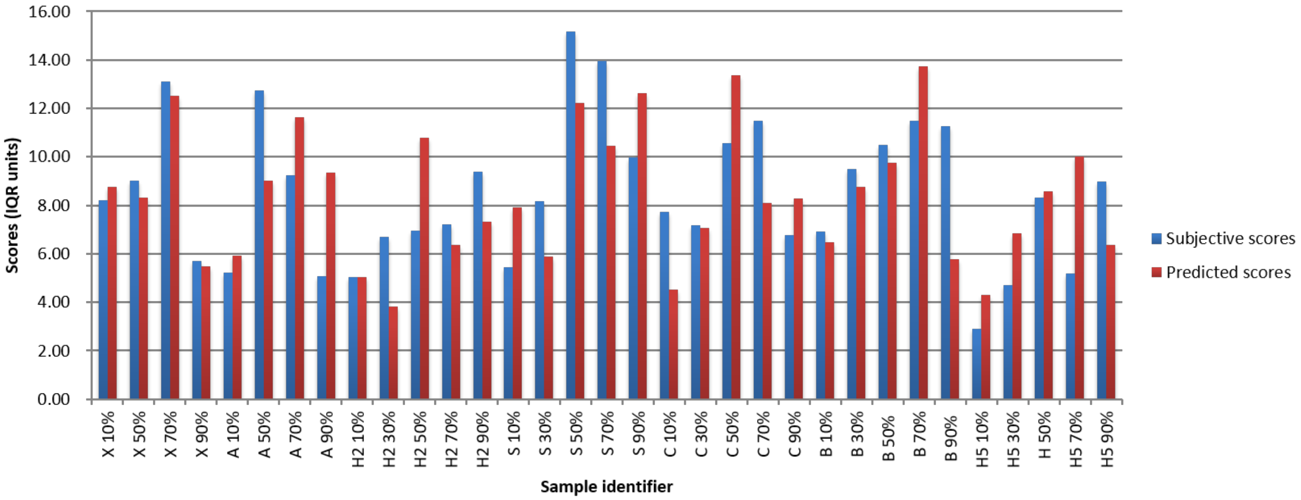


Fig. 2.7. Average training error and testing error linear regression from a 5-fold cross-validation.

Table 2.2.
Summary of the features at each frequency band. There are in total eight scalar-valued features.

Frequency range	Features	Description of Feature
Low	$\overline{\lvert\Delta h\rvert}$	average difference in the heights between two adjacent rectangles
	$\mathbf{max}\{\lvert\Delta h\rvert\}$	maximum difference in the heights between two adjacent rectangles
	$\frac{\overline{\lvert h\rvert}+const}{\sigma(\lvert h\rvert)+const}$	measures the uniformity in the heights of the rectangles
	$\frac{\overline{\lvert w\rvert}+const}{\sigma(\lvert w\rvert)+const}$	measures the uniformity in the widths of the rectangles
Medium	$\sum \frac{\Delta y^2}{\Delta x}$	sum of the non-uniformity of projection
	$\mathbf{max}\frac{\Delta y^2}{\Delta x}$	maximum non-uniformity of projection
High	peaking factor	energy in the neighborhood of the peak / energy in all other areas
	energy concentration factor	peaking factor / width of the neighborhood of the peak

3. CHARACTERIZTION OF 2.5D PRINTING

3.1 Introduction

Conventional printers are constrained to reproduce image content in a 2D space as they are not capable of using any height or surface information that exists in a third dimension. In recent years, digital printing technology has found its applications in extending print to a third dimension to create 2.5D prints, where surface height variance can be incorporated in the print. 2.5D printing technology developed by Océ allows the superposition of several layers of ink on a flat media, which creates a variation in the surface heights according to the input design sent to the printer. The variation of surface heights opens the possibility to create different shapes and texture patterns on top of the traditional printing media. It not only increases the types of objects that a printer can produce but also enhances print's appearance to a large extent. The 2.5D printing can find its significant applications in the reproduction of artwork, and the production of packaging, signage, and decorative materials. For example, the reproduction of oil painting can be more realistic as the surface texture created by 2.5D printing is capable of presenting the oil painting effect.

With the development and application of 2.5D prints, many questions have been raised regarding the quality of 2.5D print, including the printing accuracy and surface appearance. An in-depth characterization of 2.5D printing is necessary to develop a set of quality metrics to objectively measure the printing quality, to investigate the factors that affect printing quality, and to determine the limits in detail reproduction. The results can not only be used to assess the printing quality but also provide help in printing quality prediction and guidance in designing the input files. This paper deals with research towards characterizing 2.5D printing process about the printing accuracy and surface quality. The evaluation is performed on two 2.5D printer prototypes developed by Océ: the wet-on-wet printer, and the wet-on-dry printer.

Currently, few image quality metrics have been designed specifically to address the 2.5D printing quality assessment. The extra dimensionality in height has brought many challenges to the quality evaluation for 2.5D prints. For example, the appearance of 2.5D prints strongly depends on the illumination environment and the viewer's position. This provides an advantage for the 2.5D prints in its applications as decorations, but at the same time requires more complicated design of surface evaluation system. Additionally, the shape and physical structure are big concerns to 2.5D prints, which also need to be assessed to measure the fidelity of the input designs.

Our initial attempts to characterize the 2.5D printing start off with the similarities of the 2.5D printing to the traditional 2D printing and 3D printing. We seek to build the connections that allow us to take advantage of the previous research on 2D image quality evaluation and 3D object quality evaluation. On the one hand, each appearance of 2.5D prints under a different viewing condition can be interpreted as a 2D scene, which can be applied with 2D image quality metrics. Another link can be made by taking the surface height map of 2.5D prints as a 2D image. The surface height map is a grey level image of which the pixel values are the heights of the 2.5D prints at corresponding locations. The examination of the 2D height map measures the potential distortion in the reproduction of heights. A lot of research and discussion have been done from the last several decades (Grice,1999; Chandler, 2013; Wang, 2002; Pedersen, 2011; Moorthy, 2011; Sheikh,2006). On the other hand, some properties of 2.5D prints, such as shape and surface texture, are also critical aspects in 3D prints quality control (Dimitrov, 2006; Pan, 2005; Polzin, 2013). Hence, parallels can be made between reproduction accuracy and surface characteristic of 2.5D prints and that of 3D prints.

In this project, we focus on the evaluation of the 2.5D printing process in shape, structure, and surface. To achieve this goal, we define metrics and develop models that aim to assess 2.5D printing quality in two aspects: overall fidelity and surface finish. For the aspect of overall fidelity, three metrics are developed: modulation transfer function (MTF), difference and root-mean-squared error (RMSE) between the input height map and scanned height map, and print surface angle accuracy. To characterize the surface finish property, we measured the surface roughness, generated surface normal maps, and developed a light reflection model that serves as a simulation of the difference that may be perceived by human observers between ideal prints and real prints. These metrics and models help as initial attempts for 2.5D printing evaluation and invoke discussions for future work. Three sets of test targets were designed and printed by the Océ 2.5D printer prototypes for the calculation of the above metrics: (i) twisted target, (ii) sinusoidal wave target, and (iii) ramp target. The twisted and ramp targets are designed to evaluate both the overall fidelity and the surface finish. We choose suitable metrics or models for the two targets, respectively, according to their surface pattern features. The sinusoidal wave target was used only for the MTF calculation as a probe to the frequency limit a 2.5D printer can achieve.

The rest of this chapter is organized as follows. Section 3.2 gives a brief introduction to 2.5D printing. Following that, the scanning methods are introduced in Sec. 3.3, and three sets of test targets are presented in Sec. 3.4. The quality metrics are described in Sec. 3.5 and their results presented in Sec. 5. In Sec. 6, this paper ends with some conclusions and suggestions for future work.

3.2 2.5D Printing

Similar to the 2D printing technology, the surface color of 2.5D prints is defined by a digital color image file sent to the printer. Beyond that, to define the surface height, a height map of the same size as the color image is generated as a second input to the printer. It is a grayscale image with its pixel values determining the relative heights at corresponding addressable locations in the final print.

We work with two Océ 2.5D printers that use different inkjet technologies, which we refer to as wet-on-wet and wet-on-dry printing. Both printing technologies create surface height variation by accumulating inks on the print media. Each of them produces a distinct type of surface 2.5D and is used for different applications. The wet-on-dry printer is a flatbed printer that uses flat press, yet the wet-on-wet printer uses roller papers. Another difference between the two printers is that the wet-on-dry printer uses UV curable ink, which allows printing higher altitudes than the wet-on-wet printer that uses ordinary ink. The wet-on-dry printer prints the color as a thin layer on top of accumulated white ink, yet the wet-on-wet printer accumulates the color ink directly. Additionally, the height of the final print is calculable with the wet-on-dry printer by a formula, but the wet-on-wet printer is not designed to provide prints with precisely predictable height.

The wet-on-dry printer controls the final prints heights with the input grayscale height map using the following formula

$$A = H \times TU, \tag{3.1}$$

where A is the desired print height, H is the number of the specified unit values, and TU is the thickness unit with a value 8.6 μm for the target print engine. The value of H is determined as follows: The minimum value of H is fixed to be 0 and the maximum value of H is denoted as H_{MAX}, which is to be determined by the user. By mapping the digital pixel values in the input height map to $[0, H_{MAX}]$, we get the value of H for every pixel. Following the calculation using Eq. (3.1), the input height map can be translated to the desired height map with element units of μm.

3.3 Scanning Methods

Two scanning methods are used to obtain the height map of the relief prints. The first scanning method uses the LMI Gocator[1] which uses a red laser to scan the object line by line. It can measure the objects in a minute, and it returns the height map of

---

[1] LMI Technologies Inc., 1673 Cliveden Ave. Delta, BC Canada.

the object in millimeters directly. The resolution achieves $0.017-0.049$ mm along the scanning direction and $0.190-0.340$ mm along the direction of the sample movement. In our experiment, the scanning resolution is set to be 97 points/inch along the laser scanning direction and 254 lines/inch along the object moving direction.

The second scanning method measures surface texture characteristics using a 2D flatbed scanner Canon 5600F [5,6]. It returns a height map with low resolution and is not characterized to provide measurements in physical units. Hence the measure can only be used for comparison between targets. We will refer the unit as 'measurement unit' in the following.

In this paper, we are interested in print quality in the height dimension. The surface height variation of the final print is scanned to generate a height map, from which we develop metrics either with or without reference to the input height map.

3.4 Test Targets

Three sets of targets with different shapes are designed to characterize the two relief printing technologies. In the following, we introduce each test target by showing the designed height map, the associated properties, the relief printer used to print it, and the scanning method.

3.4.1 Twisted target

The twisted target has the height map shown in Fig. 3.1. From left to right, the peak location of the target is shifting from top to bottom. It is printed by the wet-on-dry printer and scanned by the LMI Gocator. It is designed to measure the overall fidelity of the relief printing.

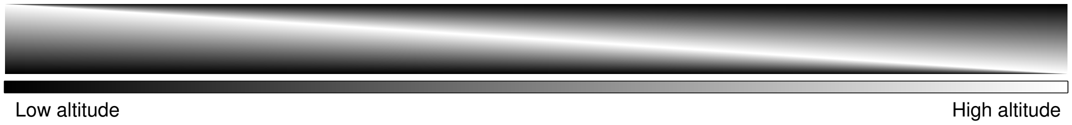


Fig. 3.1. Height map of twisted target, which is to be printed by the wet-on-dry printer.

3.4.2 Sinusoidal wave target

The sinusoidal wave targets are designed to characterize the MTF of both printer prototypes. For the wet-on-dry printer, six sinusoidal wave target sets are designed with different frequency ranges, amplitude ranges, and bias levels. The parameters are shown in Table 3.1.

Table 3.1.
Specification of six sinusoidal wave target sets printed by the wet-on-dry printer.

Target set	Frequency range (cpi)	Amplitude range (Pixel value)	Bias levels (Pixel value)	Maximum droplet setting
A	[7.2 : 7.2 : 57.6]	[1 : 10 : 121]	0	6
B	[57.6 : 7.2 : 108]	[1 : 10 : 121]	0	6
C	[7.2 : 7.2 : 57.6]	[4 : 5 : 64]	127	12
D	[57.6 : 7.2 : 108]	[4 : 5 : 64]	127	12
E	[7.2 : 7.2 : 57.6]	[2 : 2.5: 32]	192	18
F	[57.6 : 7.2 : 108]	[2 : 2.5 : 32]	192	18

Figure 3.2 shows the height map of the sinusoidal wave target set A as an example. It contains an 8×13 array of square sinusoidal patches. Each row consists of eight sinu-

soidal patches with frequencies $f \in \{7.2,\ 14.4,\ 21.6,\ 28.8,\ 36.0,\ 43.2,\ 50.4,\ 57.6\}$ cycles/inch. At each frequency, the patches cover amplitudes $A \in [1:10:121]$ unit of digital value.

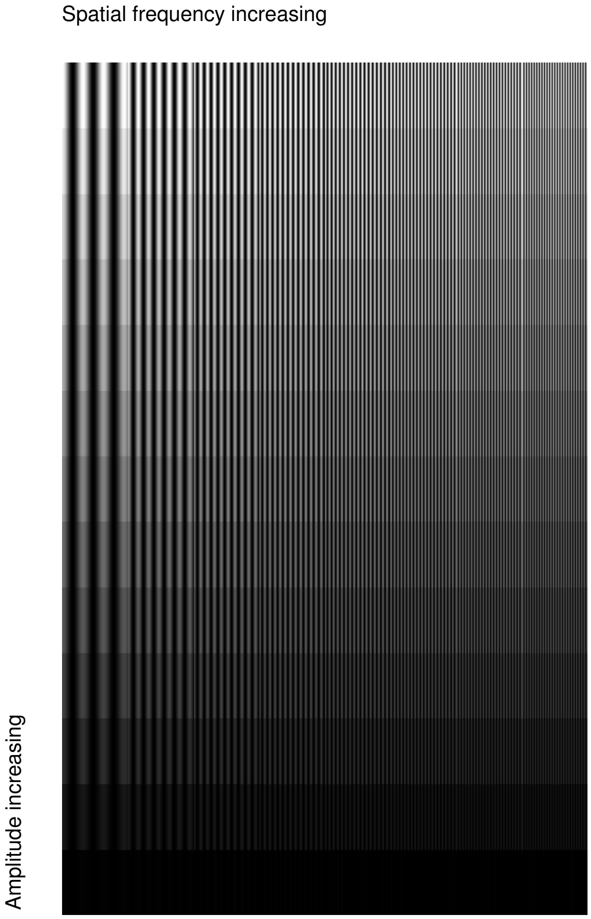


Fig. 3.2. Sinusoidal wave target page designed to measure the MTF of the wet-on-dry printer

The second set of sinusoidal targets was designed to characterize the MTF of the wet-on-wet printer. It consists of 9 patches with frequencies $f \in \{30,\ 50,\ 80\}$ cycles/inch and amplitudes $A \in \{42,\ 84,\ 126\}$ unit of digital value. The input height maps of the nine sinusoidal patches are shown in Fig. 3.3.

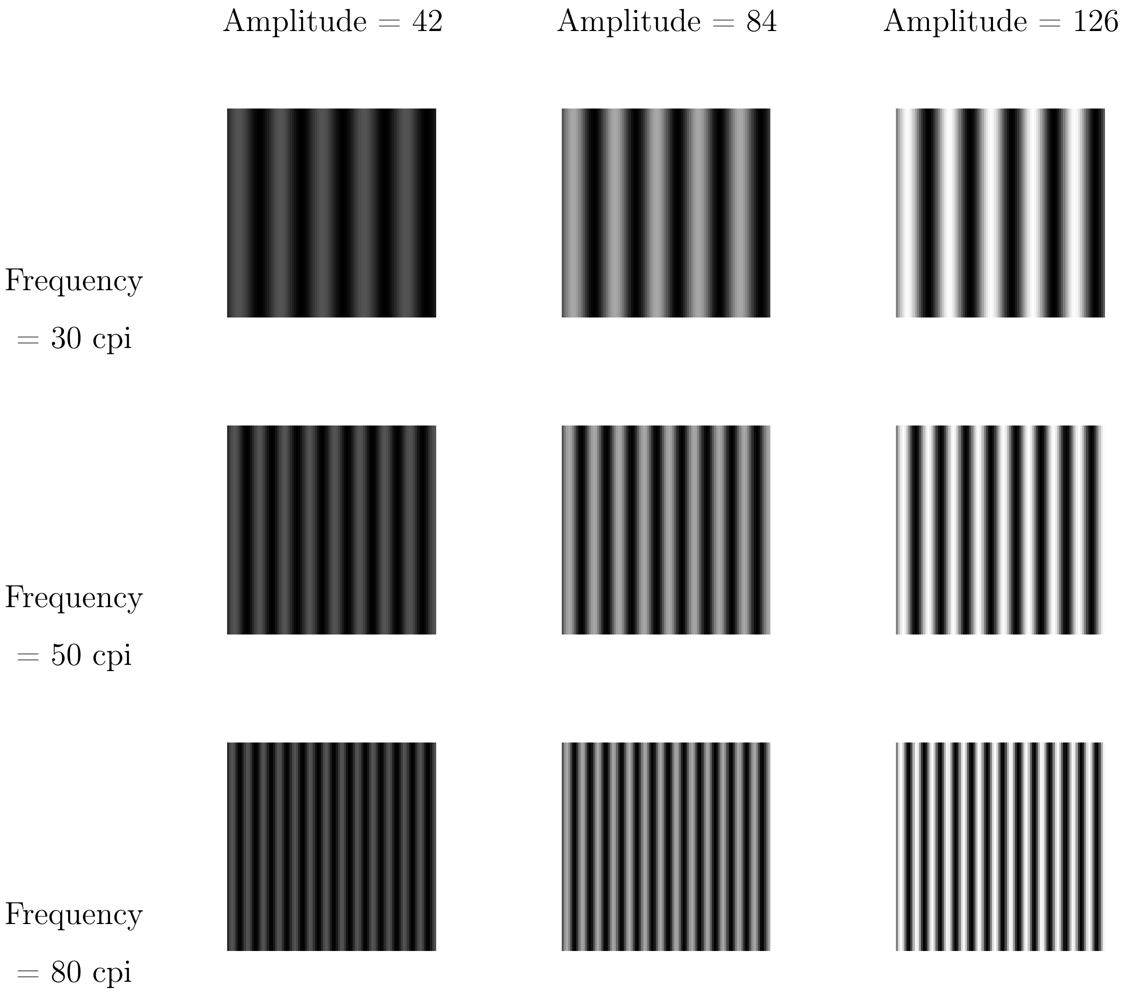


Fig. 3.3. Sinusoidal wave target page designed to measure the MTF of the wet-on-wet printer

3.4.3 Ramp target

The ramp target was designed to measure the angular accuracy and light reflection properties of the wet-on-dry printer. The test page of the ramp target contains one row of cylinder-shaped patches and 8 rows of ramp-shaped patches. The cylinder-shaped patches have the same heights as the peak height of the ramp-shaped patches. They are to be used for calibrating the scanning height. Its height map is shown in

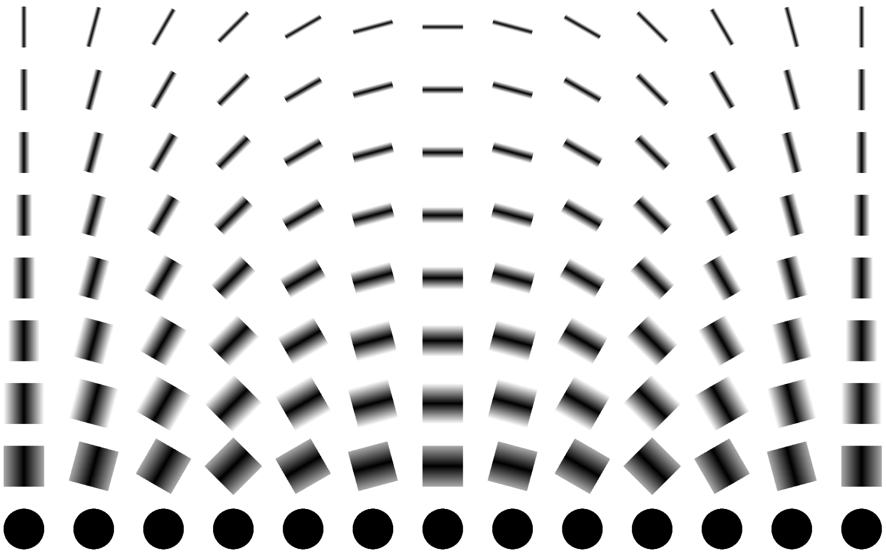


Fig. 3.4. Ramp target designed to characterize the wet-on-dry printer.

Fig. 3.4. The height map of the real prints is obtained by using a flatbed scanner [6].

3.5 Quality Metrics

As with 2D printing, there are always differences between the digital input files and the printed samples in relief printing process. The printed altitude and texture affect the quality and visual experience of the final prints to a large extent. As a result, an in-depth analysis of the height accuracy for relief printing is a necessary step in the calibration of relief printing. In this document, we developed quality metrics to provide an objective evaluation of the relief printing quality regarding the shape and visual experience. The resulted assessment can be used for printing compensation and adjustments.

We approach along two aspects to evaluate the relief printing quality. The first aspect evaluates the accuracy in the reproduction of the input digital image file, i.e., overall fidelity. We adopted three metrics: MTF, difference, and RMSE, and surface angle accuracy. The second aspect concerns about the surface properties of the relief prints that involves the visual experience. We developed three metrics that measure the surface roughness, surface normal map, and light reflection properties of the surface. In the following sections, the proposed methods and metrics will be introduced in detail.

3.5.1 Overall fidelity

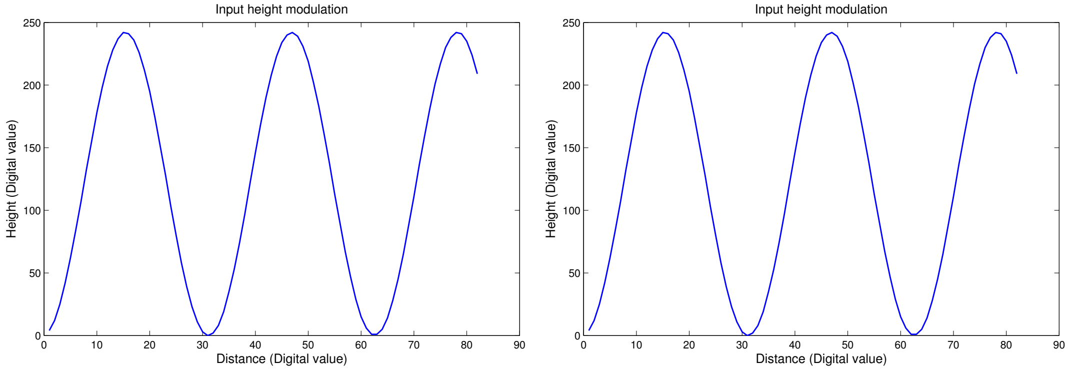
Modulation transfer function - MTF

The MTF of a printing system is a function of frequency or amplitude characterizing how much the system attenuates an input modulation signal [7,8]. A few methods have been developed to measure the MTF of the 2D digital image reproduction devices, including the sinusoidal method, slanted-edge method, and grill method. In this paper, we applied the MTF to evaluate the reproduction accuracy of the height modulation for relief printers. The sinusoidal method is applied to the sinusoidal wave targets to measure the MTF of the printer as a function of frequency and height.

As the frequency increases, it is difficult for the relief printers to reproduce detailed information in the prints. Since the relief printers work in the way that the ink is accumulated layer by layer, the fusion between two layers increases the possibility of dot gain. This is especially the case if the print time between two layers is not enough for the lower layer to cure. Figure 3.5(b) and Fig. 3.5(b) show examples of the height modulation of the input signal and the final prints, from which one can see the smoothing effect in the actual prints due to the movement of the ink. The UV curing accelerates the ink hardening speed, as a result, the wet-to-dry printer which utilizes this technology can produce more detailed structures.

The MTF is defined as the ratio of the output modulation $M(Output)$ to the input modulation $M(Input)$:

$$MTF = \frac{M(Output)}{M(Input)}. \tag{3.2}$$



(a) (b)

Fig. 3.5. (a) Height modulation of the input signal. (b) Height modulation of the output signal.

Figure 3.5(b) shows examples of height modulation for the input height map and output height map of sinusoidal wave targets.

The surface details of relief prints are characterized by the surface height variation. Hence the bias level of the prints does not affect the surface texture accuracy. Based on this fact, the DC component is to be removed from both height maps for modulation calculation. We obtain the frequency response $F(f)$ by extracting the DFT magnitude at the corresponding frequency. The height modulation M is defined as

$$M = F(f)/F_0. \tag{3.3}$$

where F_0 is the normalization factor.

Difference and RMSE

The mean absolute difference and RMSE as low-level features are used commonly in 2D image comparison and 3D printing quality evaluation [9,10]. Here, we calculate the difference and RMSE between the desired height map and the scanned height map. By these two metrics, we aim to have an overall assessment of how strongly the printed heights deviate from the desired heights.

This metric is only applicable for the wet-on-dry printer, since the desired height can be calculated by Eq. (3.1) for the wet-on-dry printer. However, there is no such height mapping equation for the wet-on-wet printer.

The mean absolute difference between two height maps of size $M \times N$ is defined as:

$$MAD = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} |(I_{ij}^{D} - I_{ij}^{S})|, \tag{3.4}$$

where I_{ij}^{D} denotes the desired height map and I_{ij}^{S} the scanned height map. On the other hand, RMSE is defined as:

$$RMSE = \sqrt{\frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} (I_{ij}^{D} - I_{ij}^{S})^2}. \tag{3.5}$$

Print surface angle

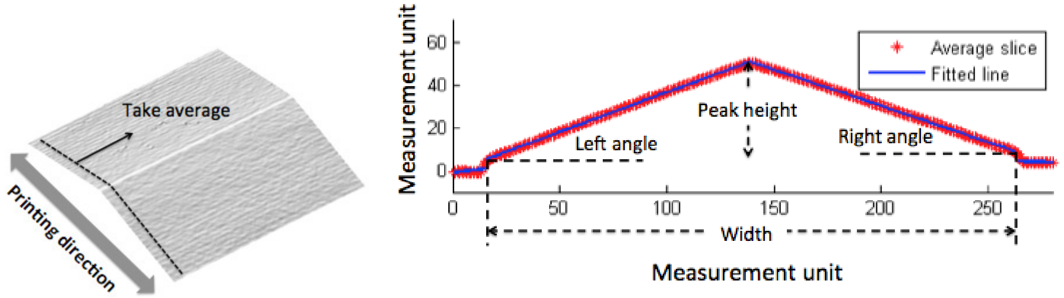
The ramp target was designed to measure the angular accuracy of the relief prints. We have calculated four metrics: $leftangle$, $rightangle$, $peakheight$ and $bottomwidth$. With the scanned height map, surface slices are taken as shown in Fig. 3.6(a) along the indicated direction. An average slice is calculated as the mean of all the slices. We fit the average slice with a straight line and then calculate the four metrics that are illustrated in Fig. 3.6(b). The reason why we separate the left angle from the right angle is that there might exist asymmetries in the prints, which can be explored by comparing the two angles.

3.5.2 Surface finish

Surface roughness

Surface texture plays an important role in the appearance of prints [11]. The metric developed in this section serves as a measure of the texture of the surface.

Our approach extends the ISO standard [12] for line raggedness to 2D to take the entire relief print surface into account. For a surface of interest, such as that shown in



(a)

(b)

Fig. 3.6. (a) Surface slices on the ramp target are taken as shown in the dashed line along the direction indicated by the arrow. (b) Definitions of the four metrics: left angle, right angle, peak height, and width.

Fig. 3.7(a), we generate a surface roughness profile by applying a 2D high pass filter to it, as shown in Fig. 3.7(b). We then calculate the difference between the surface roughness profile and its mean to obtain a 2D residual plane of size $M \times N$, which is denoted by e_{ij}. Two metrics are calculated: the first one is the root mean square value of the 2D residual, and the second one is the maximum among the residuals.

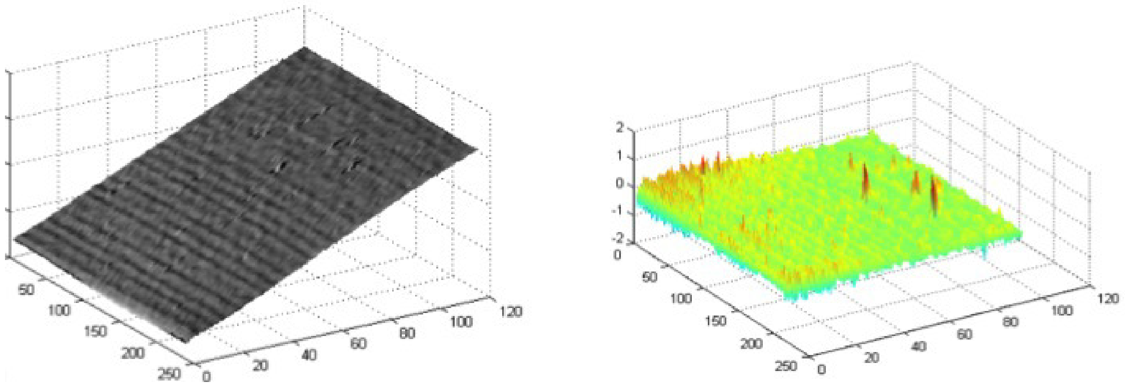
$$R_{RMS} = \sqrt{\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} e_{ij}^2}, \tag{3.6}$$

$$R_{MAX} = MAX(e_{ij}). \tag{3.7}$$

Surface normal

The surface normal vectors affect the light reflection, and hence the appearance, of the surface. We are interested in depicting the surface texture characteristics by computing the normal vectors from the height maps.

Our approach is to generate a surface normal map by rescaling each of the x, y, and z components of the surface normal vectors to [0, 255] and using the rescaled x,



(a) (b)

Fig. 3.7. (a) Surface of interest for the ramp target. The height of the surface is in measurement units. (b) Surface roughness profile in measurement units.

y, and z components as the R, G, and B values in the RGB color space of the surface normal map. By this conversion, the x, y, and z components of the surface normal vectors are transferred to R, G, and B values and can be visually displayed as the surface normal map. This surface normal map provides a visual description of the print surface texture and a comparison to the desired surface texture calculated from the input height map.

Light reflection simulation

Relief prints are different from 2D prints in that they have surface height variation, which leads to varying visual experiences under different viewing conditions including the incident illumination, and different print properties, such as the surface Bi-directional Reflectance Distribution Function (BRDF), and the surface normal [13–16]. We simulate the surface appearance by calculating the light reflection from the height map using the Phong reflection model [17]. Furthermore, with the input height map and the scanned height map, we can simulate both the input and printed appearance and compare the difference between them. To extract metrics that describe the difference, we calculate the difference between the input and output light reflection. A threshold is applied to the absolute value of the difference; so we can select out the high difference values that would capture our attention. We call the values that are above the threshold 'peaks', and the ones that are below the threshold 'background'.

Four metrics are developed based on the absolute light reflection difference, which we denote as D_{ij}. The first one is the root mean square value of the difference D_{ij} between the input and the scanned height maps. The second one is the contrast, which is defined as the ratio of the sum of the peak values to the sum of the background values. The third metric is called density for which we take the sum of the peak values divided by the area of the peaks. The last metric is the density contrast which is the ratio of the density of the peaks to the density of the background.

Table 3.2.
Four metrics based on the light reflection calculation.

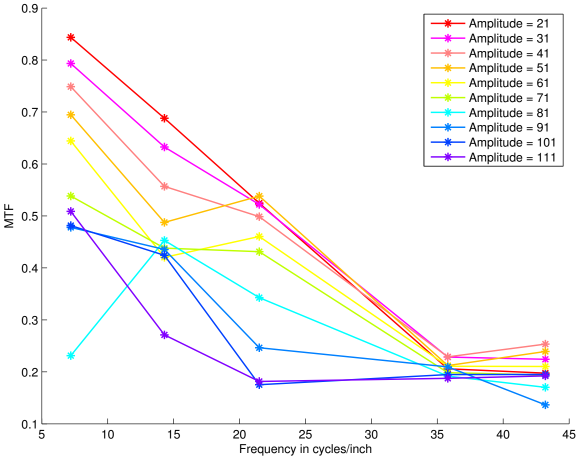
Root Mean Square Error	$\sqrt{\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} D_{ij}^2}$
Contrast	$\frac{\sum Peak\ \ values}{\sum Background\ values}$
Density	$\frac{\sum Peak\ values}{\sum Peak\ area}$
Density Contrast	$\frac{Density\ of\ the\ Peaks}{Density\ of\ the\ Background}$

3.6 Results

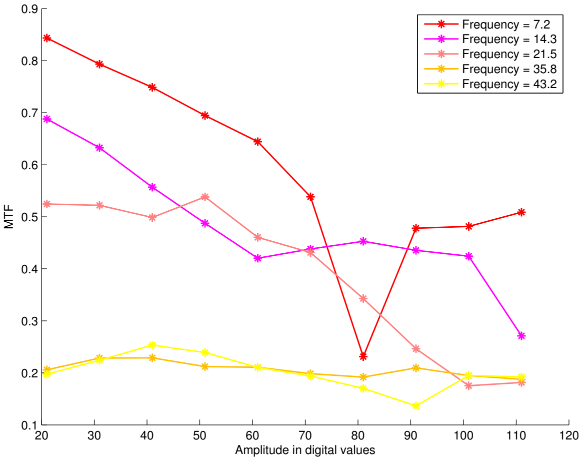
3.6.1 Overall fidelity

Modulation transfer function

We calculated the MTF for the sinusoidal wave target set I printed by the wet-on-dry printer and the result is shown in Figs. 3.8(a) and 3.8(b). It can be seen in



(a)



(b)

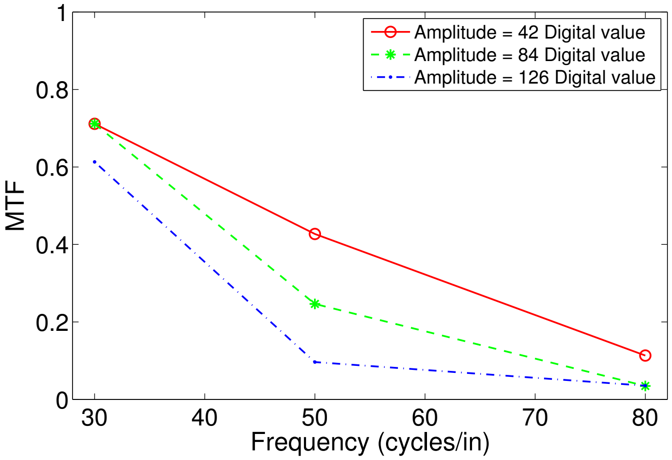
Fig. 3.8. MTF for part of sinusoidal wave target set I printed by the wet-on-dry relief printer. (a) MTF as a function of frequency for 10 fixed amplitudes. (b) MTF as a function of amplitude for 5 fixed frequencies.

Fig. 3.8(a) that the MTF decreases as the spatial frequency of the target increases. The result only shows the MTF for frequencies under 45 cycles/inch, as the measurement contains too much noise when the frequency exceeds 45 cycles/inch and no principle component could be selected from the Fourier Transform. The frequencies where the MTF drops below 0.5 can be considered as cutoff frequencies for the printer to reproduce fine details. We also observe the dependence of the MTF on the input amplitude in Fig. 3.8(b). At frequencies lower than 30 cycles/inch, the MTF decreases with increasing amplitude. However, at around 30 cycles/inch, the MTF shows no dependence on the amplitude and remains near a low value of 0.25. Other than the noise introduced by the printing process, the poor performance of the MTF at high frequencies may also be due to the low scanning resolution and quality. One data point appears to be an outlier, possibly due to errors in the data acquisition as seen in Figs. 3.8(a) and 3.8(b) on the curve for amplitude 81 at 7.2 cycles/inch. A better scanning method and scanner calibration might resolve this issue.

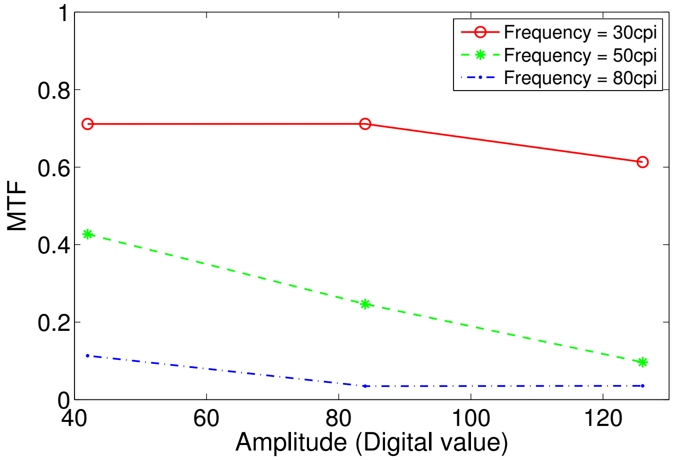
The MTF for the sinusoidal wave target set II printed by the wet-on-wet printer is shown in Figs. 3.9(a) and 3.9(b). From Fig. 3.9(a) we can see that the MTF decreases with increasing frequency and is as low as $0.1-0.2$ at 80 cycles/inch. Figure 3.9(b) shows how the input amplitude affects the MTF. It reveals that the amplitude has a larger influence on the MTF when the frequency of the signal is around 50 cycles/inch. The performance is consistently good at the low frequency (30 cycles/inch) and consistently poor at the high frequency (80 cycles/inch). Note that the media used by the wet-on-wet printer is roller paper, as a result, surface waviness exists and will be reflected in the scanned height map as low-frequency noise. The noise is extracted by a low pass filter and removed from the scanned data.

Difference and RMSE

Difference and RMSE are calculated for the twisted target printed using the wet-on-dry printer. Figure 3.10 shows a difference map obtained by subtracting the



(a)



(b)

Fig. 3.9. MTF for sinusoidal wave target set II printed by the wet-on-wet relief printer. (a) MTF as a function of frequency for 3 fixed amplitudes. (b) MTF as a function of amplitude for 3 fixed frequencies.

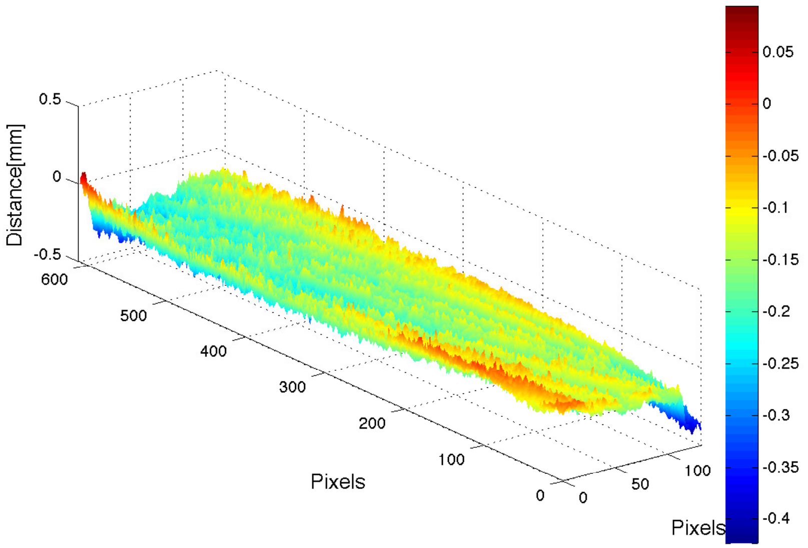


Fig. 3.10. Difference map for the twisted target printed using the we-on-dry printer. It is calculated by subtracting the scanned height map from the desired height map.

scanned height map from the desired height map. For most of the area in the difference map, the difference is negative meaning that according to the scanned data, the final print is mostly higher than the desired print. This is because the height mapping equation Eq. (3.1) is just an approximation of the final print height and needs to be calibrated. If more data were available, the height difference from our calculation could be used to adjust the height mapping equation. We also observe red regions along the edges and at the corners of the twisted target prints in Fig. 3.10, which indicates that the prints are attenuated more in those areas than in the center areas.

Table 3.3 gives the results of the two metrics for the twisted target printed with the wet-on-dry printer. The height range of the scanned height is $0.117 - 2.675$ mm so we can conclude that the values of the mean absolute difference and RMSE are small. Although there are fluctuations over the whole print, the height is mostly reproduced with a low percentage error.

Table 3.3.
Mean absolute difference and RMSE for the twisted target.

Metric	Absolute Value	Relative Error [†]
Mean absolute difference	0.174 mm	6.49%
RMSE	0.182 mm	6.81%

[†] (Absolute value divided by range)×100

Print surface angle

With the limited scanning data, we compare the angular accuracy of the ramp targets that were printed using the wet-on-dry printer as shown in Fig. 3.11. The results are shown in Fig. 11. Note that the ramps are designed as follows: From ramp 1 to ramp 7, the target angles increase linearly, the bottom widths decrease linearly, and the peak heights remain the same.

In Fig. 3.12(a) we see a nearly linear increase in angle from ramp 1 to ramp 7. The left angles are always slightly larger than the right angles, which indicates a constant asymmetry in the geometry of the prints. The variance in peak heights across the seven ramps shown in Fig. 3.12(b) is small. In Fig. 3.12(c), the width drops nearly linearly from ramp 1 to ramp 7. From the results, we conclude that there exists a position dependence on the printing direction of the geometry of the relief prints.

3.6.2 Surface finish of relief prints

Surface roughness

Surface roughness is calculated for the ramp targets shown in Fig. 3.11. It is calculated separately for the left and right parts. Two histograms are generated for the values of R_{RMS} and R_{MAX}, as shown in Figs. 3.13(a) and 3.13(b). Both metrics show an increasing trend from ramp 1 to ramp 7, indicating that the surface becomes

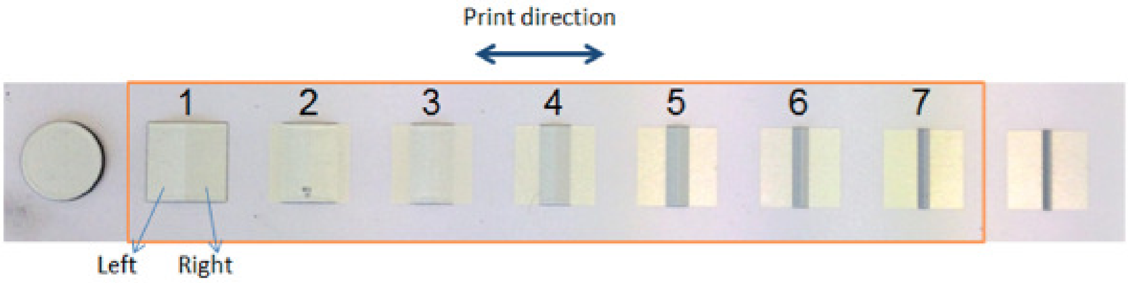


Fig. 3.11. Ramp targets used for calculation of angular accuracy. This is an image of the targets that have been printed using the wet-on-dry printer. The left and right side are defined as noted.

rougher as the slope becomes steeper. However, it is worth noting that in Fig. 3.13(b), which shows the maximum of the residuals, ramp 2 has a very high value. This is due to a significant local print artifact in ramp 2. But this local defect is not picked out in the RMS value of the residuals. To determine which metric to use, a psychophysical experiment is needed to relate the metric values to human perception.

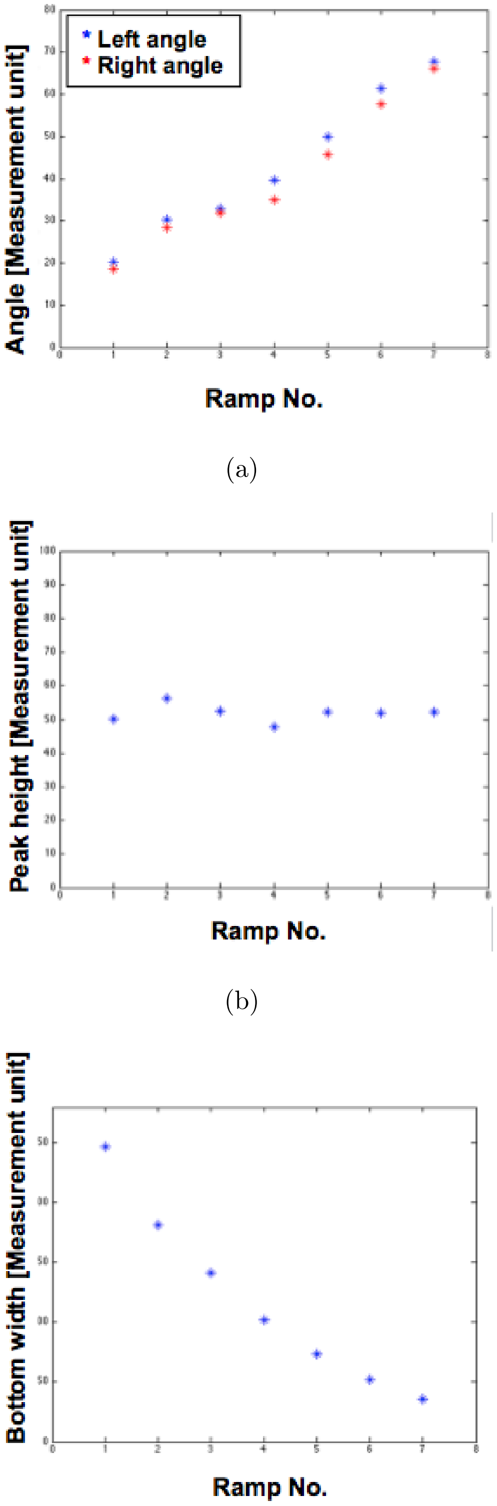
Surface normal

The surface normal maps are generated for the twisted target for both the input height map and the scanned height map obtained from the twisted target printed by the wet-on-dry printer. The result is shown in Fig. 3.14(a), which provides a visualization of the surface normal vectors. The curved banding pattern in the surface normal map from the scanned data depicts the surface roughness of the real print. This variation aligns very well with our observation of the real print. By comparing with the surface normal map from the input data, we can describe what the differences are and the shape and location of artifacts.

Light reflection simulation

We applied light reflection simulation to the ramp targets printed by the wet-on-dry printer, aiming to obtain a heuristic visualization of the surface characteristics for both the ideal and real prints. Some comparisons are provided in Fig. 14 between the surface appearance of the photo of the real print, the light reflection simulation based on the scanned height map, and the one based on the input height map. With limited control of the experimental environment, we tried our best to match the light direction and color between each photo and the corresponding simulation. We can see that the light reflection simulations work well to match the photos of the real print. As expected, the same print exhibits different surface appearances under different lighting conditions and viewing angles. The defects that are obvious under a certain lighting condition might not be visible if the illumination direction changes. We expect that the light reflection simulation can provide a fast mean for visualization of the real prints from the scanned measurements.

The four metrics based on the light reflection calculation for ramp 1 to ramp 7 are presented in the histograms in Fig. 3.16. Of the four metrics, the density metric yields a very similar result to the RMS value of the residuals in Fig. 3.13(a) that describes the surface roughness. As the slope of the ramps becomes steeper, the values of the metrics increases. Again, ramp 2 stands out in the values of the RMS, density, and density contrast due to the local print defect. Among the four metrics, the density metric aligns best with our visual observations. However, as mentioned in the previous section, a psychophysical experiment is desired to measure and validate the metrics we have developed.

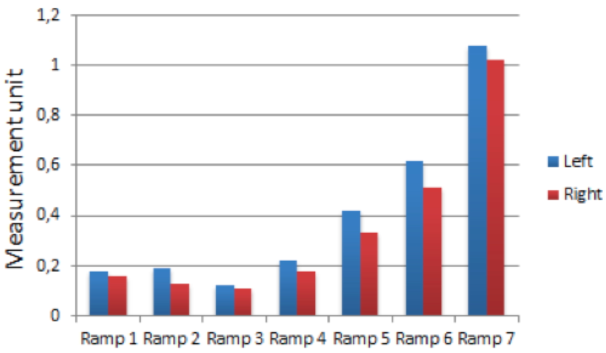


(a)

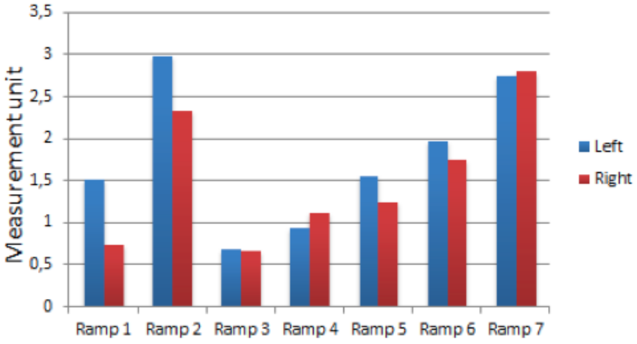
(b)

(c)

Fig. 3.12. Measurement for the ramp target shown in Fig. 3.11 printed with the wet-on-dry printer. (a) angles, (b) peak height, and (c) width.

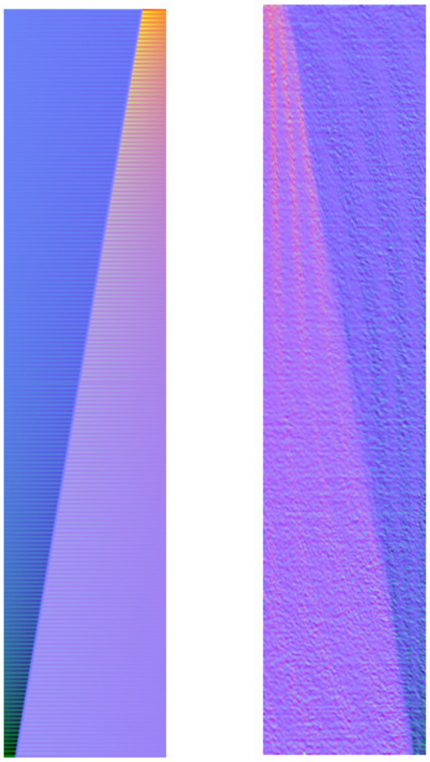


(a)



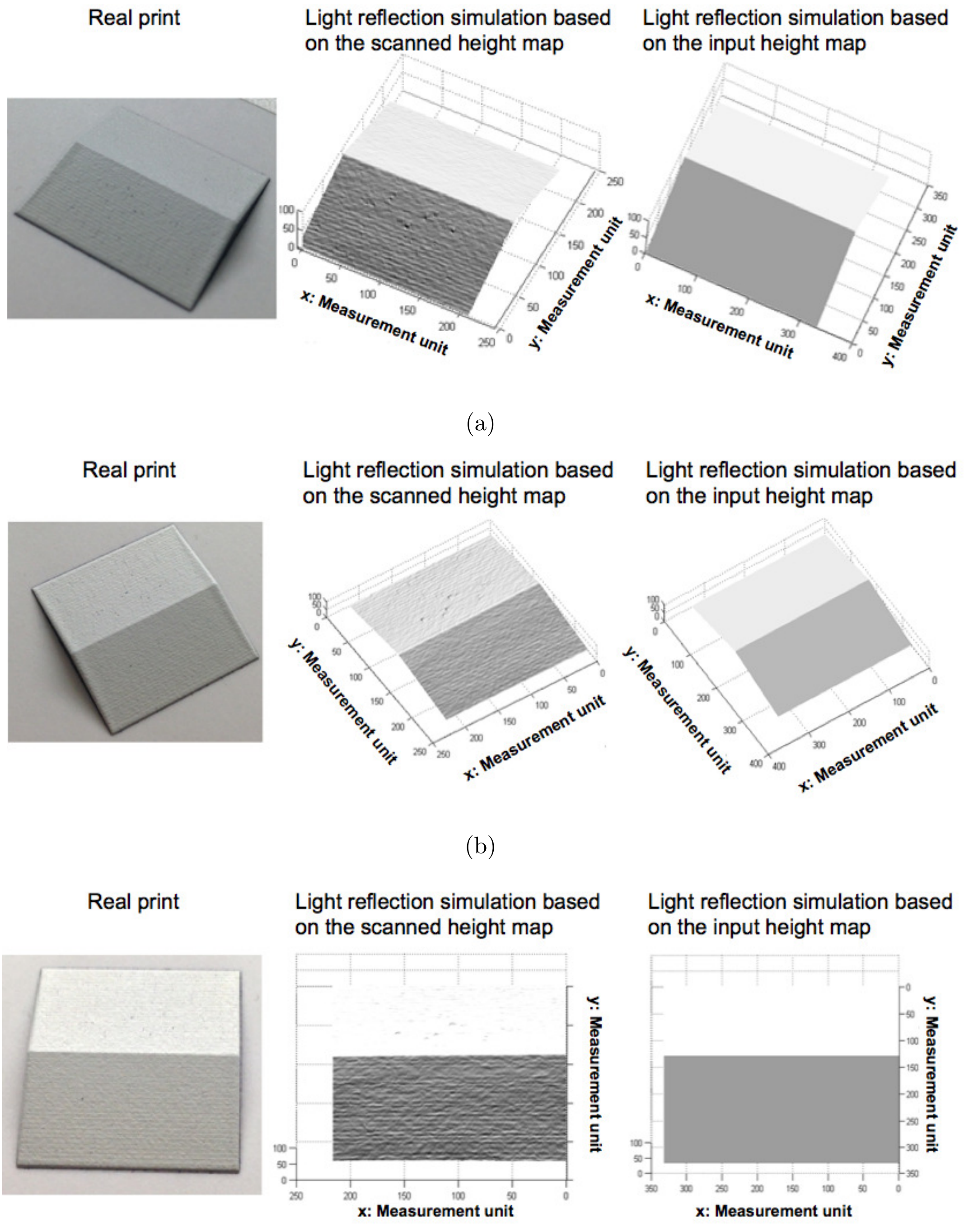
(b)

Fig. 3.13. Residuals for the ramp target shown in Fig. 3.11 printed with the wet-on-dry printer: (a) RMS value of the residuals, (b) Maximum value of the residuals.



(a) (b)

Fig. 3.14. Surface normal maps for the twisted target: (a)From the input height map. (b)From the scanned height map of the twisted target printed by the wet-on-dry printer.

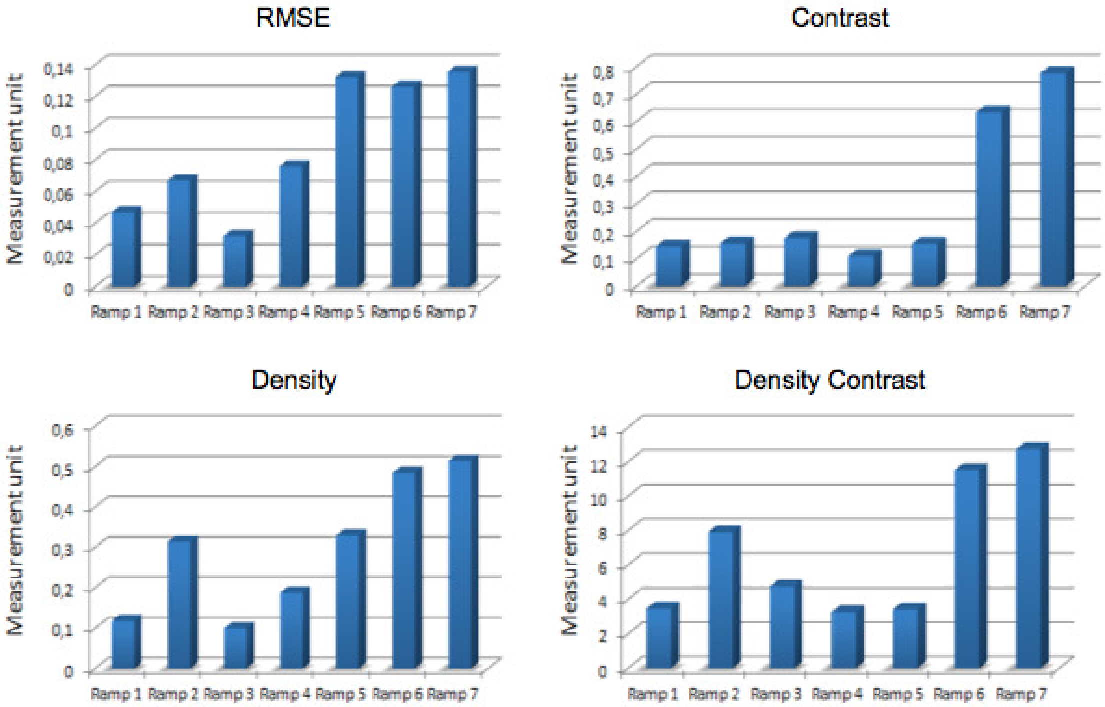

Real print
Light reflection simulation based on the scanned height map
Light reflection simulation based on the input height map
x: Measurement unit
y: Measurement unit

(a)

(b)

(c)

Fig. 3.15. Surface light reflection comparison. (a), (b), and (c) are with different lighting conditions and viewing angles.


RMSE

Contrast

Density

Density Contrast

Fig. 3.16. Histograms of the four metrics: root mean square error, contrast, density, density contrast.

4. FEATURE BASED STREAK DETECTION IN SCANNED IMAGES

4.1 Introduction

Streaks caused by dust on the scanner sensor are one type of major defects in the scanned images. Detecting the streak defect is a necessary yet challenging task. Accurate detection of the streak is the prerequisite for the defection analysis and a successful following up streak healing. Defect detection has been a widely explored topic in printed and scanned image [18–20]. However, the streak caused by dust on scanned images possesses characteristics which differentiate it from other defect types. Fig. 4.1 shows a cropped patch containing streak. Due to the small size of the dust, the streak has a very thin structure and could be barely perceived even by human eyes from a distance. Majority of the streaks caused by tiny dust are observed to be interacting with the pixels in the scanned page. Hence, the appearance of the dust varies depending on the background. As the dust stays at the same position most of the time, the steaks are vertically straight lines along the scanning direction. The thin structure of the streak and the interaction with background pixels bring lots of challenge to the detection task. We have developed methods which consider the characteristic of the steak in both features and classification methods.

4.2 Related Works

Research on streak detection can be seen in both printed images and texture analysis. In [18] a projection-based method is proposed to detect the streak in the printed images with mottle and uniform noise. The projections across the entire image at different window sizes are computed. Subsequently, mutual information is used to

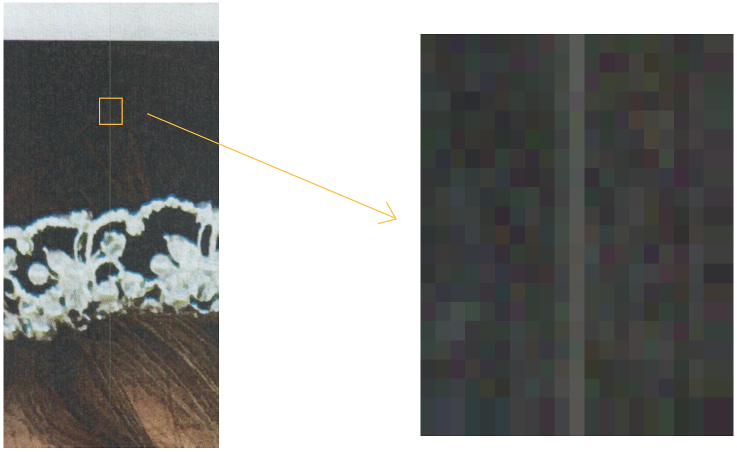


Fig. 4.1. An example of image patch that contains a streak. From the zoomed-in image on the right, it can be seen that the streak is about two pixels in width, and has a very faint appearance.

detect the peak location and peak sizes. [19] describes a method to characterize the streak in printed images using wavelet decomposition. Streaks are considered as one type of macro-uniformity defects in [21] and are measured regarding objectionableness. More streak detection literature can be found in texture analysis. [20] proposed a framework for detecting streaks and lines on polished or textured surfaces using line integrals.

4.3 Feature Based Streak Detection

A block diagram of the proposed detection framework is provided in Fig. 4.3. A detection window is used to slide over the input image in the horizontal direction. The size of the detection window is typical $M \times w$ pixels, where M is the number of rows of the input page, and w is a parameter taking values between $7 - 10$ pixels. Fig. 4.2 shows an example of the detection window with a width of 13 pixels. The streak is detected within each detection window on a row basis, i.e., each row is examined to determine if a streak defect exists or not. The intensity of each row within a detection window is called a row profile. Denote the row profile as $f_m(x)$, where m is the row index and x is the horizontal coordinate. The detection window can then be expressed as $\{f_m(x): \ 1 \leq m \leq M\}$ The features are extracted for each row, followed by a binary classification which determines if the row includes a streak defect.

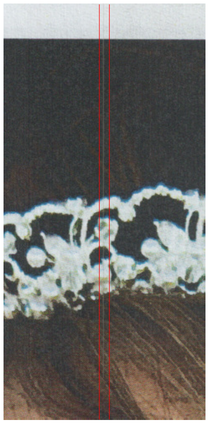


Fig. 4.2. A detection window of 13 pixels in width. The image is scanned along the horizontal direction with the detection window. Within each detection window, the streak is detected on a row-basis. The intensity of a row in the detection window is defined as a row profile, based on which the feature are extracted.

4.3.1 Pre-processing

Halftone descreening using Gaussian filtering

For most of the cases, the documents to be scanned are printed pages. In general, the printed documents suffer from underlying halftone patterns [22, 23], which are preserved during the scanning process and interfere with the defect analysis on the scanned pages. It is essential to eliminate the effect of the halftone screen before the application of any streak detection algorithms. Meanwhile, any descreening algorithm should avoid destroying the fine structures that are intrinsic to the content image.

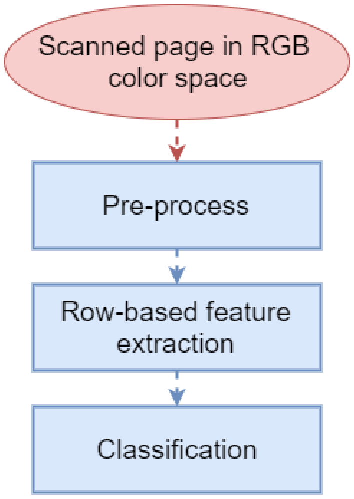


Fig. 4.3. A brief overview of the streak detection framework.

To this effect, the image is applied with a Gaussian low pass filtering which has a gentle cutoff of the high-frequency components. A Gaussian filter has an impulsive response which is a Gaussian function. The kernel of the Gaussian filter is chosen to be corresponding to the halftone frequency. Considering the dust on the sensor are usually thin and subtle, the resulting streaks only span several pixels in width. To avoid smoothing out the streak structure by the use of Gaussian filtering, we apply a 1D Gaussian low pass filtering alone the vertical direction. Denote the input image as $\mathbf{I}(x, y)$, where x and y are the discrete spatial-domain coordinates; the Gaussian filtering can be expressed using the following equation:

$$\mathbf{I_G}(x, y) = \mathbf{I}(x, y) * G(y), \tag{4.1}$$

where $G(y)$ represents the 1D Gaussian filter defined as

$$G(y) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{y^2}{2\sigma^2}} \tag{4.2}$$

Noe that for RGB input images, the Gaussian smoothing is applied to each of the R, G, and B channels.

4.3.2 ΔE

The scanned pages are RGB color images. To characterize the human visual perception, the images are converted into the CIE L*a*b* color space, which mimics the visual response of the eyes. Moreover, the difference between the fine frequency component and the coarse frequency one is of interest in our work, as the finely structured streaks are the target to be detected. Hence, the image in the CIE L*a*b* color space is further converted into the ΔE space. The ΔE is defined, according to CIE76, as the Euclidean distance between two colors. In our work, the two colors represent the high frequency and the low-frequency components of the image, respectively. The high-frequency component is just the image itself. The low-frequency component is obtained on a row basis. At every row of the L*a*b* image, a baseline is computed for each color component by applying a median filter of size p. Denote the components in the three color spaces as L$^*(x, y)$, a$^*(x, y)$, and b$^*(x, y)$. The baseline image can be expressed as

$$\begin{aligned} L_B(x, y) &= \underset{x' \in S_x}{\text{median}}(L(x', y)) \qquad S_x = x - p, ..., x + p \\ a_B(x, y) &= \underset{x' \in S_x}{\text{median}}(a(x', y)) \qquad S_x = x - p, ..., x + p \\ b_B(x, y) &= \underset{x' \in S_x}{\text{median}}(b(x', y)) \qquad S_x = x - p, ..., x + p. \end{aligned} \tag{4.3}$$

The ΔE between the original image and the baseline is then

$$\Delta E^* = \sqrt{(L^* - L_B^*)^2 + (a^* - a_B^*)^2 + (b^* - b_B^*)^2}. \tag{4.4}$$

The ΔE space is invariance to the illumination changes, which is an important factor in the streak detection in the unconstrained environment with a variety of pages to be scanned. Fig. 4.4 shows a pre-processed output of the image patch in Fig. 4.2. In practice, ΔL is often used as it also represents the perception difference and easier to compute. In the experiment, we test with both ΔE and ΔL and compare the results.

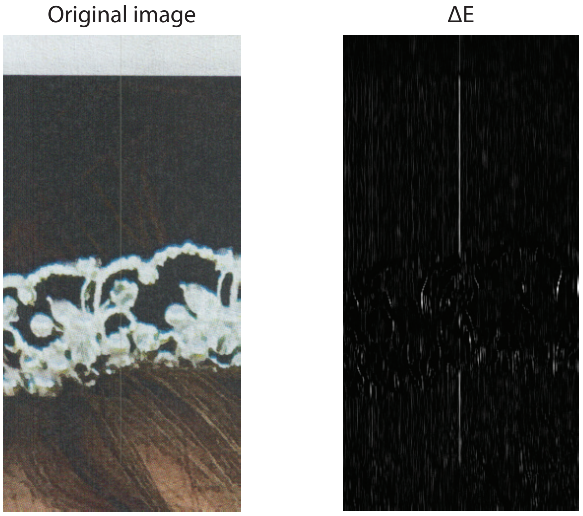


Fig. 4.4. Left: a patch of the image that contains a thin streak in the center; Right: the output in the unit of ΔE after applying Gaussian filtering and color space conversion.

4.4 Feature Extraction

The proposed streak detection algorithm uses a feature-based method. Features are extracted to characterize the streak based on which the classifier is able to identify the streak in the mist of other thin vertical structures intrinsic to the document. The design of the features takes into consideration the vertical structure of streaks, the low contrast between the streaks and the neighborhood pixels, and the relationship between the adjacent rows. Following the above guidelines, two features are proposed: the average peak location derivative (APLD) and the modified Local Binary Pattern (LBP). As illustrated in Fig. 4.5, the APLD features are extracted following a matched filtering on the row profile, while the MLBP feature is computed directly from the detection window.

4.4.1 Average Peak Location Derivative (APLD)

The APLD feature is proposed based on three observations of the streak defects: (1) the streak usually spans several pixels (1-5 pixels) at each row and has a peak

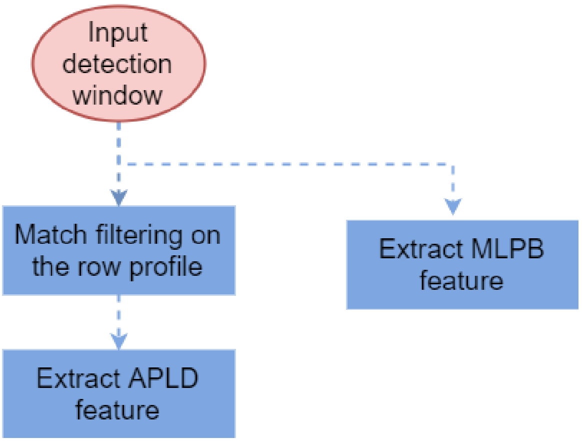


Fig. 4.5. Overview of the feature extraction process.

(maximum or minimum intensity) at the center; (2) if a streak is caused by one single dust, its pattern along the row direction merely changes across the rows, i.e. along the vertical direction; (3) if a streak caused by one single dust, its horizontal location rarely shifts across the rows. There is one rare situation where the dust might have rolled during the process of scanning, in such case the streak is found to shift by $1-2$ pixels.

Matched filtering

Before introducing the APLD feature, we first provide a brief review of the matched filtering which provides the signal to the APLD feature extraction. Matched filtering is a common solution for detecting the desired pattern in the presence of noise. It is optimal in the sense of maximizing the signal to noise ratio (SNR). The streak caused by the dust has a low contrast to the context, a matched filtering is applied aiming to emphasize the shape a the streak.

As introduced above, the row profile is the pixel intensity along the row direction within a detection window. Usually, the streak has a peak shape reflected in the pixel density along the row, as shown in the example provided in Fig. 4.6. If we shrink the range to the size of a detection window, the streak has a hump shape as shown in

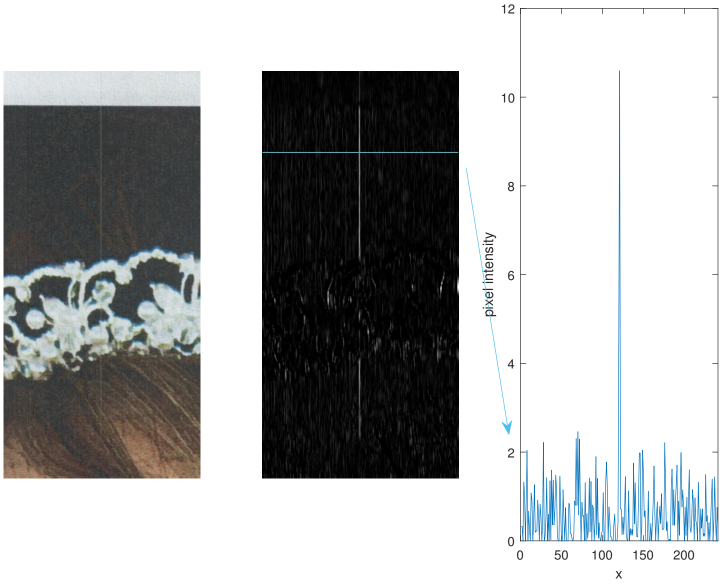


Fig. 4.6. Left: a patch that contains a streak; Middle: pre-processed output; Right: pixel intensity at the 100th row.

Fig. 4.7. We name the pixel intensity at each row within a detection window a 'row profile'.

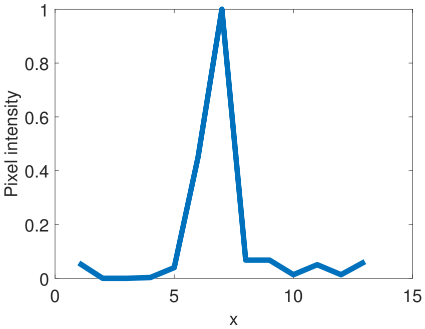


Fig. 4.7. Row profile of a streak. The length of the row profile equal to that of the detection window.

To highlight the hump structure, the matched filter is the average of the streak row profiles from the annotated images. Because averaging is a statistic operation, the matched filter we extracted is a statistical model of the streak. Fig. 4.8 plots the extracted streak profile which is normalized so that the sum is one.

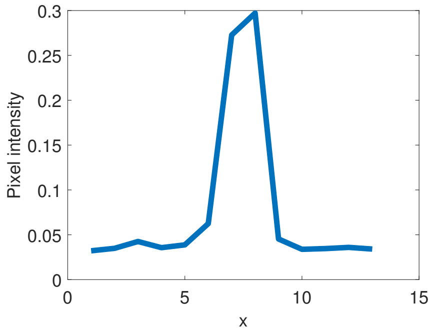


Fig. 4.8. Averaged streak extracted from the groundtruth.

Use $h[k]$ to denote the matched filter, and a row profile of interest as $s[k]$, the matched filtering can be expressed as

$$y[n] = abs(\sum_{k=-m}^{m} h[n-k]s[k]), \tag{4.5}$$

where m equals to one half of the width of the detection window, and $y_i[n]$ denotes the magnitude of the matched filtering output. It should be noted that the streak might be darker or lighter than the background. To introduce invariance to the contrast, only the magnitude of the matched filtering result is used. Fig. 4.9 shows the matched filtering output by applying the matched filter in Fig. 4.8 on the row profile in Fig. 4.7.

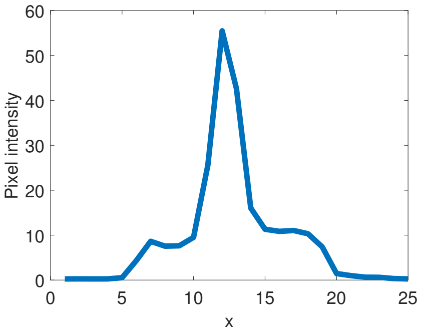


Fig. 4.9. Matched filtering output.

We apply the filter to each row following a densely scanning fashion. Usually, the signal-to-noise ratio (SNR) is computed from the matched filtering result to measure the similarity of the signal to the matched pattern. However, the streaks caused by dust are mostly subtle with low contrast to the neighboring pixels. Therefore, we need a more efficient means to distinguish the pattern from the noisy background. One of the characteristics of the streak mentioned above states that the streak is mostly straight with little variation in the shape and location, which indicates that the position, instead of the intensity, of the peak, could also serve as an efficient feature to identify the streak.

Average peak location derivative (APLD)

As the streak has a straight vertical structure, the corresponding peak location of the matched filtering output rarely changes within the detection window. Fig. 4.10 and Fig. 4.11 plot the peak locations for the detection windows with and without streaks. It can be seen that, even with the presence of noisy background, the peak location at streak locations shows a strong consistency across the rows. On the other hand, in the detection window without any streak, the peak locations are randomly distributed with a high variance.

Use $g_k[i]$ to represent the matched filtered output at the kth row, where i is the horizontal coordinates. An example of $g_k[i]$ is shown in Fig. 4.9. The peak locations at all rows in the detection window are denoted by $p(k)$. To describe the consistency of the peak location in the matched filtering output, we propose to compute the peak location derivative (PLD), which is the local variance of the peak locations in a neighborhood of N rows, where N is a hyper-parameter. The locally averaged peak location at each row k can be expressed as

$$\overline{p}(k) = \sum_{k'=i-\frac{N}{2}}^{k'=i+\frac{N}{2}} p(k'). \tag{4.6}$$

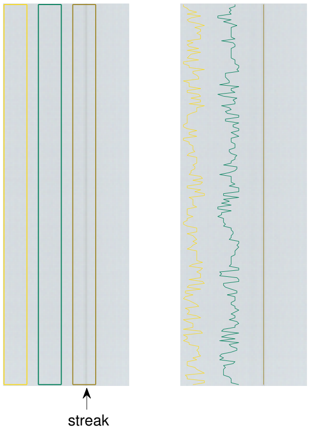


Fig. 4.10. Peak locations from the matched filtering output. Left: A patch that contains a streak. Three detection windows are marked in yellow, green, and brown colors, respectively. Right: For each of the detection windows, the peak locations at each row is marked using the corresponding color. The brown detection window contains a streak. Hence the peak locations are consistent.

Based on the local average, the local variance of the peak locations can be computed by

$$\overline{d}(i) = \frac{\sum_{k=i-\frac{N}{2}}^{k=i+\frac{N}{2}} \left[p(k) - \overline{p}(i)\right]^2}{N+1}. \tag{4.7}$$

4.4.2 Modified Local Binary Patterns (M-LBP)

Local binary pattern (LBP) is one of the best performing features in texture examination. It is invariant to monotonic gray-level changes and has low computational cost. The basic LBP operator assigns a binary number to every pixel in an image by thresholding the neighboring pixels with the center pixel value and takes the eight labels as the feature vector. The numerical value that corresponds to the eight binary numbers can be used as a single feature value for each pixel. Let $\vec{x}$ denote the location of the pixel of interest. Use $\vec{d_j}$ to denote the eight neighborhood pixels in a clockwise

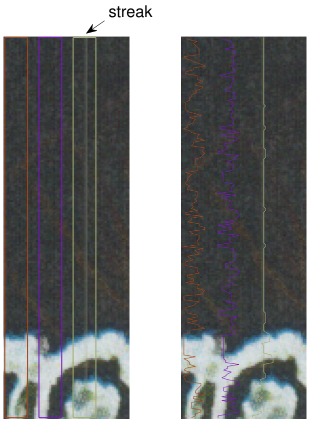


Fig. 4.11. Peak locations from the matched filtering output. Left: A patch that contains a streak. Three detection windows are marked in red, purple, and yellow colors, respectively. Right: For each of the detection windows, the peak locations at each row is marked using the corresponding color. The yellow detection window contains a streak, hence the peak locations are consistent. At the location where there is an edge that disturbs the streak, the peak location also changes.

direction with j being the neighborhood index. The binary feature vector **T** for the pixel x consists of elements $t(\vec{d_i})$ as follows:

$$t(\vec{d_i}) = \begin{cases} 0, & \text{if} I(\vec{d_i}) < I(\vec{x}) \\ 1, & \text{otherwise.} \end{cases} \tag{4.8}$$

The decimals that corresponds to the feature vector can be computed by

$$LBP(\vec{x}) = \sum_{i=1}^{8} t_i \cdot 2^{i-1}. \tag{4.9}$$

See Fig. 4.12 for an example of the basic LBP operator.

To deal with local patterns at different scales, the LBP can be extended to a larger neighborhood size. In some research work, difference sizes are combined to extract multi-scale information of the texture and to achieve performance improvement.

Before utilizing the idea of LBP in the streak analysis, it is worth reviewing the characteristics of the streak. Due to the vertical structure of the streak, the pixel

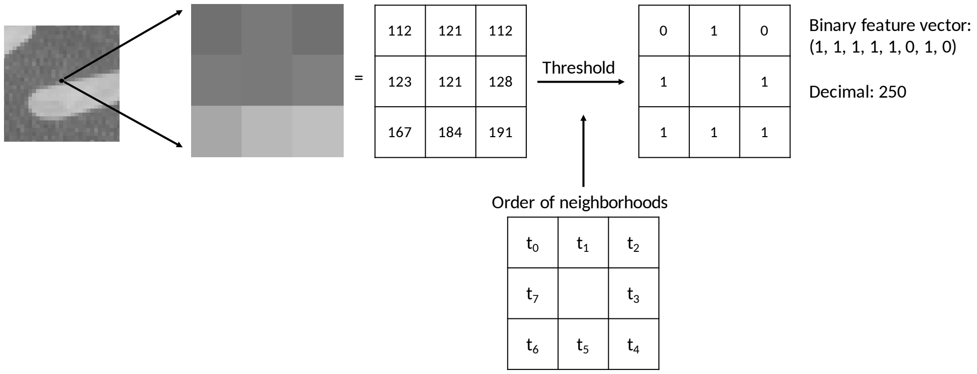


Fig. 4.12. Basic LBP operator. The LBP binary pattern is computed in a neighborhood of 3 × 3 pixels.

from a streak usually has a pixel value which is dependent on and similar to the pixels above and below, whereas different from the pixels on the left and right sides. Instead of texture analysis, we are searching for a pattern with a specific structure, where the vertical pixel difference is dominant. Hence, the order of the neighborhood pixels is rearranged such that it gives more weights to the pixels on the left and right. The order of the modified LBP is shown in Figure 4.13

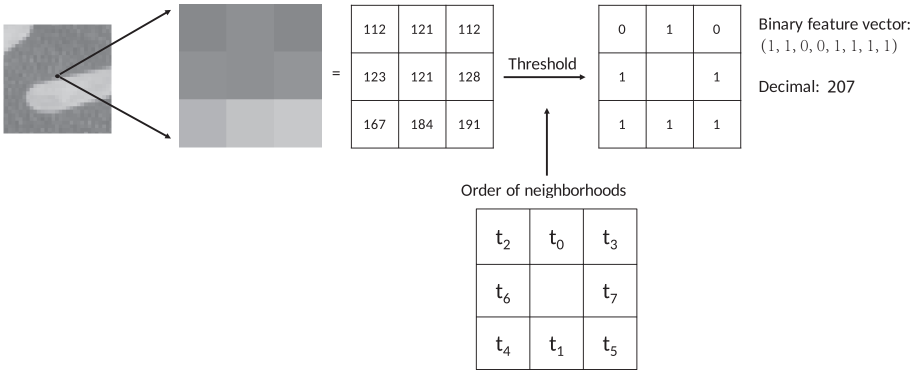


Fig. 4.13. The order of pattern in the modified LBP operator

4.4.3 Other features

To increase the discrimination ability, we developed 6 other features as follows.

Signal-to-background ratio

The signal-to-background ratio (SBR) is extracted from the matched filtering output as follows:

$$SBR = \frac{peak\ signal\ power}{sum\ entire\ power} = \frac{f_s^2}{\sum(f_E^2)} \tag{4.10}$$

where f_s is the peak value of the output, and f_E represents the entire window.

Slope of the matched filtering output

To compute this feature, we look for the left and right edges around the peak value in the matched filtering output. The edge points are defined by the element that falls in the range [0, 0.1]. Denote the edge points as x_L and x_R. Denote the peak location as X_P, the slope can be express as:

$$slope = (\frac{g(x_R) - g(x_P)}{x_P + 1} + \frac{g(x_P) - g(x_L)}{x_P - x_L + 1})/2 \tag{4.11}$$

Row profile peak location derivative

We apply the ALPD to the raw row profile, instead of the matched filtering output.

Row profile peak value derivative

We apply the same technique used for ALPD, but compute the derivative for the peak value instead of the peak location.

Row profile local maximum flag

To avoid the false alarm evoked by edges, we detect if the maximum value is a local peak in the row profile. This feature can be expressed as:

$$flag = \begin{cases} 0, & \text{if the maximum value is a local peak} \\ 1, & \text{otherwise} \end{cases} \tag{4.12}$$

4.5 Ground Truth

A ground truth labeling tool is written using MATLAB [24] Graphical User Interface (GUI). The labeling tool records the horizontal and vertical locations of the streaks. Fig. 4.14 shows the labeling interface. Because the streaks are sometimes hard to perceive, the GUI tool provides 'zoom in' option. To zoom in, one should click on the numbers that are arranged spatially corresponding to the image on the left. To record the peak location, one needs to click on the starting and ending points of the streak. A dialog will show up displaying the locations one just clicked on. If the 'Record' is pressed, the GUI tool will read the position and write to the file automatically. To make sure that the selected peak location is correct, 'show annotated streak' displays marks on the streak right away based on the annotation one just made, as shown in Fig. 4.15. 129 images are annotated and the streaks have a total length of 150245 pixels.

4.6 Evaluation

Classifiers are trained using SVM from the extracted features to predict the existence of streak. The predictor is applied to each row within the detection window for the test samples. Each row is labeled as defective or not. As the streak spans several pixels in width, we consider the detection and the annotation matches as long as the detection window contains the actual streak. To test the performance of the prediction model, false alarm rate and miss rate are obtained.

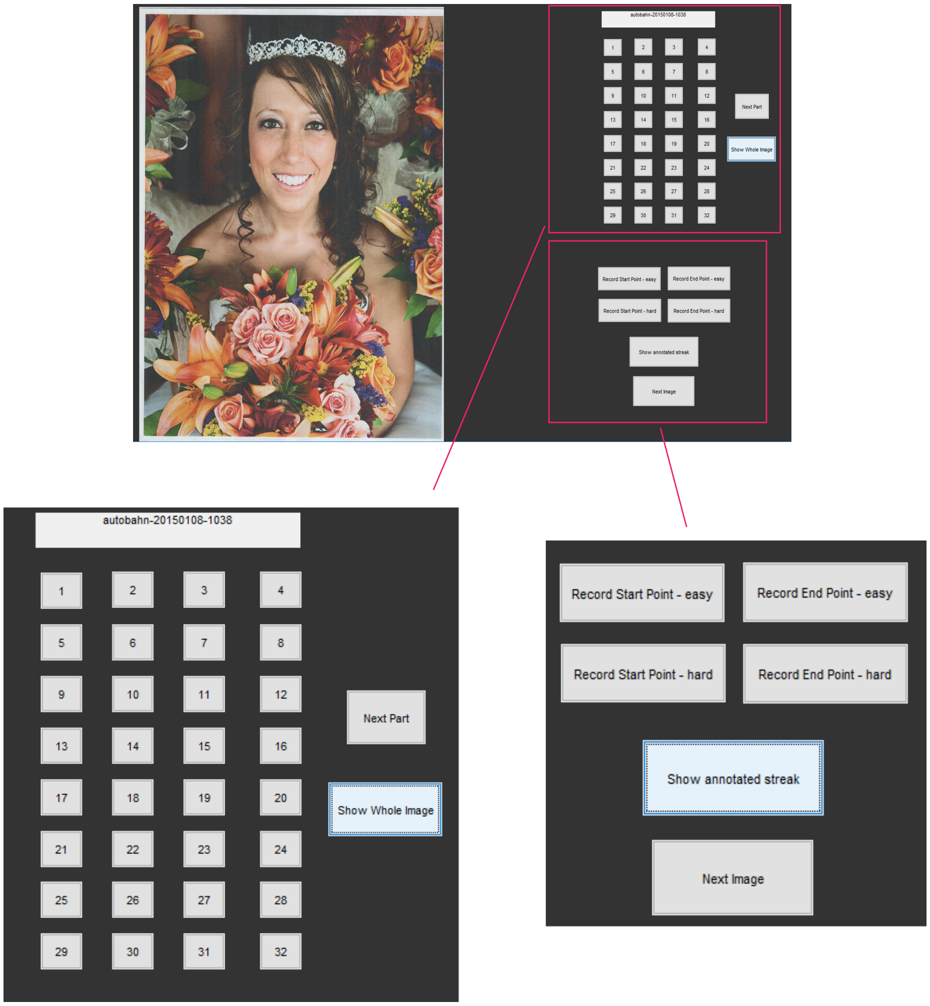


Fig. 4.14. GUI of the ground truth labeling tool.

4.7 Experiment

4.7.1 Dataset

40 pages with a large variety of contents are scanned at 300 dots per inch (dpi). Fig. 4.16 shows some examples of the scanned pages. The top half of the images are annotated. There are in total 247 streaks with 15 725 defective rows. Negative samples are selected from the non-defective regions consisting of 391825 rows.

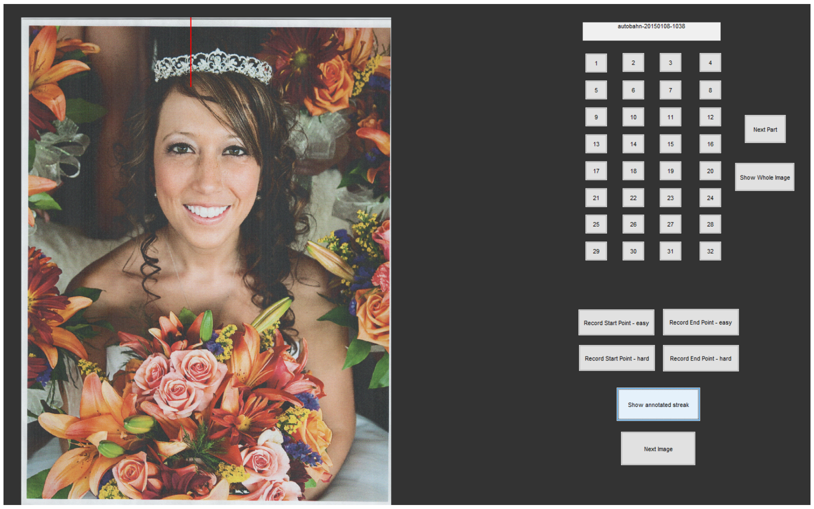


Fig. 4.15. A preview of the annotated streak is provided by clicking on 'Show annotated streak'. In case it looks wrong, the annotation can be discarded and redone.

4.7.2 Result

Table 4.1 summarizes the results of the experiment.

Table 4.1.
Summary of the streak detection result.

Mean accuracy	93.06 %
Miss rate	18.68 %
False alarm rate	6.51%

FUNCTIONAL & PARAFUNCTIONAL

Functional testing is system-level testing that looks at the program as a collection of functions. A "function" might be an individual feature or a broader capability that relies on several underlying features.

We analyze a function in terms of the inputs we can provide it and the outputs we would expect, given those inputs.

(See W.E. Howden, *Functional Program Testing & Analysis*, 1987)

Black Box Software Testing Copyright © 2000-10 Cem Kaner & James Bach 43

Fig. 4.16. Examples of the scanned pages from the dataset.

5. HIDDEN MARKOV MODEL BASED STREAK DETECTION

5.1 Introduction

The streak detection framework proposed in Chapter 4 adopts a framework where each row in the detection window is considered as one sample. Each row sample acts independently both in the model training and in the streak detection. However, this row-based streak detection framework could not employ one important characteristic of the dust-generated streak: row-dependency. Along a single streak, the streak defect at each row is a precise evidence of the existence of dust on the sensor and strongly suggests that the same dust, if not removed, could further cause streak defect at the following rows. This characteristic relies on the fact that the page is scanned from top to bottom. On the other hand, one certain row with no streak, although cannot serve as an absolute evidence, is an indication that the dust does not exist (or does not cause any streak) and the following rows are less likely to suffer from the streak defect. This characteristic represents the dependency between the nearby rows which has a very similar scenario to the Hidden Markov Model (HMM). In this chapter, we adopt the HMM model in streak detection. Experiments demonstrate that the performance improves regarding both detection accuracy and detected streak length.

5.2 Related Works

HMM is a well-known technique to incorporate the temporal context information into classification problems. It has been widely used in image processing, text analysis [25], as well as defect detection. HMM is employed to recognize unusual subsequences in wire ropes where the Viterbi scores are used as indicators for defects [26]. In [27], the HMM is utilized as a texture unit descriptor to predict the pixel values of the texture.

5.3 Preliminary

5.3.1 HMM

HMM is useful when modeling the discrete state of a system as it evolves in time. The Markov chains represent data as a sequence of states in time, where each state is and only is dependent on the previous one. At the same time, observations made at each temporal location are considered as dependent on the corresponding hidden state. As a result, the HMM incorporates the temporal context information as well as the observations in the decision making. Let $\{L_m\}_{m=0}^{M}$ denotes the discrete-time random process taking its state values on a countable set Ω. It forms a Markov chain if

$$P\{L_m = s|L_n \ for \ all \ n < m\} = P\{L_m = s|L_{m-1}\}, \tag{5.1}$$

where $s \in \Omega$ and $0 < m < M$. Note that the state space Ω is discrete and countable.

The distribution of the states at each temporal location is represented by the marginal probability $\pi_s^{(m)}$ as

$$\pi_s^m = P\{X_m = s\}. \tag{5.2}$$

To describe the dependency of the states between the adjacent rows with states $s, s' \in \Omega$, the transition probability $P_{s,s'}^{(m)}$ is defined as

$$P_{s,s'}^{(m)} = P\{X_m = s|X_{m-1} = s'\}. \tag{5.3}$$

Assuming that the marginal probability and the transient probability are both homogeneous, we have

$$\pi_s^{(m)} = \pi_s, \ P_{s,s'}^{(m)} = P_{s,s'}. \tag{5.4}$$

Based on Equation 5.1, the probability of a particular state sequence for $\{X_m\}_{m=0}^{M}$ is

$$p(x) = \pi_{x_0}^{(0)} \prod_{m=1}^{M} P_{x_{m-1},x_m}, \tag{5.5}$$

where x_m represents the state at m.

Practically the states X_m are typically hidden, while observations Y_m depending on the corresponding states X_m can be made at each temporal location. X_m and Y_m are two parallel stochastic processes with the dependency relationships illustrated in Fig 5.1 The observations are controlled by the states, for which the emission probability is defined as

$$p(y|s) = p(Y = y|L = s). \tag{5.6}$$

The joint density function for the double sequence is given by

$$p(y, l|\theta) = \pi_{l_0} \prod_{m=1}^{M} \{f(y_m|l_m)P_{l_{m-1},l_m}\}, \tag{5.7}$$

where $\theta = [\pi_s, P_{s,s'} :\ for\ s, s' \in \Omega]$

5.4 Problem Formulation

The detection window with M rows is modeled as a Markov chain $\{X_i\}_{i=0}^{M}$, where i is the row index and M is the height of the window. The state space $\Omega = \{0, 1\}$ represents the existence of streak defect, with 1 being non-defective and 2 being defective.

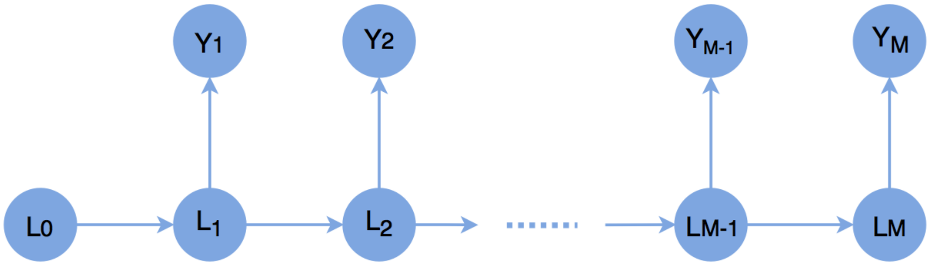


Fig. 5.1. $\{X_m\}$ represent the states and $\{Y_m\}$ represent the observations (features). The arrows explains the dependency relationships.

In our work, the transition probability $P^{(i)}_{s,s'}$ is assumed to be independent on time, i.e., the Markov Chain is homogeneous. While the transient behavior of the Markov chain ensures the influence of the initial state at $i = 0$ on the subsequent states, the homogeneousness, given sufficient time, will nevertheless lead the Markov chain to converge to a steady-state after the transient behavior is consumed. This matches with our problem for the following reasons: 1) the dust might be displaced or brought in at any time during the scanning process, which means that both transition probabilities $P\{X_i = 1|X_i - 1 = 2\}$ and $P\{X_i = 2|X_i - 1 = 1\}$ have no dependency on the vertical location along the detection window. 2) the probability of dust existence is very low, as most pages with 2500 pixels in width contain less than five streaks. Once the dust is displaced, it is less likely that dust will show up again at the same position. The state of the rows will eventually converge.

The features we developed in the last chapter are quantized before being used as the observations. To include both features in the HMM, a multi-observation HMM is developed based on the assumption that the two features are independent. The feature space can then be expressed as

$$\Omega^Y = \{s|s \in Z,\ 1 \leq s \leq S\} \tag{5.8}$$

The joint density function for the sequence $\{Y_i\}$ and $\{X_i\}$ is given by

$$p(x, y) = \pi_{x_0} \prod_{i=1}^{M} \tag{5.9}$$

5.5 Parameter Estimation

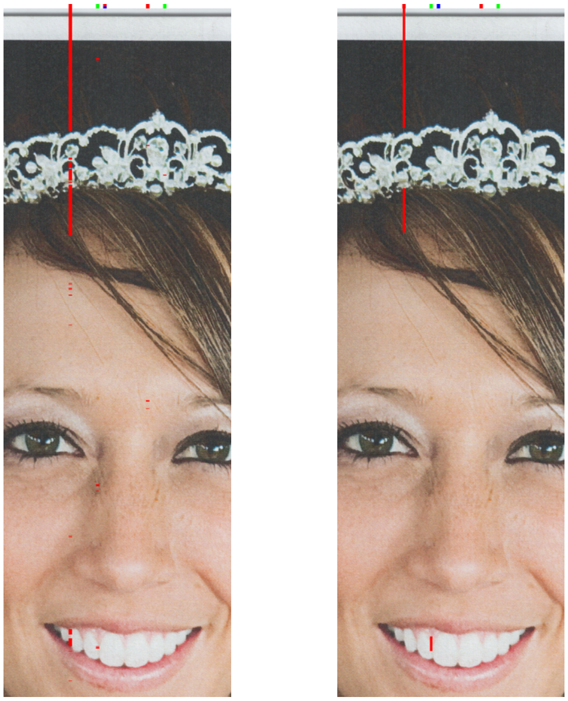
As can be seen from Equation 5.5, the states of Markov chain only depends on the marginal probability and the transition probability. While this brings simplicity to the Markov chain modeling, it demands an accurate estimation of the two parameters. We try two methods for parameter estimation: obtaining the sample probability from data and EM algorithm.

Table 5.1.
Summary of the streak detection result.

Statistics	SVM	HMM
Mean accuracy	93.06%	95.85 %
Miss rate	18.68 %	12.77 %
False alarm rate	6.51%	3.90 %

5.6 Experiment

Table 5.1 compares the method using SVM and HMM. Note that SVM uses seven features and HMM uses two features only. Two improvements can be observed. First, the detection accuracy increased by 2.8%. Second, the average detected streak length increased. Fig. 5.2 provides an example of the detection by SVM and HMM. It can be seen that, while both methods recognized the actual streaks, SVM returns some falsely detected streaks with very short length and HMM does not. HMM treats each row as dependent on the previous ones. When there is no streak found previously, HMM will assign a low probability to the following rows to be defective despite the feature values.



(a) SVM (b) HMM

Fig. 5.2. Comparison of the detection results using SVM and HMM. It can be seen that HMM rejects the detections with too short length.

6. PEDESTRIAN DETECTION USING PIXEL DIFFERENCE MATRIX PROJECTION

6.1 Introduction

Pedestrian detection has drawn much attention over the past few decades due to its wide applications in video surveillance, driving assistance, and self-driving systems. The pedestrian size within an image is an important factor for the discrimination performance of a detector. Most of the pedestrian datasets contain pedestrians with median heights ranging from 50 to 100 pixels [28] [29] [30]. Pedestrians with heights under 30 pixels are considered as low resolution ones and difficult to be detected. For the application of driving assistance, detecting medium or larger size pedestrians (50 pixels or more than 50 pixels high) leaves sufficient time for response. However, for the outdoor video surveillance, the pedestrians are usually small and with low resolution (30 pixels or less than 30 pixels high) [31] [32], which causes performance degradation when the pedestrian detectors [30] [33] [28] are directly applied.

There has been extensive research on developing a high-performance pedestrian detection system for use in the real-world applications. A majority of the research on pedestrian detection has been focused on designing discriminative features. Particularly, traditional descriptors in object detection are utilized in pedestrian detection tasks directly, in combination, or in a modified way.

Notable examples include the Haar features [29], the Local Binary Pattern (LBP) [33], and the Principle Component Analysis (PCA) [34]. A significant improvement in human detection performance was brought by the introduction of the histogram of oriented gradients (HOG) [30], establishing a baseline for future works. While the HOG remains among the most popular low-level descriptors, its variants have been widely explored [35] and methods combining HOG with other features have been proposed to extend the feature “space" [33] [36]. More recently, the integral channel features (ChnFtrs) detector [37] exploited a detection framework where features were first pooled from different low-level channel descriptors and then fed to a boosting classifier. The ChnFtrs significantly improves the pedestrian detection performance and has attracted lots of efforts focusing on extending the method, such as trying different channel features and feature pooling techniques [38]. Among these top performing detectors, the HOG remains to serve as the most widely used low-level channel features [39].

The success of HOG features arises from its capability to capture the contour information of the pedestrians. In fact, contour information has proved to be highly discriminative for classifying the human image from the non-human image [40]. The contour is usually captured by constructing gradient vectors. However, the pixel-wise calculation of $arctan$ increases the computational complexity. At the same time, for the low-resolution pedestrian detection, the presence of background interference and the degradation of contour information decrease the reliability of the gradient vectors on characterizing the edges. In order to address issues like this, we aim to design a set of light-weight features which is simple and fast to calculate and at the same time, incorporates effective shape information to achieve high detection accuracy.

We built our feature based on local pixel differences. Pixel differences have been used extensively for object recognition and texture characterization tasks. The local binary pattern (LBP) descriptor is based on the sign of the difference between the central pixel value and its neighborhood pixel values. In the Haar wavelet, the convolution of the wavelet functions and the image returns the differences between the adjacent pixel intensities. The gradient vector in the original HOG feature is calculated from the pixel differences along two orthogonal directions. In our method, we design a set of pixel difference patterns at different orientations and scales. This results in a group of pixel difference maps which contain intrinsic global relationships. We formulate the computation of the pixel difference maps as matrix manipulations of the original image and a set of pre-designed projecting matrices. With this formulation, the calculation of channel maps is no more than a series of matrix multiplications and summations. We then group the pixel difference channel maps into cell maps via average-pooling to eliminate the influence of local noises. A block normalization is applied to the cell maps to introduce tolerance to the illumination changes.

Recently, the Convolutional Neural Network (CNN) has found its success in pattern recognition area.The proposed DMP method can be viewed as one layer of the CNN. The details of the analogy are provided in Section 6.4.5. In contrast to the CNN which has a requirement on the data size, the DMP can be trained with a reasonably small amount of data.

Our contributions can be summarized as follows. Firstly, we design a set of pixel difference patterns to characterize the local structures along different orientations. An average-pooling is used to extract the cell-based local shape. We also explore the influence of cell size on the detection performance. Secondly, we propose a formulation to calculate the pixel difference using closed-form matrix products. As a result, each local pixel or region difference pattern can be expressed by projecting the input image with pre-calculated projection matrices. This simplifies the computation and increases the flexibility in further analysis and modification of the algorithm.

6.2 Related Works

Efforts have been made in exploring features that best fit the purpose of image-based pedestrian detection. Shape, texture, and motion are among the most common characteristics to be extracted to discriminate the human body from other objects. These characteristics have either been used separately or in combination in the literature for pedestrian detection.

A lot of research has been focused on describing the local shape of the human object. Among these approaches, the gradients, gradient histograms, and texture are often found to serve as low-level features, upon which more complicated features are constructed using pooling or learning-based methods. In [41], “shapelets" features are learned from multi-orientated local gradients using Adaboost, where the gradients are essentially the pixel differences within a local neighborhood. The Histograms of Gradients (HOG) [30] are constructed in a local rectangular region where the edge orientation magnitudes are voted into different histogram bins corresponding to the quantized orientation angles. In [33], the HOG was combined with a particularly designed Local Binary Pattern (LBP) to combine complementary information in order to improve the performance. Extensions of features such as wavelet and PCA are also utilized for pedestrian detection. A set of overcomplete Haar wavelet feature is used for human detection in [29] [42]. In [34], the bi-directional PCA extracts features by reducing the dimension in the row and column directions separately in a pedestrian image.

Table 6.1 lists the existing methods in the literature with the features captured, the classifier used, the datasets adopted, and some remarks. Our method belongs to the approach that builds upon gradients in the representation of pixel differences.

6.3 Preliminary

In this section, we briefly introduce the HOG features following the original design by Dalal and Triggs [30] as a baseline for immediate reference. We also provide background knowledge for the Least Squares Estimation (LSE) method which is adopted for our detector learning.

6.3.1 Histogram of Gradient (HOG)

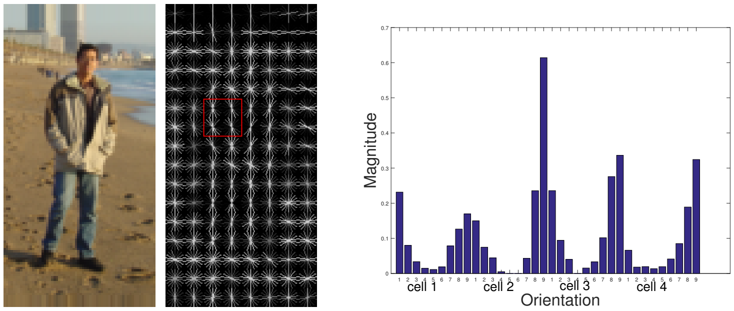
The HOG descriptors were originally proposed by Dalal and Triggs [30]. A feature vector is extracted from the detection window by concatenating local block-based histogram of gradients. The set of features is then sent to a linear SVM for pedestrian and non-pedestrian classification. Fig. 6.1(a) displays a sample positive detection window of 128×64 pixels from the INRIA dataset [30]. The detection window is divided into non-overlapping cells, each with a size of $k \times k$ pixels. Within each cell,

Table 6.1.
A brief summary of selected approaches to pedestrian detection.

Approaches	Features	Classifier	Datasets	Remarks
Shapelet [41]	Shape, Gradients	AdaBoost	INRIA	Requires high computational time to learn the shapelet feature
HOG [30]	Shape, Gradient histogram	SVM	INRIA	The most popular feature for pedestrian detection; Requires orientation binning
LBP [33]	Texture	SVM	INRIA	Requires decimal coding and histogram binning
Wavelet [29]	Texture	SVM	MIT	Needs to determine which wavelet scales to use
Bi-directional PCA [34]	Eigen-descriptors	SVM	INRIA	Lacks local information. Details may be smoothed out by PCA dimension reduction
Proposed DMP	Pixel differences	LSE	INRIA, NICTA, Daimler-CB	Pixel difference features obtained by image projection. No learning is needed to obtain the features

the gradient orientations over all pixels are grouped by a histogram with an angular range from 0 to 180 degrees with N bins. The number of bins is an adjustable parameter, which is frequently set as 9, meaning that each bin covers a 20 degrees range. The height of each bin is voted by the corresponding gradient magnitudes. The histogram of orientations provides a sparse representation of the distribution of the local gradients, while maintaining its robustness to local fluctuation. The influence of the cell size on the performance was studied in [30], where it was shown that $6 \sim 8$ pixels gave the best result on the INRIA dataset. Furthermore, to introduce invariance to illumination and contrast, the histograms are concatenated and normalized within blocks formed by a small group of cells. Typically, the blocks are overlapped so that each cell contributes to different blocks and is normalized respectively.

Fig. 6.1(b) displays an illustration of the HOG descriptor applied to the sample in Fig. 6.1(a). The direction of the strikes represents the gradient orientations and the lightness of the strikes reflects the gradient magnitudes. Fig. 6.1(c) plots the HOG feature for the block consisting of four neighborhood cells marked in red in Fig. 6.1(b). The block-based feature is a concatenation of the normalized HOG features of the four cells.



(a) (b) (c)

Fig. 6.1. Left: A test sample containing pedestrian. Middle: HOG descriptor. Right: Concatenated HOG features of a block containing 2 × 2 cells.

6.3.2 Linear least squares regression

Linear least squares regression is a widely used method which estimates parameters by minimizing the squared discrepancies between the observed data and the target values. Given a vectorized observation $\boldsymbol{x} \in \mathbb{R}^d$, the data can be fit with a function of the following form:

$$f(\boldsymbol{x}, \boldsymbol{\alpha}) = \sum_{k=0}^{K-1} \alpha_k p_k(\boldsymbol{x}), \tag{6.1}$$

where $\boldsymbol{\alpha} = [\alpha_0, \alpha_1, ..., \alpha_{K-1}]^T$, and $p_k(\boldsymbol{x})$ constitutes the basis expansions of $\boldsymbol{x}$ [43].

For supervised binary classification problem, each training observation comes with a label $y \in \{0, 1\}$. With m pairs of observations and labels $(\boldsymbol{x}_i, y_i), i = 1, 2, ..., m$, the parameter vector $\boldsymbol{\alpha}$ by minimizing the sum of the squared errors:

$$E(\boldsymbol{\alpha}) = \sum_{i=1}^{m} (y_i - f(\boldsymbol{x}_i, \boldsymbol{\alpha}))^2 = ||\mathbf{y} - \mathbf{P}\boldsymbol{\alpha}||_2^2, \tag{6.2}$$

where $\mathbf{y} = [y_1, y_2, ..., y_m]^T$ and $\mathbf{P} = [p_0(\boldsymbol{x}), p_2(\boldsymbol{x}), ..., p_{K-1}(\boldsymbol{x})]$ with a dimension of $m \times K$. A more stable version is formulated by adding a weight decay regularization term [44] to the criterion function:

$$E(\boldsymbol{\alpha}) = \frac{1}{2}||\mathbf{y} - \mathbf{P}\boldsymbol{\alpha}||_2^2 + \frac{b}{2}||\boldsymbol{\alpha}||_2^2, \tag{6.3}$$

where b is set to a small value to control the weighting of the regularization factor in the second term. We can then solve for $\boldsymbol{\alpha}$ that minimizes E by

$$\boldsymbol{\alpha} = (\mathbf{P}^T\mathbf{P} + b\mathbf{I})^{-1}\mathbf{P}^T\mathbf{y} \tag{6.4}$$

where $\mathbf{I}$ is an identity matrix.

6.4 Difference Matrix Projection (DMP)

In this section, we propose an efficient feature construction method for pedestrian detection. Firstly, a set of pixel difference patterns are designed to describe the multi-orientation characteristics of the pedestrian. Then an average-pooling is applied to local rectangular regions in order to extract the oriented pixel difference information. Finally, a block-wise normalization is adopted to address the problem of illumination variation.

Pixel difference computation is a frequent operation when extracting features that characterize the shape and texture of objects. We formulate the computation of local pixel differences into a global image projection using pre-designed projection matrix. With the linear transformation of the image, the pixel difference computation is conducted by a limited number of global matrix multiplications in closed-form, which introduces flexibilities in designing the pixel difference patterns and simplifies the computation.

An overview of our feature extraction pipeline is illustrated in Fig. 6.2, and the details are given in the following sub-sections.

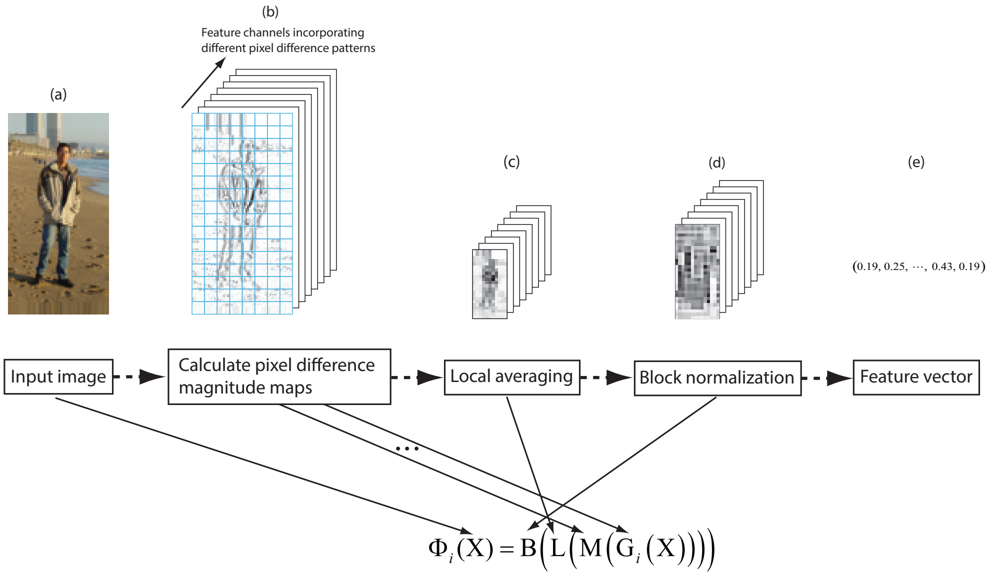


Fig. 6.2. An overview of the proposed DMP feature extraction process.

6.4.1 Pixel difference map computation

Our features are built based on pixel differences. Before presenting the features, we first introduce our formulation of pixel differences calculation in the matrix form for the purpose of simplifying the computation and generalizing the algorithm. Consider an input image $\mathbf{X} \in \mathbb{R}^{M \times N}$. A shifting matrix $\mathbf{H}_l$ is defined to shift the elements of the image to the right by l columns via matrix multiplication:

$$\mathbf{X}_l^h = \mathbf{X}\mathbf{H}_l, \tag{6.5}$$

where $\mathbf{X}_l^h$ is the shifted image, and h denotes that the shifting is along the horizontal direction. The shifting matrix is constructed as follows:

$$\mathbf{H}_l = \begin{pmatrix} 0 & \mathbf{I} \\ \mathbf{I} & 0 \end{pmatrix}_{N \times N} \begin{array}{l} \} \;\; N-l \\ \} \;\; l \end{array}, \tag{6.6}$$

where $\mathbf{I}$ is the identity matrix, N is the horizontal dimension of $\mathbf{X}$, and l is the shifting distance in the unit of number of columns. With the shifting operation, we can obtain the pixel difference map along the horizontal direction $\mathbf{D}_l^h$ by

$$\begin{aligned} \mathbf{D}_l^h &= \mathbf{X}(\mathbf{I} - \mathbf{H}_l) \\ &= \mathbf{X}\mathbf{P}_l^h, \end{aligned} \tag{6.7}$$

where $\mathbf{P}_l^h$ is the projection matrix constructed from Equation 6.6

$$\mathbf{P}_l^h = \mathbf{I} - \mathbf{H}_l = \begin{pmatrix} \mathbf{I} & -\mathbf{I} \\ -\mathbf{I} & \mathbf{I} \end{pmatrix}_{N \times N} \begin{array}{l} \} \;\; N-l \\ \} \;\; l \end{array}. \tag{6.8}$$

Each element of $\mathbf{D}_l^h$ is the difference between two pixels separated by l columns in $\mathbf{X}$. When $l = 1$, the difference is between two horizontally adjacent pixels. The value of l can be adjusted when constructing the projection matrix in Equation 6.8, resulting in pixel differences at various scales.

Similarly, we construct a shifting matrix $\mathbf{V}_l$ to shift the image upward by l rows as follows:

$$\mathbf{V}_l = \begin{pmatrix} 0 & \mathbf{I} \\ \mathbf{I} & 0 \end{pmatrix}_{M \times M} \begin{array}{l} \} \;\; M-l \\ \} \;\; l \end{array}, \tag{6.9}$$

where M is the vertical dimension of the original image $\mathbf{X}$, and l denotes the number of rows to shift by. Note that $\mathbf{V}_l$ has a similar form to $\mathbf{H}_l$ except the dimension of the matrix. This is because a pre-multiplication is needed to shift the image vertically. We can then derive the equation for the pixel differences along the vertical direction $\mathbf{D}_l^v$ as

$$\mathbf{D}_l^v = \mathbf{P}_l^v \mathbf{X}, \tag{6.10}$$

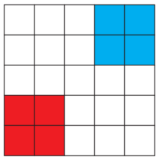


Fig. 6.3. Pixel difference pattern involving pixel groups within a neighborhood of 5 × 5 pixels.

with the projection matrix

$$\mathbf{P}_l^v = \mathbf{I} - \mathbf{V}_l = \begin{pmatrix} \mathbf{I} & -\mathbf{I} \\ -\mathbf{I} & \mathbf{I} \end{pmatrix}_{M \times M} \begin{matrix} \} & M-l \\ \} & l \end{matrix}, \tag{6.11}$$

where **I** is the identity matrix.

Note that we can always interpret a shifting by two rows (or columns) as two runs of shifting by one row (or column), which means that the shifting matrices at different scales satisfy the following relationsho

$$\begin{aligned} \mathbf{H}_l &= (\mathbf{H}_1)^l \\ \mathbf{V}_l &= (\mathbf{V}_1)^l. \end{aligned} \tag{6.12}$$

If a more complicated pixel difference map is desired, such as the pattern illustrated in Fig. 6.3, we can formulate the calculation with combined usages of different shifting matrices

$$\mathbf{D} = \mathbf{X} + \mathbf{X}\mathbf{H}_1 + \mathbf{V}_1\mathbf{X} + \mathbf{V}_1\mathbf{X}\mathbf{H}_1 - \mathbf{V}_3\mathbf{X}\mathbf{H}_3 - \mathbf{V}_3\mathbf{X}\mathbf{H}_4 - \mathbf{V}_4\mathbf{X}\mathbf{H}_3 - \mathbf{V}_4\mathbf{X}\mathbf{H}_4. \tag{6.13}$$

With the help of Equation 6.12, the above computation can be simplified as follows

$$\begin{aligned} \mathbf{D} &= \mathbf{X}(\mathbf{I} + \mathbf{H}_1) + \mathbf{V}_1\mathbf{X}(\mathbf{I} + \mathbf{H}_1) - \mathbf{V}_3\mathbf{X}\mathbf{H}_3(\mathbf{I} + \mathbf{H}_1) - \mathbf{V}_4\mathbf{X}\mathbf{H}_3(\mathbf{I} + \mathbf{H}_1) \\ &= (\mathbf{I} + \mathbf{V}_1)\mathbf{X}(\mathbf{I} + \mathbf{H}_1) - (\mathbf{I} + \mathbf{V}_1)\mathbf{V}_3\mathbf{X}\mathbf{H}_3(\mathbf{I} + \mathbf{H}_1). \end{aligned} \tag{6.14}$$

By defining a set of intermediate projection matrices based on the shifting matrices

$$\begin{aligned} \mathbf{L}_1 &= (\mathbf{I} + \mathbf{V}_1) \\ \mathbf{R}_1 &= (\mathbf{I} + \mathbf{H}_1) \\ \mathbf{L}_2 &= (\mathbf{I} + \mathbf{V}_1)\mathbf{V}_3 = (\mathbf{I} + \mathbf{V}_1)(\mathbf{V}_1)^3 \\ \mathbf{R}_2 &= \mathbf{H}_3(\mathbf{I} + \mathbf{H}_1) = (\mathbf{H}_1)^3(\mathbf{I} + \mathbf{H}_1), \end{aligned} \tag{6.15}$$

we can represent the pixel difference pattern in Fig. 6.3 as

$$\mathbf{D} = \mathbf{L}_1\mathbf{X}\mathbf{R}_1 - \mathbf{L}_2\mathbf{X}\mathbf{R}_2, \tag{6.16}$$

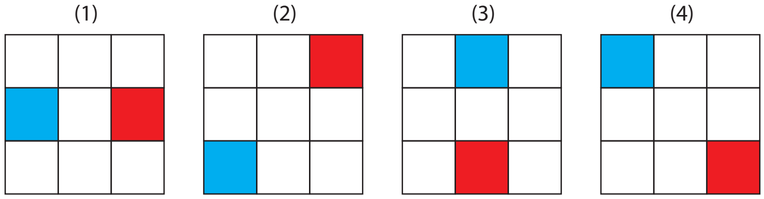
where the projecting matrices $\mathbf{L}_1$, $\mathbf{R}_1$, $\mathbf{L}_2$, and $\mathbf{R}_2$ can be pre-calculated and readily used to obtain the pixel difference maps.

6.4.2 Local pixel difference descriptor

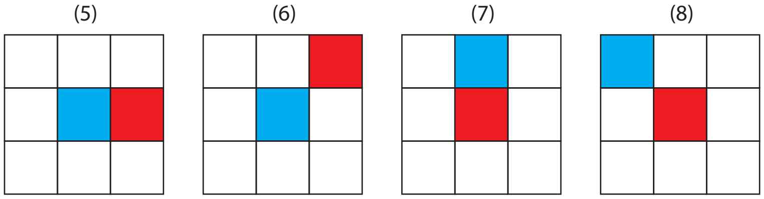
Constructing local features within a small-sized neighborhood and pooling the extracted local features to form a global feature vector is a widely adopted method for image recognition tasks. The local structures are critical to describe the characteristics of an object. As for the pedestrian detection problem, the shape and contour of the person are among the most discriminative factors to build an effective descriptor. A well-designed feature should be able to emphasize the local contours, and at the same time, exclude the noise and fluctuation.

We construct the local pixel difference descriptors that characterize the local shape and texture within a neighborhood of $k \times k$ pixels. The local neighborhood is rectangular-based due to the simplicity of computation. By adjusting the value of k and the pixel groups involved in the difference calculation, we can construct a multi-scale and multi-orientation local descriptor. Fig. 6.4 gives some examples of pixel difference patterns. The upper set in Fig. 6.4(a) is carried out in a neighborhood of 3×3 pixels, and the difference is between pixels with a distance of two pixels. The four pairs of pixel differences depict the gradient information along 4 directions: $0°$, $45°$, $90°$, and $135°$. The lower set in Fig. 6.4(b) consists of the differences between pixels with a distance of one pixel. With the two pixel difference pattern sets, the gradient information with fine and coarser details are extracted.

We call each pixel difference pattern a feature channel. Each channel represents pixel difference information along a different orientation at a difference scale. For convenience, we index each difference pattern by a channel index i. Due to the variance in color and illuminations of pedestrian clothes and the background, the



(a) Pixel difference pattern set 1 within 3×3 pixel neighborhood.



(b) Pixel difference pattern set 2 within 3×3 pixel neighborhood.

Fig. 6.4. Pixel difference patterns at different scales. The pixel differences are taken between the cyan cell and the red cell. The blank cells are not involved in the calculation. Only the magnitudes of the differences will be used to construct the features.

sign of pixel differences is of no importance in discriminating persons. Hence, only the magnitude of the pixels differences are used in constructing features.

By utilizing the image shifting operation introduced in section 6.4.1 and using $\mathbf{G}_i$ to denote the linear operation of matrix projection, we are able to calculate each pixel difference channel map with matrix multiplications as shown in Equation 6.17. Note that the projection matrices are designed for the specific difference patterns, and can be readily applied to the image to obtain the desired pixel difference maps:

$$
\begin{aligned}
\mathbf{G}_1(\mathbf{x}) &= \mathbf{X}(\mathbf{I} - \mathbf{H}_1^2) \\
\mathbf{G}_2(\mathbf{x}) &= \mathbf{X} - \mathbf{V}_1^2\mathbf{X}\mathbf{H}_1^2 \\
\mathbf{G}_3(\mathbf{x}) &= \mathbf{X}(\mathbf{I} - \mathbf{V}_1^2) \\
\mathbf{G}_4(\mathbf{x}) &= \mathbf{X}\mathbf{H}_1^2 - \mathbf{V}_1^2\mathbf{X} \\
\mathbf{G}_5(\mathbf{x}) &= \mathbf{X}(\mathbf{I} - \mathbf{H}_1) \\
\mathbf{G}_6(\mathbf{x}) &= \mathbf{X} - \mathbf{V}_1\mathbf{X}\mathbf{H}_1 \\
\mathbf{G}_7(\mathbf{x}) &= \mathbf{X}(\mathbf{I} - \mathbf{V}_1) \\
\mathbf{G}_8(\mathbf{x}) &= \mathbf{X}\mathbf{H}_1 - \mathbf{V}_1\mathbf{X}
\end{aligned}
\tag{6.17}
$$

Using $\mathbf{M}$ to denote the pixel-wise operation of obtaining the absolute value, the pixel difference magnitudes can be expressed as

$$\mathbf{\Phi}_i^{diff} = \mathbf{M}(\mathbf{G}_i(\mathbf{X})). \tag{6.18}$$

With the flexibility that both the local neighborhood size and the pixel groups are adjustable, it is possible to extract various features and different combinations using a limited number of global matrix multiplications. Fig. 6.4 illustrates the possible pixel difference patterns which are adopted for pedestrian detection in this paper.

6.4.3 Cell-based local descriptors

The raw pixel difference magnitudes are very sensitive to local noises where they are ineffective for pedestrian detection if used directly. To increase the robustness and reduce the feature dimension, we take the average of the differences within a cell of $p \times p$ pixels along each feature channel respectively. As a result, we obtain the pixel differences statistics in local regions.

The above local regional averaging operation can again, be computed using matrix manipulation. We construct two projection matrices to sum the elements within the local regions along the horizontal and the vertical directions respectively. For the horizontal direction, the projection matrix is

$$A_L = \begin{pmatrix} I_{1\times p} & & 0 \\ & \ddots & \\ 0 & & I_{1\times p} \end{pmatrix},$$

where p is the cell size, and M is the horizontal size of the pixel difference map. For the vertical direction, the projection matrix has the form

$$A_R = \begin{pmatrix} I_{p\times 1} & & 0 \\ & \ddots & \\ 0 & & I_{p\times 1} \end{pmatrix}$$

where N is the vertical size of the pixel difference map. With the two projection matrices, local averaging of the pixel difference magnitude map $\mathbf{\Phi}_i^{diff}$ of channel i can be expressed by a single equation as

$$\mathbf{\Phi}_i^{ave} = A_L \mathbf{\Phi}_i^{diff} A_R, \tag{6.19}$$

which is another global matrix multiplication operation.

6.4.4 Block normalization

Local variations in illumination and contrast might cause the feature values to vary significantly. To eliminate these effects, we perform a local contrast normalization, where the cells are grouped into blocks for a block-wise normalization. In our experiment, the block consists of 2 × 2 cells with an overlapping of one cell. The features of the four cells within each block are then concatenated into a single feature vector and normalized. By overlapping the blocks by one cell, most cells in the image appear multiple times and are normalized with respect to different sets of cells. The final feature descriptor of the detection window is formed by concatenating all the normalized block features. For a detection window of 128 × 64 pixels, the final feature vector has a dimension of 3360.

The feature extraction workflow is summarized in Fig. 6.2. The operations $\mathbf{G}_i$ and $\mathbf{L}$ are linear operations which can be carried out by simple analytic matrix multiplications as explained above. Compared to the derivation of HOG in Fig. 6.5, we avoid calculating the gradient magnitudes and angles in a pixel-wise manner. Instead, the direction information is incorporated by calculating the pixel differences along multi-orientations. In this way, the oriented differences are represented by the different feature channels, where the elements of the feature map in each channel are the pixel difference magnitudes. Next, because the feature channels are separately calculated, we can carry out an average-pooling to extract the local statistics of oriented pixel difference magnitudes. The local statistics within each cell describes the overall intensities of pixel difference magnitudes along different orientations. This prevents the construction of a histogram and largely simplifies the computational cost. On the other hand, the feature extraction is accomplished by a series of linear and non-linear transformations on the original image, where the linear transformations are expressed by matrix multiplications with pre-calculated projection matrices.

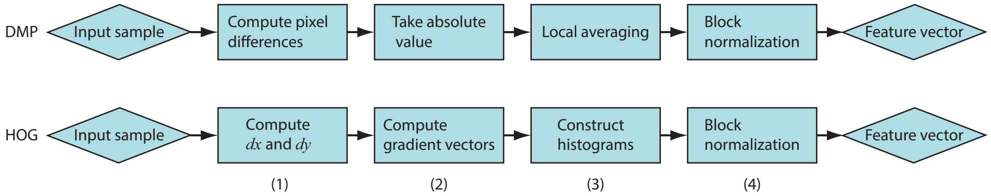


Fig. 6.5. Pipeline of the DMP and the HOG feature extraction algorithm.

6.4.5 The connection between our method and the CNN model

In this section, we show that the proposed DMP method is analogous to a single layer of the CNN structure which consists of the convolution, activation, and pooling stages. Fig. 6.6 illustrates the proposed DMP method from the perspective of CNN. In DMP, the computational intensive convolution operation is replaced by a set of simple sparse matrix multiplication operations. Each multiplication corresponds to one convolution along a certain orientation and at a certain scale. Then, the magnitude operation for each pixel difference can be considered as applying an activation function $f(x) = \text{abs}(x)$ to the convolved results. Next, a cell-based aggregation is applied which is similar to the average pooling in the CNN. Finally, a block normalization is added to the CNN layer followed by a classification step. The proposed DMP method is different from the shallow CNN in that the filter banks are pre-defined instead of trained from back propagation. In fact, the choices of pixel difference patterns are flexible by varying the scales and involved pixels to cater for different image resolutions. An effective set of pixel difference patterns can be determined by cross validations on specific problems. For the pedestrian detection in this paper, we choose the pixel difference patterns in 6.4 within a 3×3 neighborhood to suit the small-scale pedestrians.

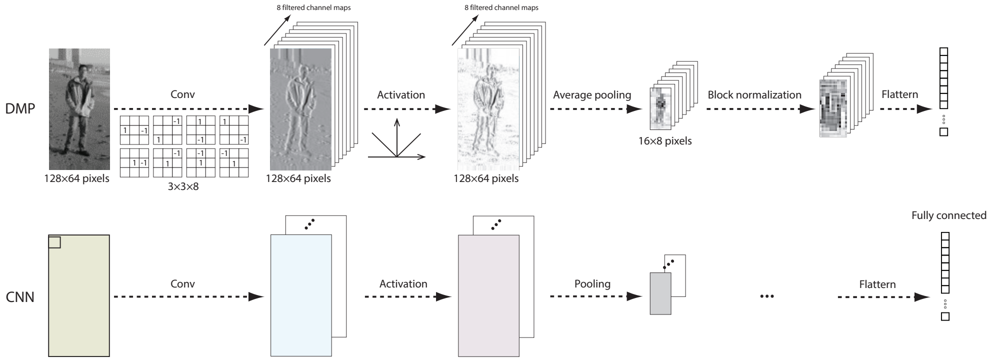


Fig. 6.6. Understanding DMP from a different point of view. Upper: the DMP algorithm; Lower: one-layer CNN model

6.4.6 Training and testing

The pedestrian detection is accomplished by classifying each sample into two categories: one that contains pedestrian(s) (positive sample) and one that does not (negative sample). The samples are cropped detection windows from photos and images. The process of pedestrian classification is illustrated in Fig. 6.7. Suppose there are n training samples $\mathbf{X} \in \mathbb{R}^{n \times m}$ labeled by $\mathbf{y} = \{y_1, ..., y_n\}$, we compute the DMP features $\mathbf{F}^{Tr} \in \mathbb{R}^{n \times d}$, where d is the DMP feature dimension. The training features are sent to the linear LSE to learn the weight vector $\boldsymbol{\alpha}$ by

$$\boldsymbol{\alpha} = (\mathbf{P}^T\mathbf{P} + b\mathbf{I})^{-1}\mathbf{P}^T\mathbf{y} \tag{6.20}$$

where $\mathbf{P} = [\mathbf{F}^{Tr}, [1, ..., 1]^T]$ and b is the regularization constant parameter. In the prediction stage, the class labels are obtained by multiplying the learned weight vector to the testing DMP features.

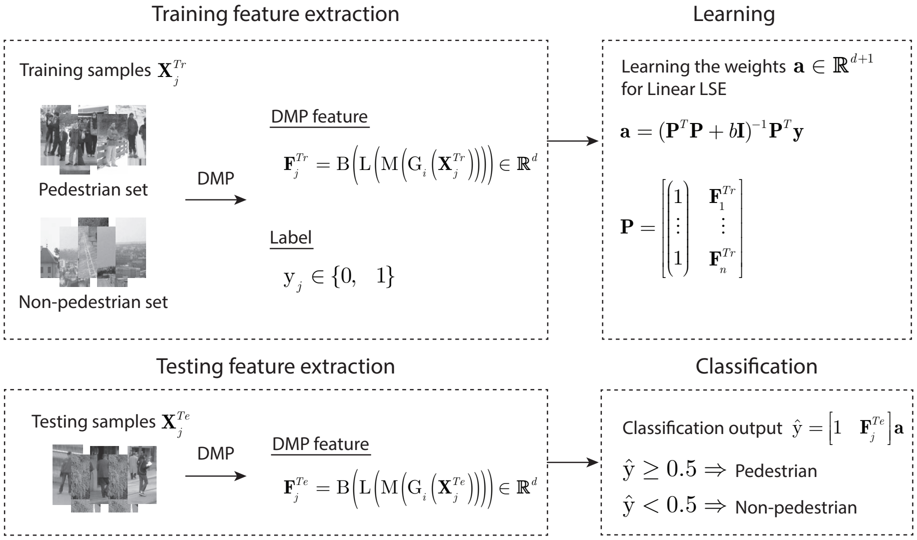

Training feature extraction
Training samples $\mathbf{X}_j^{Tr}$
Pedestrian set
Non-pedestrian set
DMP
DMP feature
$\mathbf{F}_j^{Tr} = \mathrm{B}\Big(\mathrm{L}\Big(\mathrm{M}\Big(\mathrm{G}_i\left(\mathbf{X}_j^{Tr}\right)\Big)\Big)\Big) \in \mathbb{R}^d$
Label
$\mathrm{y}_j \in \{0, \quad 1\}$
Learning
Learning the weights $\mathbf{a} \in \mathbb{R}^{d+1}$ for Linear LSE
$\mathbf{a} = (\mathbf{P}^T\mathbf{P} + b\mathbf{I})^{-1}\mathbf{P}^T\mathbf{y}$
$\mathbf{P} = \begin{bmatrix} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} & \begin{matrix} \mathbf{F}_1^{Tr} \\ \vdots \\ \mathbf{F}_n^{Tr} \end{matrix} \end{bmatrix}$
Testing feature extraction
Testing samples $\mathbf{X}_j^{Te}$
DMP
DMP feature
$\mathbf{F}_j^{Te} = \mathrm{B}\Big(\mathrm{L}\Big(\mathrm{M}\Big(\mathrm{G}_i\left(\mathbf{X}_j^{Te}\right)\Big)\Big)\Big) \in \mathbb{R}^d$
Classification
Classification output $\hat{\mathrm{y}} = \begin{bmatrix} 1 & \mathbf{F}_j^{Te} \end{bmatrix}\mathbf{a}$
$\hat{\mathrm{y}} \geq 0.5 \Rightarrow$ Pedestrian
$\hat{\mathrm{y}} < 0.5 \Rightarrow$ Non-pedestrian

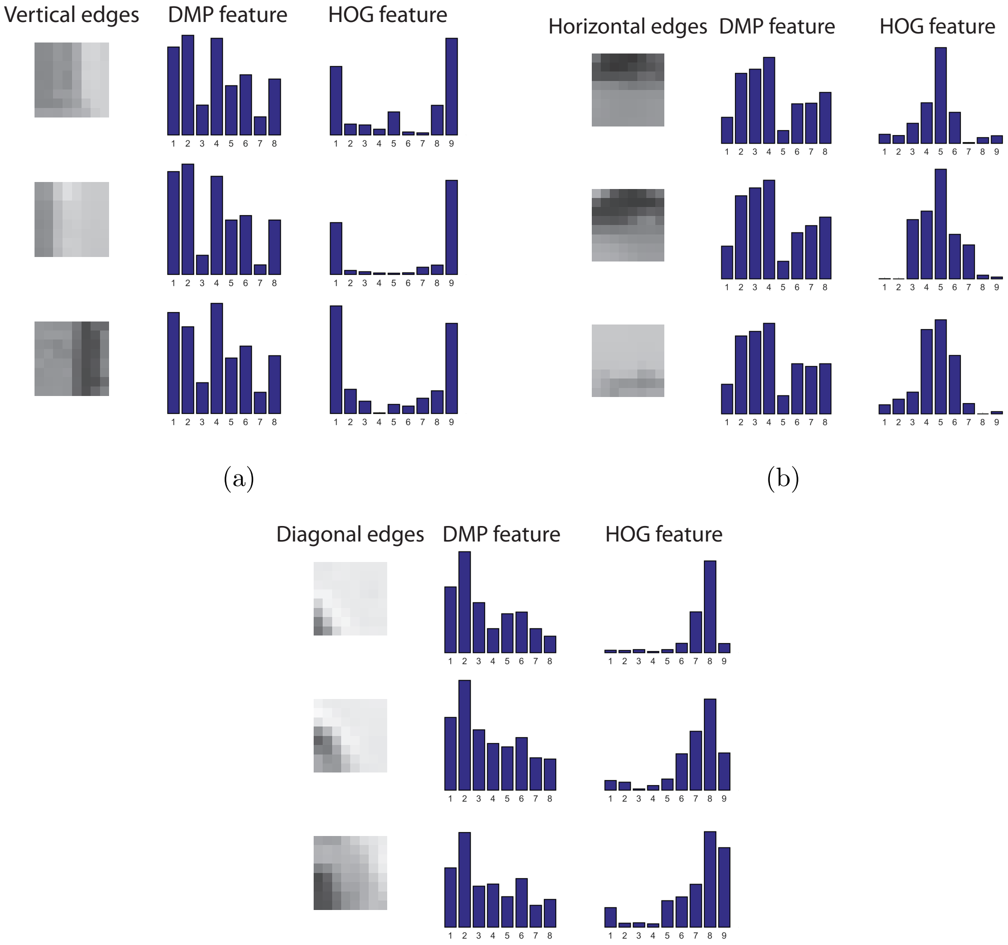
Fig. 6.7. DMP-based pedestrian classification process.

6.4.7 Feature analysis

Our proposed set of pixel difference patterns describes the texture at each pixel in the cell, while average-pooling summarizes the pixel differences into a feature vector which introduces tolerance to noise. For the cell-based features, a cell is the smallest area from which the feature vectors are extracted. Hence, it is the smallest unit to describe the structure of the human object. In this section, we provide two case studies based on 8×8 pixels image cells to illustrate the underlying mechanism of our feature extraction algorithm.

a) Case study 1: The feature vector extracted from the local cell should have the ability to discriminate local shapes. We compare the extracted DMP features for image cells containing various shapes in Fig. 6.8. The image patches are extracted from real pedestrian images, hence they all contain different levels of noise. The HOG feature vectors are also provided for comparison. The DMP and HOG features are represented by the bin plots in Fig. 6.8. The DMP feature vector has 8 bins corresponding to 8 channels of pixel differences, while the HOG feature vector has 9 bins corresponding to the 9 quantized orientations. We observe that the DMP feature vectors have similar shape for cells containing the same edge type, but also possess distinguishing shapes for different edges. The discrimination power of the DMP features is comparable to that of the HOG features.

a) Case study 2: We also examine the tolerance to noise by applying our feature on image patches with and without local noises in Fig. 6.9. In Fig. 6.9(a), some artificial noise pixels are added to the original image patch, while in Fig. 6.9(b) the noise pixels are manually removed from the original image patch. We can see that, with and without the presence of noises, both DMP and HOG maintained the relative intensities among different bins in the feature vector.



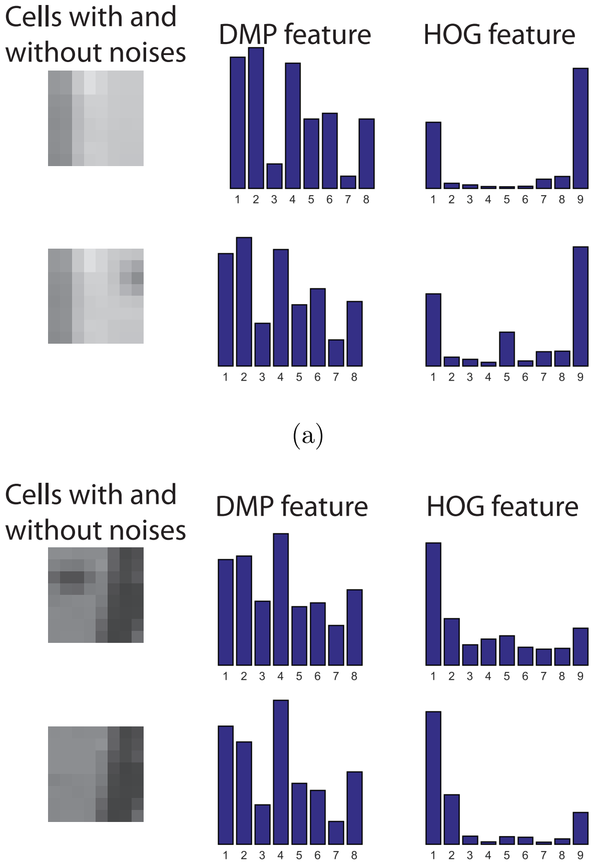
(a) (b)

(c)

Fig. 6.8. An illustrative example of DMP feature vector describing different shapes. The HOG features are provided for comparison. (a) DMP and HOG feature vectors for a 8×8 pixels cell with vertical edges. (b) DMP and HOG feature vectors for a 8×8 pixels cell with horizontal edges. (c) DMP and HOG feature vectors for a 8×8 pixels cell with diagonal edges.

6.5 Experiments

We first observe the influence of the parameter setting in our method in Subsection 6.5.1. In Subsection 6.5.2, we evaluate our generic features for pedestrian detection using three state-of-the-art datasets: the INRIA person dataset [30], the Daimler Chrysler pedestrian classification benchmark dataset [45], and the NICTA pedestrian



(a)

(b)

Fig. 6.9. An illustrative example of noise tolerance. (a) DMP and HOG feature vectors for a 8×8 pixels cell containing vertical edge (upper) and the same cell with artificial noise added (lower). (b) DMP and HOG feature vectors for a 8×8 pixels cell containing vertical edges with noise (upper) and the same cell with noise removed (lower).

dataset [46]. These three datasets are among the most used pedestrian datasets to evaluate the discrimination ability of classifiers to identify pedestrians from non-pedestrian samples. We also provide a comparison of the detection performance between our method and some widely used feature extraction methods. Finally, we conduct a Friedman test to observe whether there is statistical significance among the performances of the compared detectors in Subsection 6.5.3.

To evaluate our method and compare it with other methods, we use the Detection-Error-Trade-Off (DET) curve which plots the miss rates over the false positive rates per window (FPPW) on a log-log scale. FPPW is defined as the number of false positives divided by the number of all negative detection windows. It is equivalent to false positive rate (FPR) as each detection window is one test sample in our experiment. Another commonly adopted performance evaluation method is the false positive per image (FPPI), which is obtained by counting the misclassified detection windows per image. In [40], it is argued that both FPPW and FPPI have their own advantages and distinguishing focuses. In FPPI evaluation, the pedestrian classification is conducted on a sliding window which scans the image exhaustively. This scenario is believed to cover all the cases in the real pedestrian detection task, which FPPW is not able to cover. However, FPPW focuses on comparing features and classifiers. It excludes other factors that might affect the final performance evaluation, such as Non-maximal suppression (NMS) used in FPPI. Because our purpose is to evaluate the discrimination ability of the feature to classify pedestrians, the FPPW evaluation method is adopted throughout this paper.

6.5.1 Effect of cell size setting

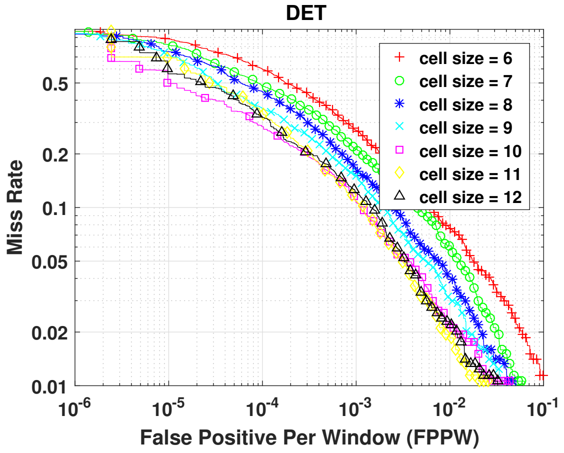
We present the observation of our detector with different parameter settings using the INRIA and the Daimler-CB datasets. The cell size p in our method controls the area of the pedestrian body part within which the pixel differences are averaged, as shown in Equation 6.19. If the cell size is too big, useful edge structure information may be smoothed out in the resulting features. On the other hand, if the cell size is too small, it may fail to remove the local noise effectively through the process of averaging. As one main factor that affects the optimal cell size p is the pedestrian size, we experiment on two datasets: the INRIA dataset with relatively large pedestrian size (pedestrian sample size: 128×64 pixels) and the Daimler-CB dataset with small pedestrian size (pedestrian sample size: 36×18 pixels).

For the INRIA dataset, the performance is evaluated for cell sizes ranging from 6 × 6 to 12 × 12 pixels at a step of one pixel. From Fig. 6.10(a), we see that a cell size of 10 × 10 produces the best result. In fact, the cell sizes between 10 × 10 and 12 × 12 pixels all give similar detection performance, while sizes smaller than 8 × 8 pixels lead to obvious performance degradations. The Daimler-CB dataset consists of relatively small pedestrian samples, so it is tested on cell sizes ranging from 2 × 2 to 7 × 7 at a step of one pixel. Fig. 6.10(b) shows that the optimal cell size is 3 × 3 pixels which is, as expected, significantly smaller than the one for INRIA dataset. At the same time, the performance degrades as the cell size gets larger. The conclusion is that the optimal cell size in our detector depends on the pedestrian size.

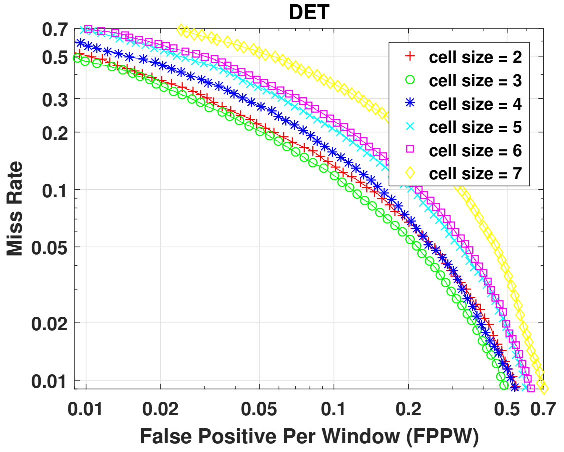
6.5.2 Performance comparison

We compare the performance of our feature to that of a list of state-of-the-art features: histogram of gradients (HOG) [30], local binary pattern (LBP) [33], wavelets [29], and principle component analysis (PCA). The HOG is capable of providing excellent performance on pedestrian detection and has become the most widely adopted features for pedestrian detection for the last decades. While the HOG depicts the contour information of the object, the LBP is a texture descriptor widely used in texture analysis and face recognition [47]. The Haar Wavelet efficiently identifies the local-oriented pattern at different scales. Its variants have been widely explored by researchers for pedestrian detection purpose [29, 48, 49]. The PCA is a powerful and popular dimension reduction technique in object detection. In the following, we shall describe the evaluation settings for each method in detail.

For the HOG, we follow the setup indicated by Dalal et al in [30], except for the parameter of cell size. In the original paper, it was shown that 6 ∼ 8 pixels was the optimal cell size for the INRIA dataset. To provide a fair comparison between the performances of HOG and our method, we determine the cell size using cross-validation on the training data for each dataset, as what we do for our DMP features.



(a) INRIA dataset.



(b) Daimler-CB dataset.

Fig. 6.10. Evaluation of different cell sizes. (a) INRIA dataset. (b) Daimler-CB dataset.

For the LBP feature, we adopt the cell-structured LBP proposed by Wang in [33], which outperformed the traditional HOG detector on the INRIA dataset. We follow the parameter settings on LBP computation, but set the cell size as an adjustable parameter to be determined by cross-validation for a fair comparison.

For the PCA, the eigenspace dimension is also determined by cross-validation for all the datasets. We choose to tune the above parameters instead of directly adopting the setting provided by the original paper because different dataset contains samples of different resolutions, which might affect the optimal parameter settings. It is necessary and fair to find the best parameter setting for each method on different databases.

For the Haar wavelet, we use the overcomplete sets of Haar wavelet features that were first utilized on pedestrian detection in [29]. As illustrated in the literature, wavelets of very fine scales represent the noise and the wavelet filters at very coarse scales are as large as the object. Thus, we only extract features from the two medium scales of wavelets. For example, for the image size of 36 × 18 for the Daimler-CB dataset, we choose the wavelets at scales of 4 × 4 pixels and 8 × 8 pixels for each orientation. Besides, the sign of intensity difference, which contains the information of the color contrast between the pedestrian and the background, is irrelevant to human detection. So we only encode the magnitude of the coefficients in the overcomplete Haar features.

Table 6.2 provides a summary of algorithms and the parameter settings as a result of cross-validation on the training set. The details of each dataset and the experiment results are presented in the following subsections.

Table 6.2.
Summary of datasets, methods, and parameter settings.

Database	size	HOG	DMP	LBP	PCA	Wavelets
		cell size	cell size	cell size	dim.	# levels
Daimler-CB	36 × 18	2	3	4	160	2
INRIA	128 × 64	10	10	18	160	2
NICTA 1	20 × 8	2	1	2	140	2
NICTA 2	20 × 16	2	1	2	160	2

Daimler Chrysler pedestrian classification benchmark dataset

The Daimler Chrysler pedestrian classification benchmark dataset [45] consists of five disjoint sets: three for training and two for testing. Each set contains 4800 pedestrian examples (from 800 identical pedestrians with reflections and small shifts) and 5000 non-pedestrian examples. All samples are scaled into a fixed size of 18 × 36 pixels with a border of two pixels. The small size of the pedestrian samples features the observation of pedestrians at a far distance from the car in real driving assistance systems. The non-pedestrians samples were selected by matching a pedestrian's shape template in pedestrian-free images. The carefully designed negative set, together with the small size of the pedestrian samples, makes the pedestrian detection a challenging task on the Daimler-CB dataset.

We adopt the same experiment configuration as that in [45]. A three-fold cross-validation over the three training sets is conducted to determine the optimal settings for the parameters. To evaluate the test performance, three classifiers are trained, each using two out of the three training set and tested on the two test sets separately. The final detection performance is obtained by averaging the six classification results.

The performance is shown in the DET curves in Fig. 6.11. Our detector outperforms all the compared ones at both low and high FPPW rates. At 0.01 FPPW, our detector has a lower miss rate than the HOG with a margin of 16%, and than LBP by 14.93%.

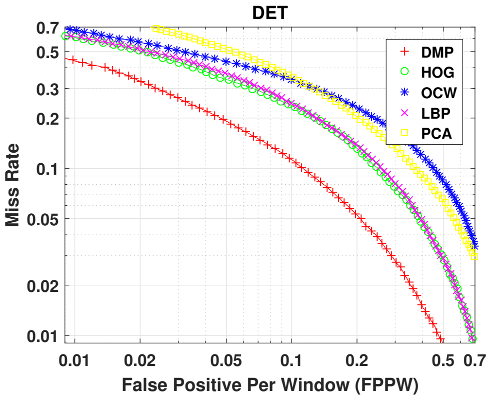


Fig. 6.11. Performance comparison on the Daimler-CB dataset.

INRIA dataset

The INRIA dataset was proposed in [30] with a goal of creating a dataset that includes more variations in pose, clothing, illumination, background, and occlusions. The training set contains 2474 positive samples and 1218 pedestrian-free images, and the test set includes 1132 positive samples and 453 pedestrian-free images. The size of the human sample is 128 × 64 pixels, which is quite high resolution for pedestrian images. We randomly crop ten 128 × 64 windows from each of the 1218 pedestrian-free training images to create a negative training set with 12180 samples.

The classification results are shown in Fig. 6.12. We can see that our detector has a comparable performance to that of HOG and LBP, and outperforms other detectors.The HOG and LBP show the best performance at low FPPW and low miss rate regions respectively. Compared to the HOG feature, our method has a lower miss rate at high FPPW, but a higher miss rate at low FPPW.

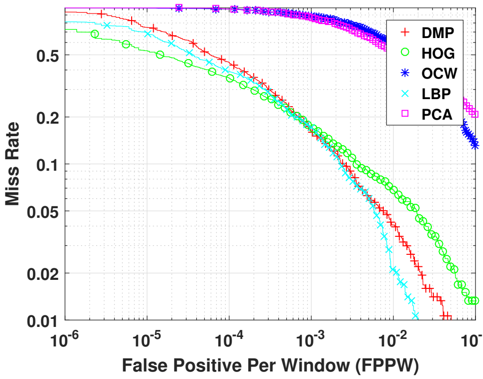


Fig. 6.12. The performances of selected detectors on the INRIA dataset.

NICTA pedestrian dataset

The NICTA pedestrian dataset contains positive and negative samples at different resolutions ($20 \times 8, 20 \times 16, 40 \times 16, 40 \times 32, 80 \times 64$ pixels) to allow researchers to test their algorithms on a range of scales and height-to-width ratios. In this work, we evaluate our method on the two lowest resolutions: 20 × 8 pixels and 20 × 16 pixels. Small sized samples represent pedestrians at a far distance, which has been the hard case in the pedestrian detection task. We choose these two datasets in order to examine our detection performance at low resolutions.

Fig. 6.13(a) and Fig. 6.13(b) display the results for NICTA datasets at the resolutions of 20 × 8 and 20 × 16, respectively. The superiority of DMP on low-resolution pedestrian samples can be observed in both of the DET curves. The HOG shows poor performances on both DET curves. The reason may be that the HOG requires a high resolution to depict the contour of the pedestrian. With the presence of high amount of noises, the HOG failed to pick up the information effectively.

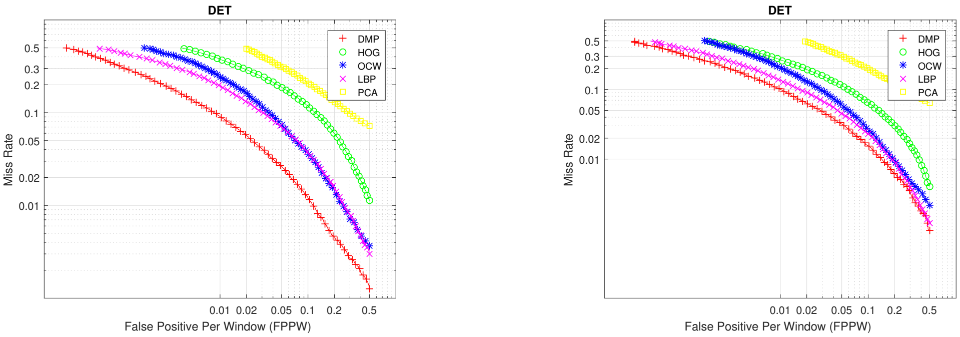


Fig. 6.13. The performances of selected detectors on the NICTA dataset. (a) Pedestrian sample size of 20 × 8 pixels. (b) Pedestrian sample size of 20 × 16 pixels.

6.5.3 Friedman test

We compare the area under the curve (AUC) of all the tested detectors by conducting the Friedman test [50] with a post-hoc analysis. Since the Friedman test rejects the null hypothesis that all the examined algorithms have an equal performance ranking, the post-hoc Nemenyi analysis is plotted in Fig. 6.14 to analyze the difference among the compared algorithms. It can be seen that the DMP ranks the best among all detectors, while there is no statistically significant difference between

DMP, HOG, LBP, and OCW. The PCA is seen to have significantly lower accuracy than that of DMP, HOD, LBP, and OCW.

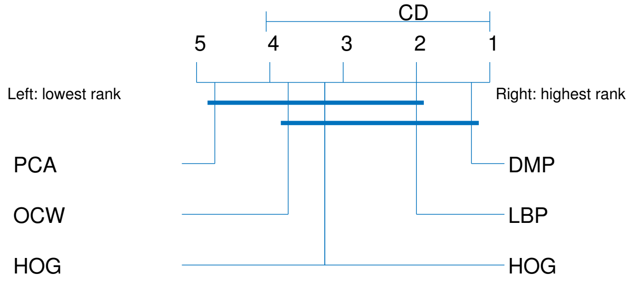


Fig. 6.14. Friedman test analyzing the critical difference in performances. The x-axis shows the rank, and the blue bar link detectors for which there is no sufficient evidence to declare a statistically significant difference in their performances.

6.5.4 CPU processing time performance

We evaluate the CPU processing time for extracting the features, training a model, and testing on 100 test samples with DMP, HOG, and LBP features. The CPU time is recorded using a PC of 3.4 GHz CPU with 16 GByte RAM running MATLAB 2016a [51].

The CPU time for HOG is measured based on the implementation following [30] using MATLAB. No MATLAB built-in function was used for extracting DMP, HOG, and LBP features. We follow the parameter settings in Tabel 6.2. It can be seen from Table 6.3 that the DMP takes less time than HOG in both feature extraction time and learning time. The DMP takes less feature extraction time due to its simple feature construction scheme and matrix multiplication form of operation. For the INRIA dataset, the cell size for the LBP feature is 16, which is the largest among all features. Hence its feature computation time is less than the DMP and HOG features. Although LBP histogram feature has 59 bins [47], the large cell size leads to a smaller feature vector compared to the DMP and HOG features. As a result, it requires the least learning and classification time. The HOG feature has a larger dimension than that of DMP, hence it requires more learning and classification time.

Table 6.3.
A comparison of CPU time in seconds on the INRIA dataset.

Approaches	Time for feature extraction	Time for learning	Total time	Time for testing 100 test samples
DMP	461.5469	56.8438	518.3906	2.4219
HOG	536.9063	83.6875	620.5938	3.6719
LBP	421.9688	9.9219	431.8907	2.7657

6.5.5 Summary of results and discussion

The observations and analysis are summarized as follows:

- The proposed DMP features have shown superiority on low-resolution datasets, i.e., pedestrian samples at far scales (pedestrian height <30 pixels) [28]. The performance of DMP is comparable to that of HOG and LBP for pedestrian samples at near scales (pedestrian height >80 pixels). The DMP method computes the local pixel differences within a local neighborhood emphasizing on capturing the high-frequency local contour information of the human object. Moreover, the pixel difference patterns in DMP are computed along four orientations instead of across finer orientation steps. This brings tolerance to the shape change of human objects due to clothes and movement. Low-resolution pedestrians are expected to possess less detailed information, hence capturing the overall contour information with relatively high invariance to noises is favored in the low-resolution pedestrian detection task.

- The computation of DMP is formulated via matrix multiplications and the DMP feature has a compact size. The CPU processing time shows that the efficiency is comparable to that of HOG. The LBP has the best CPU performance among all examined features due to its small feature dimension.

- The far scale pedestrian detection is a challenging task in the pedestrian detection research [28]. In driving assistance, detecting pedestrian at a far distance evokes an early reminder, leaving sufficient time for the driver to react. Also, low-resolution pedestrians are often found in video surveillances [52]. Our DMP feature can find its application in such practices.

7. CONCLUSION

An extensive study on feature extraction for several applications in the image processing area was performed in this thesis. The contributions are summarized below:

- A set of wavelet-based features considering multi-scale information was proposed to evaluate the macro-uniformity of printed pages.
- A figure of merit based on the psychophysical experiment was trained to predict the perceived assessment score of the macro-uniformity for the printed pages.
- A set of test pages which consists of various structures was designed and printed to study the 2.5D printing quality.
- A comprehensive set of features was proposed to characterize the relief printing quality.
- A framework for streak detection was designed and implemented
- An efficient set of features characterizing the streaks was proposed and implemented
- A streak detector was trained based on SVM.
- A model-based streak detection was proposed and compared with the SVM-based method
- A set of pixel difference patterns was proposed to characterize the local structures along different orientations.
- We proposed a formulation to calculate the pixel difference using closed-form matrix products to simplify the computation and increases the flexibility in further analysis and modification of the algorithm.

By constructing frequency band images, we took advantage of multi-scale decomposition that Haar wavelets transform provides and manage to analyze print defect according to their specified frequency ranges. To improve the prediction precision, there is still room to try different sets of features that can take advantage of Haar wavelet filtered frequency band images, and also explore various wavelet basis functions and structural approximation methods.

We described our work to characterize two relief printers: the wet-on-wet and the wet-on-dry printers. In general, the calculated metrics showed that with some exceptions, the relief printers tested reproduce the input designs with good accuracy. However, with the wet-on-dry printer, we observed a dependence on the printing direction. For the ramp target with asymmetric shape along the printing direction, the left side was always steeper than the right side. As expected, the MTF analysis showed that there is a strong relationship between the frequency of fine details and the accuracy of the reproduction. The accuracy decreased as the frequency increases. The surface finish analysis leaded to the conclusion that the surface roughness depends on the geometry of the relief prints. As the structure of the relief prints becomes steeper, the printed surface became rougher. The light reflection model provideed a heuristic visualization of the surface characteristics for the ideal prints and the real prints under the same light environment setting. From the simulation, we observed that the visual experience depends on the direction of the incident illumination and the viewing angle. That is to say; the surface exhibited a different extent of roughness when we change the direction of the incoming light or rotate the relief prints. The above-mentioned work provideed an initial approach to characterizing relief printers to provide high-quality reproduction of surfaces with sophisticated color and 3D texture properties.

A framework for streak detection was proposed. The detection process included feature extraction and classification. Novel features were proposed to characterize the streaks. A streak detector was trained using SVM based on the features. The results showed a promising detection accuracy rate. Moreover, a model-based feature extraction framework was proposed using the Hidden Markov Model (HMM). The HMM-based method was superior to the SVM-based method because the former one considers context information of the streak.

We proposed an effective and efficient feature construction method for pedestrian detection. The features were constructed based on pixel differences from various channels, followed by an average-pooling and a block-based normalization. The pixel difference channel maps were constructed by matrix projection, allowing us to write the algorithm in an analytic form. The results on three public pedestrian databases showed that our detector maintains a comparable performance on high-resolution pedestrian detection compared to HOG and LBP features, while having a significant accuracy superiority for the low-resolution pedestrian detection.

REFERENCES

REFERENCES

[1] J. Walker, A Primer on Wavelets and Their Scientific Applications, ser. Studies in Advanced Mathematics Series. Chapman and Hall/CRC, 2008.

[2] R. C. Gonzalez and R. E. Woods, Digital Image Processing, 2nd ed. Prentice hall New Jersey, 2001.

[3] W. Wang, G. Overall, T. Riggs, R. Silveston-Keith, J. Whitney, G. Chiu, and J. P. Allebach, "Figure of merit for macro-uniformity based on image quality ruler evaluation and machine learning framework," in Proceedings of SPIE Image Quality and System Performance X, vol. 8653, 2013, p. 86530P.

[4] D. R. Rasmussen, F. Gaykema, Y. S. Ng, K. D. Donohue, W. C. Kress, and S. Zoltner, "W1.1 macro uniformity," in Proceedings of SPIE Image Quality and System Performance VI, vol. 7242, 2009, pp. 724 204–724 215.

[5] R. Pintus, T. Malzbender, O. Wang, R. Bergman, H. Nachlieli, and G. Ruckenstein, "Photo repair and 3D structure from flatbed scanners." in International Conference on Computer Vision Theory and Applications, vol. 29, 2009, pp. 40–50.

[6] T. Baar and M. V. Oritz Segovia, "Colour and texture appearance modeling of 2.5D prints." in Proceedings of the 12th Congress of the International Colour Association, 2013, pp. 8–12.

[7] W. Jang and J. Allebach, "Characterization of printer mtf," Journal of Imaging Science and Technology, vol. 50, no. 3, pp. 264–275, 2006.

[8] N. Bonnier and A. J. Lindner, "Measurement and compensation of printer modulation transfer function," Journal of Electronic Imaging, vol. 19, no. 1, pp. 011 010–011 010, 2010.

[9] M. Pedersen and J. Y. Hardeberg, "Full-reference image quality metrics: Classification and evaluation," Foundations and Trends® in Computer Graphics and Vision, vol. 7, no. 1, pp. 1–80, 2012.

[10] D. Dimitrov, W. van Wijck, K. Schreve, and N. de Beer, "Investigating the achievable accuracy of three dimensional printing," Rapid Prototyping Journal, vol. 12, pp. 42–52, 2006.

[11] M. Yonehara, T. Matsui, K. Kihara, H. Isono, A. Kijima, and T. Sugibayashi, "Evaluation method of surface texture by surface roughness based on geometrical product specications (GPS)," Materials Transactions, vol. 45, no. 4, pp. 1019–1026, 2004.

[12] I. 13660, "Information technology–office equipment–measurement of image quality attributes for hardcopy output–binary monochrome text and graphic images," International Organization for Standardization and International Electrotechnical Commission, Tech. Rep. 13660:2008(E), 2001.

[13] T. Malzbender, R. Samadani, S. Scher, A. Crume, D. Dunn, and J. Davis, "Printing reflectance functions," ACM Transactions on Graphics (TOG), vol. 31, no. 3, 2012.

[14] F. Pellacini, J. A. Ferwerda, and D. P. Greenberg, "Toward a psychophysically-based light reflection model for image synthesis," in Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, 2000, pp. 55–64.

[15] R. Takano, K. Baba, S. Inoue, K. Miyata, and N. Tsumura, "Reproduction of gloss unevenness on printed paper by reflection model with consideration of mesoscopic facet," in Color and Imaging Conference, no. 1, 2012, pp. 206–210.

[16] F. B. Leloup, M. R. Pointer, P. Dutré, and P. Hanselaer, "Overall gloss evaluation in the presence of multiple cues to surface glossiness," Journal of the Optical Society of America A, vol. 29, no. 6, pp. 1105–1114, 2012.

[17] B. T. Phong, "Illumination for computer generated pictures," Communications of the ACM, vol. 18, no. 6, pp. 311–317, 1975.

[18] H. S. Rosario, E. Saber, W. Wu, and K. Chandu, "Streak detection in mottled and noisy images," Journal of Electronic Imaging, vol. 16, no. 4, pp. 043 005–043 005, 2007.

[19] J. Liu, W. Wu, B. Price, E. Hamby, and R. Minhas, "Characterizing streaks in printed images: A matching pursuit method using wavelet decomposition," in Annual Conference of the Prognostics and Health Management Society, 2010.

[20] M. S. Erkilinc, M. Jaber, E. Saber, and R. Pearson, "Line and streak detection on polished and textured surfaces using line integrals," in Proceedings of SPIE Algorithms and Systems IX, vol. 7870, 2011, p. 787008.

[21] R. Rasmussen, E. N. Dalal, and K. Hoffman, "Measurement of macro-uniformity: Streaks, bands, mottle and chromatic variations," in PICS, 2001, pp. 90–95.

[22] J.-H. Lee and J. P. Allebach, "Inkjet printer model-based halftoning," IEEE transactions on image processing, vol. 14, no. 5, pp. 674–689, 2005.

[23] S. J. Park, M. Q. Shaw, G. Kerby, T. Nelson, D.-Y. Tzeng, K. R. Bengtson, and J. P. Allebach, "Halftone blending between smooth and detail screens to improve print quality with electrophotographic printers," IEEE Transactions on Image Processing, vol. 25, no. 2, pp. 601–614, 2016.

[24] MATLAB. The MathWorks Inc., 2017a.

[25] F. Pernkopf, "3D surface analysis using coupled HMMs," Machine Vision and Applications, vol. 16, no. 5, pp. 298–305, 2005.

[26] E.-S. Platzer, J. Nägele, K.-H. Wehking, and J. Denzler, "Hmm-based defect localization in wire ropes-a new approach to unusual subsequence recognition," in Joint Pattern Recognition Symposium, 2009, pp. 442–451.

[27] H. Hadizadeh and S. B. Shokouhi, "Random texture defect detection using 1-D hidden markov models based on local binary patterns," IEICE transactions on information and systems, vol. 91, no. 7, pp. 1937–1945, 2008.

[28] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," IEEE transactions on pattern analysis and machine intelligence, vol. 34, no. 4, pp. 74–761, 2012.

[29] C. Papageorgiou and T. Poggio, "A trainable system for object detection," International Journal of Computer Vision, vol. 38, no. 1, pp. 15–33, 2000.

[30] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, 2005, pp. 886–893.

[31] H.-M. Hu, X. Zhang, W. Zhang, and B. Li, "Joint global–local information pedestrian detection algorithm for outdoor video surveillance," Journal of Visual Communication and Image Representation, vol. 26, pp. 168–181, 2015.

[32] H.-K. Chen, X.-G. Zhao, S.-Y. Sun, and M. Tan, "PLS-CCA heterogeneous features fusion-based low-resolution human detection method for outdoor video surveillance," International Journal of Automation and Computing, vol. 14, no. 2, pp. 136–146, 2017.

[33] X. Wang, T. X. Han, and S. Yan, "An HOG-LBP human detector with partial occlusion handling," in IEEE International Conference on Computer Vision, 2009, pp. 32–39.

[34] H. Kim, "Novel and efficient pedestrian detection using bidirectional PCA," Pattern Recognition, vol. 46, no. 8, pp. 2220–2227, 2013.

[35] S. Wu, R. Laganière, and P. Payeur, "Improving pedestrian detection with selective gradient self-similarity feature," Pattern Recognition, vol. 48, no. 8, pp. 2364–2376, 2015.

[36] S. Walk, N. Majer, K. Schindler, and B. Schiele, "New features and insights for pedestrian detection," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010, pp. 1030–1037.

[37] P. Dollár, Z. Tu, P. Perona, and S. Belongie, "Integral channel features," in Proceedings of British Machine Vision Conference, 2009.

[38] J. J. Lim, C. L. Zitnick, and P. Dollár, "Sketch tokens: A learned mid-level representation for contour and object detection," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2013, pp. 3158–3165.

[39] S. Zhang, C. Bauckhage, and A. B. Cremers, "Efficient pedestrian detection via rectangular features based on a statistical shape model," IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 2, pp. 763–775, 2015.

[40] D. T. Nguyen, W. Li, and P. O. Ogunbona, "Human detection from images and videos: a survey," Pattern Recognition, vol. 51, pp. 148–175, 2016.

[41] P. Sabzmeydani and G. Mori, "Detecting pedestrians by learning shapelet features," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.

[42] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, 2001, pp. I–I.

[43] K.-A. Toh and H.-L. Eng, "Between classification-error approximation and weighted least-squares learning," IEEE transactions on pattern analysis and machine intelligence, vol. 30, no. 4, pp. 658–669, 2008.

[44] K.-A. Toh, Q.-L. Tran, and D. Srinivasan, "Benchmarking a reduced multivariate polynomial pattern classifier," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 6, pp. 740–755, 2004.

[45] S. Munder and D. M. Gavrila, "An experimental study on pedestrian classification," IEEE transactions on pattern analysis and machine intelligence, vol. 28, no. 11, pp. 1863–1868, 2006.

[46] G. Overett, L. Petersson, N. Brewer, L. Andersson, and N. Pettersson, "A new pedestrian dataset for supervised learning," in IEEE Intelligent Vehicles Symposium, 2008, pp. 373–378.

[47] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," IEEE transactions on pattern analysis and machine intelligence, vol. 28, no. 12, pp. 2037–2041, 2006.

[48] P. Viola, M. J. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," International Journal of Computer Vision, vol. 63, no. 2, pp. 153–161, 2005.

[49] S. Zhang, C. Bauckhage, and A. B. Cremers, "Informed haar-like features improve pedestrian detection," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2014, pp. 947–954.

[50] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," Journal of Machine learning research, vol. 7, no. Jan, pp. 1–30, 2006.

[51] MATLAB. The MathWorks Inc., 2016a.

[52] X. Wang, M. Wang, and W. Li, "Scene-specific pedestrian detection for static video surveillance," IEEE transactions on pattern analysis and machine intelligence, vol. 36, no. 2, pp. 361–374, 2014.

VITA

VITA

Xing Liu received her BS in physics from University of Science and Technology of China in 2008 and her MS in physics from Purdue University in 2011. She worked as a visiting researcher at Yonsei University, Seoul, Korea focusing on pattern recognition from Nov. 2015 to Jun. 2018. Her main research interests include image processing, computer vision, and their applications in intelligence systems.