

This electronic thesis or dissertation has been downloaded from the King's Research Portal at <https://kclpure.kcl.ac.uk/portal/>



Improved Biologically-inspired Methods for Contour Detection

Wang, Qi

Awarding institution:
King's College London

The copyright of this thesis rests with the author and no quotation from it or information derived from it may be published without proper acknowledgement.

END USER LICENCE AGREEMENT



Unless another licence is stated on the immediately following page this work is licensed

under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International

licence. <https://creativecommons.org/licenses/by-nc-nd/4.0/>

You are free to copy, distribute and transmit the work

Under the following conditions:

- Attribution: You must attribute the work in the manner specified by the author (but not in any way that suggests that they endorse you or your use of the work).
- Non Commercial: You may not use this work for commercial purposes.
- No Derivative Works - You may not alter, transform, or build upon this work.

Any of these conditions can be waived if you receive permission from the author. Your fair dealings and other rights are in no way affected by the above.

Take down policy

If you believe that this document breaches copyright please contact librarypure@kcl.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.



Improved Biologically-inspired Methods for Contour Detection



Qi Wang

Department of Informatics, King's College London

This thesis is submitted for the degree of

Doctor of Philosophy

2017

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this thesis are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This thesis is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and acknowledgements. The copyright of this thesis rests with the author and no quotation from it or information derived from it may be published without the prior written consent of the author.

Qi Wang

February 2017

Abstract

Contour detection is the process of finding meaningful smooth curves within an image, such as the boundaries of objects. It is a process at which the biological visual system excels and is also a fundamental technique and crucial step for many applications in computer vision. This thesis is primarily concerned with biologically-inspired approaches to contour detection. In this thesis, some biologically-inspired contour detection algorithms were improved. Such improvement included the better performance obtained from different extended versions of an existing biologically-inspired method, simplification of a non-biologically-inspired method to a biologically-inspired method without degrading performance, and application of a biologically-inspired method to improve other biologically-inspired and non-biologically-inspired methods. All of the contour detection results mentioned in this thesis were both qualitatively and quantitatively evaluated by the BSDS300 and/or BSDS500 benchmarks using the quantitative measures: F-score and/or ODS, OIS and AP. All the parameter values of the evaluated methods were determined by operating the algorithm on training images in this benchmark and observing the alteration of F-score or ODS.

A neurophysiologically-inspired model, the PC/BC model of V1 devised to locate boundaries defined by intensity discontinuities, was first extended to also be able to locate boundaries defined by colour discontinuities. This extension was inspired by neurophysiological data from single neurons in macaque V1. The colour PC/BC model of V1 (F-score 0.67) have significantly better performance compared with the original PC/BC model of V1 (F-score 0.61) and slightly outperform some recently proposed contour detection algorithms which use more cues and/or require a complicated training procedure. The PC/BC model of V1 was also extended to simulate the second visual cortical area (V2) in an attempt to detect texture boundaries and corners. The two versions of the PC/BC model of V2 for texture boundary and corner detection were integrated with the PC/BC model of V1 to successfully locate some texture boundaries and enhance some weak contours obtained from the PC/BC model of V1. However, the integrated PC/BC model for texture boundary detection (F-score 0.60) and corner detection (F-score 0.51) both generated worse performance compared to the original PC/BC model of V1 (F-score 0.61).

A simplified and biologically-inspired version of the texture gradient method used in the probability of boundary (Pb) algorithm is also proposed in this thesis. The Pb algorithm is the first stage of a recent near state-of-the-art contour detection algorithm. The proposed texture gradient method (F-score 0.58) has similar performance in detecting texture boundaries to the Pb texture gradient method (F-score 0.58) but also reduces computation time by about 240 seconds when tested on 100 images and simulates texture boundary detection in V2 of human/animal visual system. Finally, in this thesis, a modified version of the sparse reconstruction-based discrimination (SRBD) method inspired by the neurophysiological evidence for sparse coding in the biological visual system was applied to refine the contour detection results produced by the colour PC/BC model of V1 and two leading contour detection algorithms. Previously, the SRBD method (F-score 0.66) had been applied to significantly improving the contour detection results of the Canny edge detector (F-score 0.58). Here, the modified version of the biologically-inspired SRBD is shown to improve the result derived from three previously proposed contour detection methods with at least one of the quantitative measures used in the BSDS500 benchmark increased by 0.01.

Table of Contents

DECLARATION.....	2
ABSTRACT	3
TABLE OF CONTENTS	5
TABLE OF FIGURES	8
TABLE OF TABLES	11
ACKNOWLEDGEMENTS	12
ABBREVIATIONS	13
CHAPTER 1 INTRODUCTION	15
1.1 CONTEXT	15
1.2 AIMS.....	17
1.3 LIST OF CONTRIBUTIONS	18
1.4 OUTLINE OF THESIS.....	19
CHAPTER 2 BACKGROUND AND RELATED WORKS	22
2.1 INTRODUCTION	22
2.2 LITERATURE REVIEW ON DIFFERENT NON-BIOLOGICALLY-INSPIRED METHODS FOR CONTOUR DETECTION.....	22
2.3 LITERATURE REVIEW ON DIFFERENT BIOLOGICALLY-INSPIRED METHODS FOR CONTOUR DETECTION	26
2.4 EVALUATION BENCHMARKS FOR CONTOUR DETECTION METHODS.....	31
2.4.1 The BSDS300 and BSDS500 benchmark.....	34
2.4.2 The Prague Texture Segmentation benchmark	37
2.5 THE PREDICTIVE CODING/BIASED COMPETITION MODEL OF V1	39
2.5.1 The LGN Model	39
2.5.2 The V1 Model.....	39
2.6 THE GLOBAL PROBABILITY OF BOUNDARY METHOD WITH THE OWT-UCM ALGORITHM.....	46
2.6.1 Brightness, colour and texture gradient method at multiple scales defined in the Pb algorithm.....	46
2.6.2 The Normalized Cuts algorithm and its integration with the multi-scaled Pb method	48
2.6.3 Closed contours derived from gPb using OWT-UCM	49
2.7 APPLICATION OF THE SPARSE RECONSTRUCTION-BASED DISCRIMINATION METHOD APPLIED TO REFINE CANNY	50

2.8 DISCUSSION	52
CHAPTER 3 THE COLOUR PREDICTIVE CODING/BIASED COMPETITION MODEL OF V1 ..	54
3.1 INTRODUCTION	54
3.2 NEUROPHYSIOLOGICAL DATA FROM SINGLE NEURONS IN MACAQUE V1	56
3.3 METHODS.....	58
3.3.1 The Photoreceptor Model	58
3.3.2 The LGN Model	58
3.3.3 The V1 Model.....	59
3.4 RESULTS	62
3.4.1 Simulation of Neurophysiological Data from colour neurons in Macaque V1	62
3.4.2 Contour Detection Results.....	65
3.4.3 Failure Cases Discussion	68
3.5 DISCUSSION	69
CHAPTER 4 THE PREDICTIVE CODING/BIASED COMPETITION MODEL OF V2.....	71
4.1 INTRODUCTION	71
4.2 THE PC/BC MODEL OF V2 FOR TEXTURE BOUNDARY DETECTION	73
4.2.1 Methods.....	73
4.2.2 Results.....	78
4.3 THE PC/BC MODEL OF V2 FOR CORNER DETECTION	83
4.3.1 Methods.....	83
4.3.2 Results.....	92
4.4 DISCUSSION	98
CHAPTER 5 A SIMPLIFIED TEXTURE GRADIENT METHOD	101
5.1 INTRODUCTION	101
5.2 METHODS.....	102
5.2.1 The Texture Gradient Method Used in the Probability of Boundary Algorithm.....	102
5.2.2 Proposed Texture Gradient Method.....	104

5.2.3 Combination with Boundaries Defined by Intensity Discontinuities Derived from the PC/BC	
Model of V1	110
5.3 RESULTS	111
5.4 DISCUSSION	116
CHAPTER 6 CONTOUR DETECTION REFINED BY A SPARSE RECONSTRUCTION-BASED	
DISCRIMINATION METHOD	119
6.1 INTRODUCTION	119
6.2 METHODS	119
6.2.1 Training Patches Extraction	120
6.2.2 Dictionaries Learning	121
6.2.3 Sparse Reconstruction Error Computation	122
6.2.4 False Contours Suppression	123
6.3 RESULTS	126
6.3.1 Contour Detection Results	126
6.3.2 Failure Cases Discussion	128
6.4 DISCUSSION	131
CHAPTER 7 CONCLUSIONS	133
7.1 SUMMARY OF FINDINGS	133
7.2 IMPLICATIONS OF THESE FINDINGS	135
7.3 LIMITATIONS OF RESEARCH	136
7.4 FUTURE RESEARCH RECOMMENDATIONS	138
BIBLIOGRAPHY	141

Table of Figures

Figure 2-1 Results of contour detections for two natural images from the BSDS300 [36] dataset.	37
Figure 2-2 Three synthetic texture images with their ground truth boundary maps from the Prague texture segmentation dataset [37].	38
Figure 2-3 The diagram of the PC/BC model of V1.	40
Figure 2-4 The procedure of the PC/BC model of V1	41
Figure 2-5 The PC/BC model of V1 with the mechanism of the lateral connectivity between the prediction neurons [39].....	43
Figure 2-6 Weight kernels of the four types of the lateral connections.....	44
Figure 3-1 Two colour natural images from the BSDS300 benchmark [36] with their grayscale version and ground truth contour maps.	55
Figure 3-2 Models of double-opponent V1 cells and single-opponent V1/LGN cells [41].	57
Figure 3-3 The procedure of the colour PC/BC model of V1.	61
Figure 3-4 Simulation of neurophysiological data [41] from a double-opponent simple cell in macaque V1.	63
Figure 3-5 Simulation of neurophysiological data [41] from another double-opponent simple cell in macaque V1.	64
Figure 3-6 Simulation of neurophysiological data [41] from a complex cell in macaque V1.	65
Figure 3-7 Results of contour detections for three colour images with their grayscale version from the BSDS300 dataset [36].	67
Figure 3-8 Performances of different contour detection algorithms evaluated by the precision-recall curve and the F-score defined in section 2.4.1 from the BSDS300 Benchmark [36].....	68
Figure 4-1 Two natural images from the BSDS300 benchmark [36] with the ground truth contour maps and the maps derived from the PC/BC model of V1 [39].	72
Figure 4-2 The diagram of the PC/BC model of V2 for texture boundary integrated with the PC/BC model of V1 [39].	74
Figure 4-3 The procedure of the PC/BC model of V2 for texture boundary detection combined with the PC/BC model of V1.	75
Figure 4-4 Results of contour detections for three images are from BSDS300 [36].....	80
Figure 4-5 Results of contour detections for three images are from Prague benchmark [37].	81

Figure 4-6 The intermediately generated maps generated from the integrated PC/BC model (top row) and the Pb texture gradient [47] (bottom row).	83
Figure 4-7 The diagram of the PC/BC model of V2 for corner detection integrated with the PC/BC model of V1 [39].	84
Figure 4-8 The procedure of the PC/BC model of V2 for corner detection combined with the PC/BC model of V1	86
Figure 4-9 Some receptive fields of the prediction neurons defined in the PC/BC model of V2.	88
Figure 4-10 The boundary-to-boundary corner lateral connections are only defined between the V2 prediction neurons representing corners that can roughly form a smooth curve.	89
Figure 4-11 Some weights of the boundary-to-boundary corner lateral connections in the PC/BC model of V2.	90
Figure 4-12 The texture-to-texture corner lateral connections are only defined between the V2 prediction neurons that represents corners with sides at similar angles and similar orientations.	91
Figure 4-13 Some weights of the texture-to-texture corner lateral connections in the PC/BC model of V2.	92
Figure 4-14 Corner detection maps for two images from BSDS300 dataset [36].	94
Figure 4-15 Results of contour detection for two images from BSDS300 dataset [36].	95
Figure 4-16 Some maps generated by the integrated PC/BC model with and without lateral connections for the two images from BSDS300 [36].	96
Figure 4-17 Some maps generated by the integrated PC/BC model with lateral connections for the two images from BSDS300 [39].	97
Figure 5-1 Filter bank for the texture gradient method used in the Pb algorithm [47].	103
Figure 5-2 Half disc filters defined in the Pb algorithm	104
Figure 5-3 Four original images from BSDS300 dataset [36] with their maximum texture gradients at each pixel generated by the proposed method and the Pb algorithm [47].	104
Figure 5-4 The procedure of the proposed texture gradient method.	105
Figure 5-5 Filter bank of the proposed texture gradient method.....	106
Figure 5-6 Half Gaussian filters defined in the proposed method.....	108
Figure 5-7 Results of texture boundary detections for two images are from BSDS300 dataset [36].	112

Figure 5-8 Performances of different contour detection algorithms evaluated by the precision-recall curves and F-score from the BSDS300 Benchmark [36].	113
Figure 5-9 Results of texture boundary detections for three images from Prague benchmark [37].	114
Figure 6-1 The procedure of the modified version of SRBD used to refine the three previously proposed contour detection methods.....	123
Figure 6-2 Performances of contour detection algorithms with and without refinement by the proposed SRBD evaluated by the precision-recall curve of the F-score defined in section 2.4.1 from the BSDS300 Benchmark [36]......	127
Figure 6-3 Performances of contour detection algorithms with and without refinement by the proposed SRBD evaluated by the precision-recall curve of the ODS (F-score) defined in section 2.4.1 from the BSDS500 Benchmark [36]......	129
Figure 6-4 Results of contour detection derived from the colour PC/BC model of V1 with and without refinement by the proposed SRBD for four images from BSDS300 [36].	130

Table of Tables

Table 2-1 Evaluation results of selected contour detectors using the BSDS300 [36] and/or BSDS500 [18] benchmarks.....	36
Table 3-1 Effect of altering the values of the four parameters introduced in the colour PC/BC model of V1 on the F-score defined in equation 2.1 from the BSDS300 Benchmark [36].	69
Table 5-1 The computation time of different stages in the proposed method and the texture gradient method used in the Pb algorithm [47] when both of the two methods are operated on 100 natural images from the BSDS300 benchmark [36].	110
Table 5-2 The computation time and F-score (defined in equation 2.1) of different methods when tested on 100 images from the BSDS300 benchmark [36].	116
Table 5-3 Effect of altering the value of six key parameters used in the proposed algorithm (integrated with the PC/BC model of V1 [39]) on the F-score (defined in equation 2.1) from the BSDS300 benchmark [36].....	117
Table 6-1 The ODS, OIS and AP of different contour detection methods.	130

Acknowledgements

I am very grateful to all the people who helped and supported me during my PhD study. I would like to express my sincere gratitude to my supervisor Dr Michael Spratling for his patient guidance, continuous encouragement and extraordinary support throughout my PhD study. The work presented in this thesis would not have been possible without the guidance from him. I would like to thank my second supervisor Dr Thrishantha Nanayakkara and my third supervisor Dr Matthew Howard for their practical suggestions on my research. I would also like to thank Dr Elizabeth Johnson in Duke University for her patient and warm-hearted explanations regarding her published papers by email.

I would like to thank Dr Yizhe Song from Queen Mary University of London and Dr Adrian Evans from University of Bath for acting as examiners for my doctoral thesis.

I am very grateful to my friends and colleagues at King's College London, especially, Dr Shan Luo, Dr Guowu Wei, Dr Ketao Zhang, Dr Peng Qi, Jie Sun, Xinsheng Zhang, Yan Yu, Chuang Liu, Chen Qiu, Menglan Jiang, and Wasif Muhammad for their friendship and the valuable suggestions and encouragements.

I owe a debt of gratitude to my parents Xiaoming Wang and Hongye Yan for their selfless love by providing fully financial support for my PhD study. I owe a debt of gratitude to my fiancée Minzhao Wang for her consistent company and encouragement throughout these four years.

Abbreviations

Abbreviation	Meaning
AP	Average precision
BEL	The boosted edge learning algorithm
BSDS	The Berkeley segmentation dataset and benchmark
BW	The orientation bandwidth in units of degrees of orientations
CFM	Confidence of false contour metric
CIELAB	A colour space including the lightness (L), red-green opponent (a^*), and blue-yellow opponent (b^*) channel.
CIEXYZ	A colour space including a luminance channel: Y and two colour channels: X and Z.
CM	Confidence metric
CNN	Convolutional neural network
c-PC/BC	The colour PC/BC model of V1
CRF	Classical receptive field
CV	The circular variance of the orientation tuning curves
DOSSC	The double-opponency and spatial sparseness constraint algorithm
gPb	Global probability of boundary
gPb-owt-ucm	Global probability of boundary-oriented watered transform-Untrametric Contour Map
H-MMC	The hierarchical multiple Markov chain algorithm
LGN	Lateral Geniculate Nucleus
LNL	Linear-Nonlinear-Linear Model
LoG	Laplacian of Gaussian
nCRF	Non-classical receptive field
ODS	F-score with a fixed threshold for all test images
OIS	Aggregate F-score with an optimal threshold per image
OMP	The orthogonal matching pursuit algorithm
O/P	The ratio of the responses to the preferred orientation and orthogonal-to-preferred orientation
OWT	Oriented watered transform
Pb	The probability of boundary algorithm
pb	Probability of boundary map

PC/BC	Predictive coding/biased competition
RGB	A colour space including the red, green, and blue channel.
SCG	The sparse code gradients algorithm
SRBD	The sparse reconstruction-based discrimination method
SVM	Support Vector Machine
TG	Texture gradient
UCM	Unrametric Contour Map
V1	Primary visual cortex
V2	Second visual cortical area
VOS	The vector order statistics algorithm

Chapter 1 Introduction

1.1 Context

Contour detection is the process of finding meaningful smooth curves within an image. These smooth curves can be formed by some image locations having a specific property. For instance, the property can be the discontinuities in intensity, colour and texture within an image. These smooth curves can also be the boundaries between different objects or between meaningful regions within an image, and then such process can also be called image segmentation [1]. However, image segmentation and contour detection, while similar, employ different processes. The aim of image segmentation is to partition image into different meaningful regions, using closed contours, but is not concerned with other kinds of meaningful smooth curves within an image. In contrast, all kinds of meaningful smooth curves should be located by the process of contour detection and these curves do not have to be closed. The contours of an image preserve the structural properties of the image and provide meaningful and valuable information for further image analysis and interpretation tasks. Object contours play an important role in human visual perception such as scene understanding and object recognition [2]. Contour detection is also a fundamental technique and crucial step for many applications in computer vision such as object localization [3], object recognition [4], robot vision [5], medical image analysis [6] and 3D reconstruction problems [7].

Because there is no universal and formal definitions of what contours are within a natural image, human judgement is the only standard to determine if a pixel belongs to a contour or not. Therefore, many contour detection methods were devised to mathematically model the features which people may use to recognize contours within an image. All contour detectors can be classified into two categories: non-biologically-inspired methods and biologically-inspired methods. The difference between these two categories is that a biologically-inspired method should be inspired or can simulate some biological activities related to contour detection but a non-biologically-inspired method need not. Although these two kinds of algorithms are different, they are highly correlated with each other. For instance, both kinds of methods are often devised to locate contours defined by different local discontinuities in intensity, colour and/or texture. Some biologically-inspired methods can also be interpreted as a modified version of different non-biologically-inspired methods.

With respect to non-biologically-inspired contour detection methods, some methods employ the convolution of different kinds of linear filters with an input image to locate boundaries defined by discontinuities in intensity and/or colour within an image [8][9]. Some algorithms were devised to locate the same kind of boundaries by comparing each pixel with neighbouring pixels instead of an edge template (linear filter) [10][11]. Finding boundaries defined by discontinuities in texture is the purpose of another set of contour detection methods [12][13]. These methods utilized different statistical techniques to represent the structure of texture within an image first, and then employ different methods to work out which pixels are located at boundaries between two different structures of texture. Some region-based algorithms [14][15] utilized different kinds of technique to group pixels with similar values into different regions within an image first and then locate boundaries between these regions for contour detection. Contour detection can also be achieved by a set of interactive methods [16][17], which firstly need an initial curve drawn by the user and then detect contours by finding a solution to an optimization problem. One of the recent near state-of-the-art non-biologically-inspired methods [18] can locate contours defined by local discontinuities in intensity, texture and colour and integrate these local boundaries with global information derived from region-based algorithms.

Since the human visual system performs contour detection better than any machine vision system, many researchers are interested in modelling biologically-inspired methods. For instance, the psychophysical research in human perception such as phase congruency [19], the Gestalt law of closure [20][21] and symmetry [22][23] inspired some researchers to devise different contour detection methods. Another set of models were proposed to simulate neurophysiological activity probably related to contour detection in the human visual system, such as the mechanisms of collinear facilitation [24] and surround suppression [25] found in the primary visual cortex (V1) in the human visual system. Many neurophysiologically-inspired methods [26][27][28][29] have utilized a bank of linear filters to model the receptive fields of neurons and then achieved the mechanisms of collinear facilitation and surround suppression by using different mathematical models. Eye movement behaviour has also been simulated for contour detection methods [30][31]. The sparse representation technique [32] inspired by the neurophysiological evidence for sparse coding in the human/animal visual system [33], had also been applied to contour detection by employing dictionaries obtained from training images to place image patches into two classes: edges and non-

edges [34], which is based on the neurophysiological hypothesis that natural images allow a sparse decomposition in some redundant bases (or dictionary). Currently, the state-of-the-art contour detection algorithm [35] is implemented using a deep convolutional neural network (CNN), a method which is inspired by the cortical visual hierarchy. This algorithm produces results similar to those of human beings when tested with a standard benchmark [18].

This standard benchmark, the Berkeley Segmentation Dataset and Benchmark (BSDS) [36][18], was defined to evaluate the performance of different contour detection algorithms based on natural images by comparing results derived from a contour detector with contours identified by human observers. Another dataset [37] that was defined to evaluate the performance of different texture segmentation methods is used to evaluate texture boundary detection methods proposed in this thesis.

1.2 Aims

Because the human visual system performs contour detection better than any machine vision system, a contour detection method which is inspired by and/or can simulate biological activities related to contour detection should theoretically have the potential to generate improved contour detection results. Therefore, the current research was focused on biologically-inspired methods. The aim was to improve the performance of different extended versions of an existing biologically-inspired method, simplification of a non-biologically-inspired method to a biologically-inspired method, and application of a biologically-inspired method to improve other biologically-inspired and non-biologically-inspired methods. The good performance of an improved biologically-inspired method potentially indicates that the biological activity simulated by this improved method is more likely to relate to contour detection in the human/animal visual system and this improved method is more likely to simulate this biological activity well.

Recently, a comprehensive neurophysiologically-inspired model: the predictive coding/biased competition (PC/BC) model of the primary visual cortex (V1) was proposed by Spratling [38]. This model can simulate the neural activity in V1. Then Spratling [39] applied the PC/BC model of V1 to perform contour detection by using receptive fields of V1 neurons (modelled as first order derivative of Gaussians) and

achieving the mechanism of collinear facilitation [24] and surround suppression [25] with some other mechanisms identified in V1 to locate boundaries defined by local discontinuities in intensity. The PC/BC model of V1 [39] can slightly outperform a recent near state-of-the-art methods when this method is also restricted to using only local discontinuities in intensity at single-scale in grayscale images. However, boundaries defined by intensity discontinuities are unlikely to cover all contours recognized by human observers within all categories of images. Hence the first aim of the current research was to extend the PC/BC model of V1 to improve its performance of contour detection by locating more contours that may be recognized by humans in an image. Such extension should be also inspired by and/or simulate some neurophysiological activities in the human/animal visual system related to contour detection. The improvement of the performance of contour detection derived from an extended version of the PC/BC model of V1 was expected to be indicated when evaluated by the standard benchmark [36][18].

Two recent near state-of-the-art contour detection methods have performance not far from humans. The second aim of the current research was to simplify the first stage of one of these methods [18] to a biologically-inspired method, which can reduce the computation time of the first stage of this contour detection algorithm without generating a worse result when evaluated by the standard benchmark [36][18]. The third aim of the current research was to employ a modified version of a biologically-inspired method [34] to improve the performance of some previously defined leading contour detection methods.

1.3 List of Contributions

The main contributions of this research are as follows:

- Inspired by neurophysiological data from single neurons in Macaque V1 [40][41], the PC/BC model of V1 is extended to be able to locate boundaries defined by local discontinuities in colour within a colour image. The colour PC/BC model of V1 has significantly better performance compared with the original PC/BC model and slightly outperforms some recently proposed contour detection method which use more cues and/or require a complicated training procedure (Chapter 3).
- The PC/BC model of V1 [39] is also extended to simulate the second visual cortical area (V2) in an attempt to detect texture boundaries [42][43] and corners [44][45][46]. This was expected to improve the performance of the contour

detection under the framework of the PC/BC model. Both versions of the PC/BC model of V2 for texture boundary and corner detection are integrated with the PC/BC model of V1 respectively to successfully locate texture boundaries and enhance some weak contours that are obtained from the PC/BC model of V1 and defined by insignificant intensity discontinuities within an image (Chapter 4). However, the two integrated PC/BC models both generate a worse performance compared to the original PC/BC model of V1 [39].

- The probability of boundary (Pb) algorithm [47] is the first stage of a recent near state-of-the-art contour detection algorithms, gPb-owt-ucm [18]. The texture gradient method utilized in the Pb algorithm [47] is modified to be more biologically-plausible (Chapter 5). The modified texture gradient method can simulate texture boundary detection in V2 [42][43]. The computation time of the modified method is reduced, but the performance of the proposed method in locating texture boundaries in the standard benchmark [36] is the same as the original texture gradient method used in the Pb algorithm.
- Inspired by the neurophysiological evidence for sparse coding in the human/animal visual system [33], the sparse reconstruction-based discrimination (SRBD) method proposed by Mairal et al. [34] has been applied in [48] to refine the contour detection results produced by the Canny edge detector [8]. The results are significantly improved compared to the unrefined Canny detector. A modified version of SRBD is applied to refine the contour detection results produced by two recent near state-of-the-art algorithms [18][49] and the colour PC/BC model of V1 (Chapter 6). Evaluation using the standard benchmark [36][18] indicate that for all three algorithms boundary detection is at least slightly improved by the proposed biologically-inspired refinement.

1.4 Outline of Thesis

How the different kinds of non-biologically-inspired and biologically-inspired contour detection algorithms model the features which people may use to recognize contours within an image are described in Chapter 2. The comparison between the standard contour detection benchmark [36][18] utilized in this thesis and other popular contour detection benchmarks [50][51][52] that can be employed to evaluate the performance of all contour detection algorithms is then described in Chapter 2. How the standard benchmark [36][18] and a texture boundary detection benchmark [37] are used to evaluate the performance of different contour detection methods are also introduced in

detail in Chapter 2. Afterwards, Chapter 2 also briefly introduces the three previously proposed contour detection methods, because the research in this thesis is closely related to these three algorithms. These three methods include a neurophysiologically-inspired method (the PC/BC model of V1 [39]), a recent near state-of-the-art contour detection methods (gPb-owt-ucm [18]), and the application of SRBD [48] to refining the result produced by the Canny edge detector [8].

In Chapter 3 and Chapter 4, different extensions of the PC/BC model of V1 are defined. Because the PC/BC model of V1 [39] can only locate boundaries defined by local discontinuities in intensity, many other contours within an image cannot be detected by this biologically-inspired model. Therefore, in Chapter 3, the PC/BC model is extended to locate boundaries defined by local colour discontinuities inspired by neurophysiological data from single neurons in macaque V1 [40][41]. Chapter 3 first briefly introduces this neurophysiological data, and then describes how the colour PC/BC model of V1 simulates this neurophysiological data. The good performance of the colour PC/BC model of V1 is also shown in Chapter 3. Texture boundaries [42][43] and corners [44][45][46] are detected in the second visual cortical area (V2) in the human/animal visual system. The PC/BC model of V1 is extended to simulate these two neurophysiological activities in Chapter 4. Hence Chapter 4 defines, in detail, the PC/BC model of V2 for texture boundary detection and corner detection, and introduces how each PC/BC model of V2 is integrated with the original PC/BC model of V1 [39]. These two integrated models were expected to improve the performance of contour detection under the framework of the PC/BC model, but they both generate a worse result. Hence Chapter 4 also presents these worse results and discusses some possible reasons why each integrated PC/BC model fails to improve the performance.

The probability of boundary (Pb) algorithm [47] is the first stage of the recent near state-of-the-art algorithm gPb-owt-ucm [18]. The texture gradient method used in the Pb algorithm devised to locate texture boundaries employed the K-Means algorithm to cluster the responses of Gaussian derivative filters to represent the structure of texture. However, the K-Means algorithm is computationally time-consuming and a non-biologically-inspired method. Hence, a simplified and biologically-inspired version of the texture gradient method is defined in Chapter 5, which directly uses the filtered responses without the application of the non-biologically-inspired K-Means algorithm to represent the structure of texture. The differences and the complexity advantage of

each difference between the proposed method and the texture gradient used in the Pb algorithm are highlighted in Chapter 5. Chapter 5 also shows the result of this simplified and biologically-inspired version of texture gradient method and discusses the rationality of the proposed method.

Inspired by the neurophysiological evidence for sparse coding in the human/animal visual system [33], the sparse reconstruction-based discrimination (SRBD) method [34] has been applied in [48] to refine the contour detection results derived from the Canny edge detector [8] producing results that are significantly improved compared to those of unrefined Canny detector. In Chapter 6, a modified version of the biologically-inspired SRBD method is proposed to refine the contour detection result derived from the colour PC/BC model of V1 proposed in Chapter 3 and the two recent near state-of-the-art algorithms [18][49]. The procedure used to apply the modified version of SRBD to refine these contour detection methods with the modified process compared to [48] are described in detail in Chapter 6. Chapter 6 also evaluates the performance of each of the three refined contour detection methods and makes comparison to the performance of their original methods. Finally Chapter 7 summarizes the findings reported in this thesis and discusses the implications and limitations of these findings, and possible future research directions

Chapter 2 Background and Related Works

2.1 Introduction

Hitherto, there is no universal and formal definition of what contours are within a natural image. Thus human judgement is the only standard to determine if a pixel belongs to a contour or not. Therefore, numerous contour detection algorithms have been devised to mathematically model the features which people may use to recognize contours within an image. These contour detectors can be sorted into two categories: non-biologically-inspired methods and biologically-inspired methods. The difference between these two categories is that a biologically-inspired method should be inspired by or be able to simulate at least one biological activity related to contour detection but a non-biologically-inspired method does not need. These two types of contour detectors are different but highly correlated with each other. For instance, some methods belonging to both categories are devised to locate contours defined by different local discontinuities in intensity, colour and/or texture. Some biologically-inspired methods can also be interpreted as a modified version of different non-biologically-inspired methods. Hence, section 2.2 reviews the different types of previously proposed non-biologically-inspired contour detection methods first. Afterwards, section 2.3 focuses on the description of different previously proposed biologically-inspired contour detection methods. Then section 2.4 introduces the benchmarks [36][18][37] that are used to evaluate the performance of the proposed contour detection algorithms and also explains why such comparison techniques are selected. Because there are three different contour detectors that are closely related to the content of the following Chapters in this thesis, these three previously proposed methods are introduced in section 2.2 and 2.3, and then described in more detail in sections 2.5 (the PC/BC model of V1 [39]), 2.6 (gPb-owt-ucm [18]) and 2.7 (SRBD [48]). Section 2.8 discusses the possible biologically-inspired extensions or improvement that can be made to these three algorithms.

2.2 Literature review on different non-biologically-inspired methods for contour detection

Firstly, different linear filters were devised to make convolution with an input image to locate intensity discontinuities. For instance, Canny [8] utilized first order derivative of Gaussian filters. Marr et al. [9] chose to use Laplacian of Gaussian filters. Daugman [53]

employed Gabor filters. Many other filters were also devised to locate discontinuities in intensity within images including but not limit to Sobel, Prewitt, Beaudet, and Robert [54], log-Gabor, difference of Gaussian, Cauchy [55], and higher order derivatives of a Gaussian.

Algorithms that employ a linear filter can be interpreted as using a measure of the match between pixels and a given edge template to locate intensity discontinuities [56]. In contrast to utilizing an edge template (linear filter), another set of methods locate intensity and colour discontinuities by comparing each pixel with its neighbouring pixels. For instance, Tomasi et al. [10] proposed a method (the so-called bilateral filter), which employed the mean Euclidian distance for grey values (or vectors in the CIELAB colour space) between each pixel and its neighbouring pixels to locate intensity (or colour) discontinuities. Other versions of the bilateral filter method with improved performance were also defined in [57][58]. The morphological gradient method defined in [11][59] employed the distance between the maximum and minimum intensities within neighbourhood of each pixel to locate intensity discontinuities. Evans and Liu [60] extended the morphological gradient method to process colour images by computing the maximum Euclidean distance between all pairs of vectors in the RGB colour space, and refined the colour morphological gradient method to be more robustness for noise. Astola et al. [61] defined another method to process colour images called Vector Order Statistics (VOS), which firstly sorted all vectors according to their aggregated distances to all other vectors and then utilized the distance between the highest and lowest ranked vector to locate colour discontinuities within an image.

Because the algorithms mentioned above are not able to locate contours defined by texture discontinuities, several statistical approaches were devised for locating texture boundaries. For instance, Huang et al. [12] utilized a two-sample statistical test of independence to compute the dissimilarity between adjacent image regions. Ojala et al. [13] employed distributions of local binary patterns and pattern contrast to evaluate the similarity of neighbouring image regions using G statistics to compare distributions. Jain et al. [62] filtered images using a bank of Gabor filters, utilized a Gaussian window to measure texture features in the filtered image, and adopted a square-error clustering algorithm to integrate texture features and locate texture boundaries. Will et al. [63] defined textures through the statistical distribution of Gabor filters responses, and the edges of textures were determined as pixels with equal posteriori probability between

two classes of textures. The Hierarchical Multiple Markov Chain (H-MMC) algorithm defined by Scarpa et al. [64], employed a Markov chain to characterize texture information through their spatial interaction with neighbouring regions, and also created the Texture Fragmentation and Reconstruction algorithm to perform unsupervised texture segmentation on the H-MMC.

Methods based on only intensity, colour or texture discontinuities are unlikely to detect all contours recognized by humans in all categories of images. Therefore, Martin et al. [47] devised a well-known method called the Probability of Boundary (Pb) algorithm, which is able to combine intensity, colour and texture discontinuities within an image for contour detection. There were three gradient methods: brightness, colour and texture which were defined in the Pb algorithm to locate boundaries defined by corresponding discontinuities. Specifically, the brightness and colour features are processed by summing up the normalized square of difference in the L channel and the a^* b^* channels of CIELAB colour space respectively between adjacent regions. Texture features are processed by first applying the K-Means algorithm to cluster filtered image responses to yield cluster centres called textons. Each pixel is then assigned a label corresponding to its closest texton. The difference in the number of pixels with the same texton label between adjacent regions summed over all possible texton labels is then used to locate texture boundaries. Afterwards, Ren [65] and Maire et al. [66] found that the combination of the intensity, colour and texture discontinuities across different scales can improve the performance of the Pb algorithm.

Besides methods based on locating intensity, colour and texture discontinuities within an image, another set of methods (which can also be called image segmentation methods) tried to group pixels into different meaningful regions first and utilized boundaries between these regions for contour detection. For instance, Adams et al. [14] proposed a seeded region growing algorithm, which iteratively explored neighbour pixels from each seed pixel and assigned each explored pixel with the label of the region it belonged to, according to the similarity between the value of this explored pixel and the mean value of pixels within different regions it adjoined. Comaniciu et al. [15] utilized the mean shift technique to cluster neighbouring pixels which have similar values into different regions. Najman et al. [67] employed the watershed algorithm to group pixels with similar values into different regions. The idea of the watershed algorithm [67] is a kind of geographical analogy which utilizes First In First Out

breadth-first principle to flood ‘water’ from several local minimal pixels, then constructed a barrage when the ‘water’ from two different local minima met, and finally immersed the whole image into a lake. These barrages are the boundaries between different regions. Arbelaez [68] proposed the ultrametric contour map method to hierarchically merge adjacent over-segmented regions derived from other image segmentation methods by comparing the similarity of the colour information between pairs of adjacent regions. Shi et al. [69] devised the Normalized Cuts algorithm, which iteratively partitioned input images into several meaningful regions. The Normalized Cuts algorithm firstly converts an input image to a graph: each pixel is treated as a node, and the weight of connection between each pair of nodes is defined by the similarity of their pixel values. Then these nodes are bipartitioned into two different regions by satisfying the following conditions: first, the aggregated weight of connection that needs to be removed between each pair of nodes belonging to the two regions must be the minimum; second, the number of nodes belonging to the two regions must be as close as possible. This procedure can be iteratively utilized to partition an image into many regions.

Contour detection can also be achieved by a set of interactive methods called active contours, which firstly need an initial curve drawn by the user and then detect contours by finding the solution of a minimization problem. This minimization problem can be interpreted as the evolution of the initial curve to the actual contours within an image according to a given functional space, in which all functions have certain desired regularity properties. Specifically, Kass et al. [16] devised the active contour method by defining regularity properties including the minimisation of the distance between the active contour and image discontinuities, the maximisation of the smoothness of the active contour, and the minimisation of the total length of the active contour. Afterwards, various versions of the active contours method were proposed to improve the performance of contour detection result by solving the minimization problem in different ways, for instance, the Fourier active contours [17], the finite element active contours [70], dual active contours [71], and the fuzzy curve tracing [72]. As well as the above mentioned regularity properties defined in different versions of active contours methods, Leventon et al. [73] and Tsa et al. [74] added the maximum of similarity between the shape of curves drawn by the user and the privilege shape of a set of training objects into the functional space of their minimization problem. Dumitras et al. [75] extended the active contours method to be able to process colour images.

A recent near state-of-the-art contour detection algorithm gPb-owt-ucm [18] was derived from the integration of some previously proposed non-biologically-inspired methods. Specifically, the multi-scale Pb algorithm [65] extended the Pb algorithm [47] to locate local intensity, colour and texture discontinuities across multiple scales within an input image. Afterwards, Maire et al. [66] combined these multi-scaled local discontinuities with the global information of contours by employing the Normalized Cuts algorithm [69]. Then contours derived from the global Probability of Boundary (gPb) [66] were firstly post-processed by the improved Watershed Transform [66] to close the nonconsecutive contours obtained from gPb and to partition the image into different regions, and secondly hierarchically merge these different over-segmented regions by using the ultrametric contour map [68]. Song et al. [76] found that there are still redundant contours within the result derived from gPb-owt-ucm [18], and these redundant contours can be further removed by employing a measure of Laplacian graph energy. Gupta et al. [77] extended the gPb-owt-ucm [18] method to make effective utilization of the depth information within an image. The gPb-owt-ucm [18] method is described in more detail in section 2.6.

2.3 Literature review on different biologically-inspired methods for contour detection

Because the human visual system performs contour detection better than any machine vision system, many researchers are interested in devising models to simulate different biological activities related to contour detection. For instance, some contour detection methods are inspired by psychological research in human perception. For instance, Oppenheim et al. [78] found that the human visual system had strong responses to points where there were a lot of frequency components at the same phase in the Fourier transform of an input image. Several methods such as [19][79] utilized this property of phase congruency in human perception to detect edges within an image. A famous psychological theory, the Gestalt laws, can also be utilized to improve the performance of other contour detection methods. For instance, the law of closure, which means that human perception tends to fill in the visual gap when parts of a familiar object are missing, has been frequently simulated by different algorithms. Specifically, Sappa [20] firstly generated a partially connected graph from many input edge points, then the minimum spanning tree algorithm was computed to generate a single path through the

boundaries of different regions within this graph and close those open contours. Ren et al. [21] found that the incorporation of curvilinear continuity structure defined by utilizing the Delaunay triangulation algorithm can improve the performance of different local contour detectors. Wang et al. [80] took fragmented contours derived from an edge detection method, and then proposed the ratio contour algorithm to sequentially connect these fragments in order to close contours according to the length of gap and average curvature among different fragments. Papari et al. [81] devised an adaptive pseudo-dilation algorithm, which can identify long curvilinear structures in the map of edge points by employing the information of context dependent structure. The pseudo-dilation algorithm [81] can also be integrated with a multi-threshold contour detector which is able to remove texture edge within the contour detection result. The law of closure was also simulated in other contour detection methods [82][83][84][85].

Another Gestalt law, the law of symmetry, which means that human perception also tends to group two unconnected symmetric elements into one combined symmetrical object, has also been the inspiration for other algorithms. For instance, Stahl et al. [22] firstly paired line segments detected from images to construct many trapezoids, and then identified and connected the sequence of these trapezoids into the closed symmetric contours. Mohan et al. [23] achieved the law of symmetry by coupling the line segments extracted from edge and corner detection algorithm and grouping them into structures using heuristic algorithms. Bai et al. [85] chose to utilize the discrete curve evolution method to extract the skeleton of different objects within an image first and then locate the contours that are symmetrical around this skeleton. There were also several other methods proposed in [87][88][89] that can simulate the Gestalt law of symmetry.

Another set of models were proposed to simulate neurophysiological activity believed to be related to contour detection in the biological visual system. Firstly, some kinds of linear filters had been found to have a similar shape to the receptive field of neurons in retina, lateral geniculate nucleus (LGN) or primary visual cortex (V1). For instance, Derrington et al. [90] and Price et al. [91] found that the Laplacian of Gaussian filter and difference of Gaussian filter is able to simulate the property of neurons in macaque LGN. Dacey et al. [92] worked out that the frequency response and bandpass of macaque retina neurons can be simulated well by the difference of Gaussians. Young et al. [93] modelled motion and spatial orientation properties of simple cells in V1 by using the derivative of Gaussian filters. Ringach [94] and Jones et al. [95] found that the

spatial profile of the receptive field of orientation-tuned simple cells in macaque and cat V1 can be well described by two-dimensional Gabor filters.

Furthermore, there are two neurophysiological mechanisms, collinear facilitation [24] and surround suppression [25], that have been identified by observing the responses of neurons in V1 within the human/animal visual system. Specifically, collinear facilitation means that neural responses to the centre stimulus can be enhanced when its neighbouring stimuli are collinear with it. In contrast, neural responses to the centre stimulus can also be reduced by surrounding stimuli, which is called the surround suppression. Since these two mechanisms found in V1 are widely believed to play an important role in contour detection, several biologically-inspired methods have been devised which include at least one of these two neurophysiological mechanisms. For instance, Ursino et al. [26] devised a model of V1 comprising ‘hypercolumn’ and ‘orientation column’ simple cells with Gabor receptive fields. These neurons enhanced neighbouring neuron responses if their preferred orientations were collinear with each other (simulating the mechanism of collinear facilitation) and suppressed neighbouring neuron responses if their preferred orientations were orthogonal with each other (simulating the mechanism of surround suppression). Wei et al. [27] simulated the classical receptive field (CRF) of simple cells in V1 by using Gabor filters, and then modelled the regions outside the CRF, called the non-classical receptive field (nCRF), by using difference of Gaussian filters. This model can achieve collinear facilitation by disinhibiting neuron responses within the nCRF that are located on two sides of CRF along CRF’s preferred orientation and achieve surround suppression by inhibiting neuron responses within nCRF that are located close to CRF such that the angle between the preferred orientation of these neurons and the CRF is similar. Yang et al. [28] used derivatives of Gaussians to model the CRFs of neurons in V1 and also utilized difference of Gaussians to simulate the nCRF. However, in this method the mechanism of surround suppression was achieved by modulating the combined inhibition weights derived from three types of local features: orientation, luminance and luminance contrast. Zhao et al. [96] applied Gabor filters to simulate the receptive field of V1 neurons and simulated the mechanism of surround suppression by computing the co-occurrence statistics of the surrounding neuron responses described by divergence of orientation histograms within surrounding regions. Xiao et al. [97] also used Gabor filters to model the receptive field of simple cells in V1 and applied a uniform framework to define both the collinear facilitation and the surround suppression. The

collinear facilitation is applied if the preferred orientations of surrounding neuron responses were collinear with the centre neuron and the surround suppression is adopted if the preferred orientations of surrounding neuron responses were parallel with the centre neuron. Other neurophysiologically-inspired models (of V1) that simulate these two mechanisms have also been proposed in [98][99][100].

Recently, a more comprehensive model: the Predictive Coding/Biased Competition (PC/BC) model of V1 was proposed by Spratling [39], which was a convolutional neural network applied to perform contour detection by using the receptive fields of V1 neurons (modelled as first order derivative of Gaussians) to locate boundaries defined by local intensity discontinuities. To improve the performance of this model, a set of lateral connections between neurons in two populations were defined to enhance weak contours that are co-circular with neighbouring strong contours (collinear facilitation) and suppress texture (surround suppression) detected by this model. Other visual properties of neuron responses identified in V1 in the human/animal visual system such as orientation tuning, size tuning, spatial frequency tuning, temporal frequency tuning, and cross-orientation suppression were also achieved in the PC/BC model of V1. The PC/BC model of V1 is described in more detail in section 2.5.

Eye movements fixate locations that probably provide useful spatial information for the interpreting of shapes and structures in an image [101]. A model incorporating simulated eye movements has been used for contour detection within an image. Mignotte [30] firstly transformed the image into a log-polar map to simulate the reconfiguration of information projected on the retina, and then utilized line segmentation detector algorithm [102] to construct smooth contours based on the log-polar map. Růka et al. [31] utilized the Laplacian operator to model the receptive field of retina, and then applied non-overlapping convolution operation to simulate visual information processing in mammalian retina. The involuntary eye tremors were also simulated in [31] by shifting the input images in a random direction, because some necessary overlapping operation exists in mammalian retina for contour detection.

Inspired by neurophysiological evidence for sparse coding in the biological visual system [33], sparse representation techniques have been widely used for different image processing tasks including contour detection. Sparse coding is based on the neurophysiological hypothesis that natural images allow a sparse decomposition in

some redundant basis (or dictionary). For instance, Mairal et al. [34] proposed a sparse reconstruction-based discrimination (SRBD) method, which utilized two dictionaries to place image patches into two classes: edges and non-edges. The dictionaries were obtained by training with two different classes of image patches extracted from a set of training images using the K-SVD algorithm [35]. Afterwards, a test image patch was represented as a sparse code using these two dictionaries. Finally, the classification of the test image patch was determined by assigning it the category of the dictionary that produced the smallest reconstruction error. Mairal et al. [48] increased the discriminative power of the SRBD method by adding a discriminative training algorithm to the procedure of dictionary learning, and then applied the SRBD method to refining the result derived from the Canny edge detector [8], which is significantly improved compared to the unrefined Canny detector. Section 2.6 describes the application of SRBD to refining the Canny [8] in more detail. Maire et al. [103] defined a two-layer framework to recursively learn generic dictionaries optimized for the sparse coding of image patches. Firstly, pooling across multiple scales without any specific task target, and then utilized a logistic regression method to learn a transfer function to optimize the reconstruction of contours or non-contours patches extracted from training images. The learnt transfer function was then applied to detect contours within test images. A recent near state-of-the-art contour detection algorithms: Sparse Code Gradients (SCG) [49] utilized a learned sparse representation method [35] to generate sparse representations of each input image. It then applied a similar gradient method as that utilized in the Pb algorithm [47] to compute the gradients of the sparse code across multiple scales and multiple orientations. Afterwards, a support vector machine (SVM) was applied to learn a classification function based on these sparse coding gradients in order to locate contours within an image. Finally, these contours derived from local information were combined with the global information of contours by employing the Normalized Cuts algorithm [69]. SCG generates results that equal the other leading method gPb-owt-ucm [18].

Two current state-of-the-art contour detection methods are both implemented using deep convolutional neural networks (CNNs). These two methods were called CEDN [104] and HED [105]. Specifically, Yang et al. [104] defined a six encoder layer and six decoder layer deep CNN and used the ground truth of both object recognition dataset and natural images dataset to train the deep CNN to learn features of different object contours and background contours. Xie et al. [105] devised the holistic-nested deep

CNN model, which was directly trained on whole images themselves instead of using image patches and applied a single stream deep CNN with multiple side outputs to progressively refine the contour detection result. There were also some other deep CNN methods devised for contour detection. For instance, Bertasius et al. [106] utilized the Canny edge detector [8] to extract candidate contour pixels, and then defined a five-layer deep CNN to learn top-down object-level features around these candidate pixels for contour detection. Ganin et al. [107] directly input test image patches into a pre-trained deep CNN and applied the nearest neighbour algorithm to match the CNN output derived from test image patches against the sample CNN output derived from training image patches with known annotations for contour detection. Shen et al. [108] defined a deep CNN method that received patches from previously obtained contour pixels. These contour patches were firstly partitioned into compact clusters in terms of their inherent structure and shape before they are trained by different deep CNNs. Afterwards, the output of these multi-classes deep CNNs were integrated using a positive-sharing loss function. Chen et al. [109] utilized domain transform filtering, instead of fully-connected conditional random fields defined in a deep CNN model, to reduce the computation time of this model of contour detection. Kivinen et al. [110] introduced a covariance feature depending on the squared responses of a filter into a deep CNN model and then integrated multiple scales and semantic levels of this covariance feature in this deep CNN model.

Current state-of-the-art deep CNN based contour detection methods can have almost the same performance as human beings when evaluated by the most widely-used natural image dataset [36][18]. However, the deep CNN based contour detection algorithms work like a black box, which means that this kind of algorithms cannot help us to understand how different specific image features can be utilized and integrated for contour detection in the human/animal visual system.

2.4 Evaluation Benchmarks for Contour Detection Methods

There are many benchmarks that have been proposed to evaluate the performance of different contour detection methods. Some of them have been widely used to evaluate many previously (and even recently) proposed contour detectors, including the BSDS300/500 [36][18], PASCAL VOC 2012 [50], NYUD [51], and RuG40 [52] datasets. The BSDS300/500 benchmark is a natural image dataset containing 100/200 test images with 200 training images. This benchmark [36][18] was originally devised

for the evaluation of contour detection methods but can also be utilized to evaluate image segmentation algorithms. The PASCAL VOC 2012 benchmark contains thousands of images in both training and test sets comprised of 20 foreground object classes (such as person, bicycle, cat, cow, motorbike etc.) and one background class. This benchmark [50] is originally applied for evaluating object recognition methods but can also be used for testing image segmentation and contour detection methods. There are 1449 indoor scene colour images with depth information in NYUD dataset [51] gathered from a broad range of commercial and residential building. This dataset [51] was originally defined for the evaluation of image segmentation algorithms that make use of depth information. More recently, the NYUD dataset [51] has also been used to evaluate general image segmentation methods (without using depth information) and contour detection methods. The RuG40 dataset [52] comprises only 40 natural images whose content are mostly one or two animals in a natural habitat. This dataset [52] is devised for evaluation of contour detectors.

In this thesis, the BSDS300/500 has been selected as the benchmark applied to evaluate the performance of the different proposed and improved biologically-inspired contour detection methods. This choice is due to following advantages of the BSDS300/500 benchmark compared with other popular benchmarks. The BSDS300/500 benchmark is described in detail in section 2.4.1. First, the BSDS300/500 is the most widely used benchmarks for the evaluation of contour detection methods, and almost all recently proposed contour detection methods have used this benchmark for evaluation. Therefore, the application of the BSDS300/500 benchmarks [36][18] is beneficial to making comparison between the proposed methods and other previously proposed methods. Second, compared with the PASCAL VOC 2012 [50] and NYUD [51] benchmark, the precision-recall measure employed in the BSDS300/500 is specifically defined for the evaluation of contour detection methods. Therefore, when contour detection methods have been tested with the PASCAL VOC 2012 and the NYUD datasets, the precision-recall measures applied in the BSDS300/500 were always chosen to be used for evaluation and comparison. Third, the ground truth in the BSDS300/500 is the map of contours identified by different human beings, whereas the ground truth in the PASCAL VOC 2012 and NYUD dataset is the map of labels of different objects marked by different human beings. In other words, the ground truth of these two datasets is the map of boundaries between different labelled objects, which can lead to some salient contours located within objects not being included in the ground truth of these two

datasets. Therefore, the application of the BSDS300/500 benchmark is more precise for evaluating whether a method can simulate the contour detection performed by the human visual system. Fourth, compared with RuG40 dataset [52], the content and quantity of test images in the BSDS300/500 benchmark is richer and bigger, and the ground truth of RuG40 dataset is the map of contours identified by only one person, hence the evaluation results from the BSDS300/500 benchmark should be more comprehensive and accurate. Fifth, the BSDS300/500 benchmark has a specific set of training images compared with the NYUD and RuG40 dataset. If the values of parameters within a contour detection method need to be determined before working on test images, the BSDS300/500 benchmark should generate a more comparable evaluation results, because the values of parameters are determined using the training image set.

Although the BSDS300/500 benchmark has many advantages, other popular benchmarks still has their own pros for evaluation of different contour detection methods. For instance, the quantity of test images in both PASCAL VOC 2012 [50] and NYUD [51] dataset are significant bigger than the BSDS300/500 benchmark [36][18], which should provide more accurate evaluation results regarding whether a contour detection method being tested can locate boundaries of different objects well. The training images and test images in PASCAL VOC 2012 dataset are initially selected for object recognition. Therefore, if a contour detection algorithm is devised as the first stage of an object recognition method, the PASCAL VOC 2012 is the most appropriate benchmark that should be used for evaluation. Compared with other three popular benchmarks, the NYUD dataset is the only one focusing on evaluating the performance of contour detection methods on indoor scene objects. Hence if a contour detection method is devised as the initial stage of some higher level computer vision applications related to indoor environment or to test if a contour detection method can perform well on indoor scenes, the NYUD dataset is the most appropriate benchmark for evaluation.

Because contour detection methods specifically devised for locating boundaries defined by discontinuities in texture are proposed in this thesis, another evaluation method, the Prague segmentation dataset [37], as well as the BSDS300/500 benchmark is used to test the efficiency of these methods. The Prague texture segmentation dataset [37] includes 20 grayscale and 80 colour synthetic natural texture images and is devised for

the evaluation of texture segmentation methods. The Prague texture segmentation dataset [37] is briefly described in section 2.4.2.

2.4.1 The BSDS300 and BSDS500 benchmark

The BSDS300 dataset [36] is a standard benchmark for evaluating the performance of contour detection and image segmentation algorithms. The BSDS300 dataset contains 100 test images and 200 training images of size 481-by-321 pixels. They are all natural images of people, animals, plants, buildings, man-made objects and some natural scenes. The BSDS300 dataset [36] provide both qualitative and quantitative evaluation methods. For each image belonging to the training and test images sets, multiple human subjects have identified meaningful contours within this image. This results in 5 to 10 binary maps of contours identified by the different human beings which are the ground truth of each image. The contour map derived from aggregating these 5 to 10 human identified contour maps is the ground truth of this image for qualitative evaluation and comparison. But these 5 to 10 maps of contours identified by humans themselves are the ground truth of this image for quantitative valuation. To quantitatively evaluate the performance of a contour detection algorithm, the BSDS300 benchmark employs a metric, the F-score, which is defined by:

$$F = \frac{2PR}{(P + R)} \quad (2.1)$$

where P is the precision and R is the recall. The precision P represents the probability of that a contour pixel identified by the detector also appearing in at least one of the human identified contour maps. The recall R represents the average probability of a contour pixel appearing in each human identified contour map also being identified by the detector. In practice, the highest F-score is 0.79, which is the performance obtained by a human observer compared with other human observers. Since the contours identified by the detector probably may be accurate but slightly mislocalized, the approach defined in [111] is applied to allow tolerance to these slightly mislocalized contour pixels and treat them as perfectly matched contour pixels.

Because the results generated from a contour detection method being tested are the probability of boundary map whose values range continuously from 0 to 1, the BSDS300 benchmark uses a set of threshold values ranging from 0 to 1 to convert the contour map of each image being tested to different binary maps. This benchmark then computes the F-score of each binary map as defined in equation 2.1 derived from each

image and shows the highest F-score with its corresponding precision, recall and threshold value as the evaluation result of the contour detector on each image. With respect to the final evaluation result of all 100 test or 200 training images, this benchmark also applies a set of threshold values ranging from 0 to 1 to convert contour maps of all images being tested to a set of binary maps, and then treats these binary maps derived from the same threshold value and from all of test/training images as a whole to compute the F-score of it as defined in equation 2.1. Finally, this benchmark can output a precision-recall curve that indicates the precision and recall values at different threshold values and also the highest F-score with its corresponding precision, recall and threshold value as the quantitative evaluation result of the contour detector being tested on all test/training images

The BSDS500 dataset, which was first utilized in [18], is an extended version of the BSDS300. There are 200 fresh natural images added as test images, and the original 100 test images become validation images. Three different quantities are employed for evaluation: the F-score with a fixed threshold for all test images (ODS), which is the same as the F-score used in the BSDS300 benchmark; the F-score with an optimal threshold per image (OIS); and the average precision (AP) for the full range of recall values, which is equivalent to the area under the precision-recall curve. The quantitative evaluation results of several contour detectors including two state-of-the-art deep learning algorithms on the BSDS300 and/or BSDS500 benchmark are illustrated in Table 2-1. Figure 2-1 shows contour detection results for two images from the BSDS300 datasets [36]. These results include the ground truth of maps identified by human observers, the contour maps produced by one leading algorithm gPb-owt-ucm [18] and by the PC/BC model of V1 [39].

If the values of different parameters used in a contour detection algorithm need to be determined before that algorithm is tested, this should be performed using the training image set. After its optimal parameter values have been obtained, this algorithm can be run on test image set to generate the final evaluation result in the BSDS300/500 benchmark.

Table 2-1 Evaluation results of selected contour detectors using the BSDS300 [36] and/or BSDS500 [18] benchmarks.

The quantitative evaluation measures include: the F-score defined in equation 2.1 with a fixed threshold for all test images (ODS), the F-score defined in equation 2.1 with an optimal threshold per image (OIS), and the average precision (AP) for the full range of recall values.

Methods	BSDS300			BSDS500		
	ODS	OIS	AP	ODS	OIS	AP
Human	0.79	0.79	—	0.80	0.80	—
CEDN [104]	—	—	—	0.78	—	—
HED [105]	—	—	—	0.78	0.80	0.83
SCG [49]	0.71	0.73	0.75	0.74	0.76	0.77
gPb-owt-ucm [18]	0.71	0.74	0.73	0.73	0.76	0.73
gPb [65]	0.70	—	—	—	—	—
Multi-scale Pb [65]	0.68	—	—	—	—	—
Canny refined by SRBD [48]	0.66	—	—	—	—	—
Pb [47]	0.65	—	—	—	—	—
Pb (brightness+texture) [47]	0.63	—	—	—	—	—
PC/BC Model of V1 [39]	0.61	—	—	—	—	—
Pb (brightness) [48]	0.60	—	—	—	—	—
Pb (texture) [47]	0.58	—	—	—	—	—
Canny [8]	0.58	—	—	—	—	—

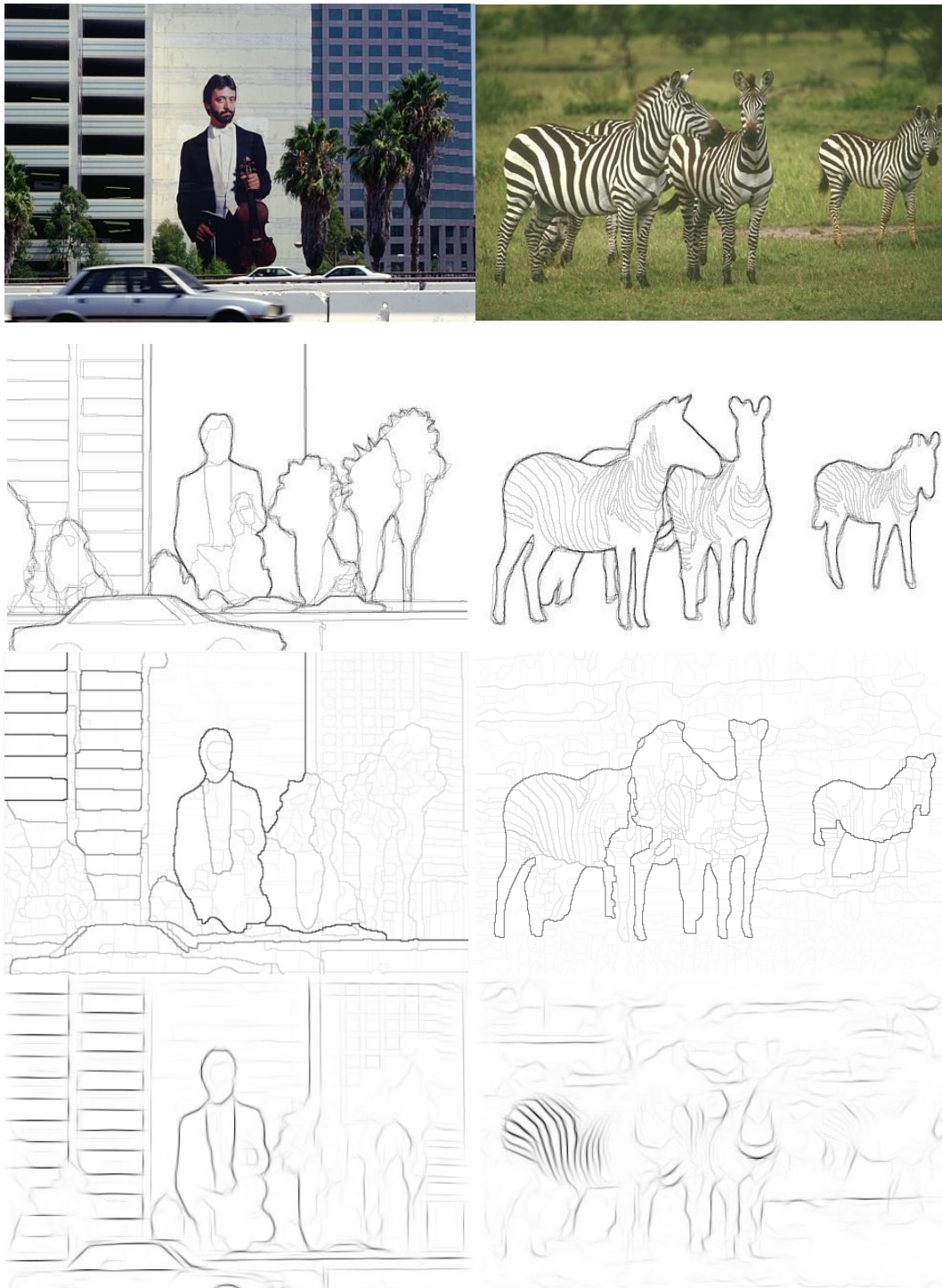


Figure 2-1 Results of contour detections for two natural images from the BSDS300 [36] dataset.

These contour detection results are produced by (from top to bottom): human observers, gPb-owt-ucm [18], and the PC/BC model of V1 [39].

2.4.2 The Prague Texture Segmentation benchmark

The Prague texture segmentation benchmark [37] is a standard benchmark devised to evaluate the performance of texture segmentation algorithms. The Prague dataset [37] contains 80 colour and 20 grayscale synthetic texture images of the resolution of 512-by-512 pixels. In this thesis, only the 20 grayscale images need to be used. This is

because contour detection methods specifically devised for texture boundary detection in this thesis are only designed to process grayscale images. Three grayscale test images in the Prague dataset [37] with their ground truth boundary maps are shown in Figure 2-2. The proposed texture boundary detection algorithms are evaluated using the BSDS300/500 benchmark, but the further evaluation of these algorithms on the Prague dataset [37] can provide a more detailed judgement on the efficiency of the proposed texture boundary detection methods due to the type of the test images in the Prague dataset. All of the quantitative evaluation measure defined in the Prague dataset such as correct segmentation (CS), over segmentation (OS), omission error (O), class accuracy (CA) and other measures cannot be used in this thesis. This is because all of these measures provide an evaluation of ‘segments’ rather than ‘contours’. In other words, these measures are all defined to evaluate whether each pixel within a test image is assigned to the correct segments and no measure is defined to evaluate whether each synthetic texture boundary pixel in a test image has been accurately located. Hence, the measure of F-score defined in equation 2.1 is also used in the Prague dataset for evaluation of the texture boundary detection methods proposed in this thesis.

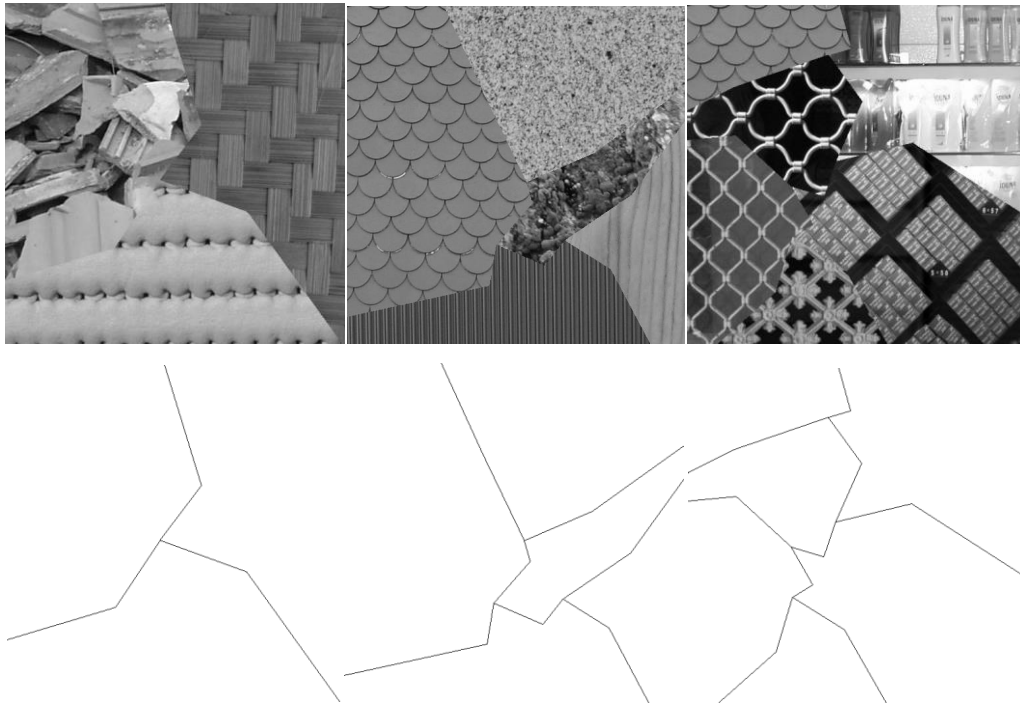


Figure 2-2 Three synthetic texture images with their ground truth boundary maps from the Prague texture segmentation dataset [37].

2.5 The predictive coding/biased competition model of V1

2.5.1 The LGN Model

A grayscale input image (I) is pre-processed by convolution with a Laplacian of Gaussian (LoG) filter (l) with standard deviation equal to $\sigma_{LGN}=2$ pixels to produce the input to the PC/BC model of V1, as defined by:

$$X = \tanh\{\kappa_{LGN}(I * l)\} \quad (2.2)$$

The convolution result is processed by a multiplicative gain (κ_{LGN}). The strength of κ_{LGN} is set to a value of 2π in this model. The positive and rectified negative responses in X are divided into two images X_{on} and X_{off} , which are used to simulate the output of neurons in retina and LGN whose receptive fields are circular-symmetric on-centre/off-surround and off-centre/on-surround respectively. The procedure of pre-processing in this LGN model is illustrated on the left of Figure 2-3. The values in X_{on} and X_{off} located near border within a distance $2.5\sigma_{LGN}$ are set to zero to prevent that the PC/BC model of V1 finding contours located along the border of the input image.

2.5.2 The V1 Model

The PC/BC model of V1, which is illustrated in Figure 2-3, is defined by the following equations:

$$E_o = G(X_o) \oslash (\epsilon_2 + \sum_{k=1}^p (v_{ok} * Y_k)) \quad (2.3)$$

$$Y_k \leftarrow (\epsilon_1 + Y_k) \otimes \sum_o (w_{ok} * E_o) \quad (2.4)$$

where $o \in \{ON, OFF\}$; X_o signifies the input to the model of V1 from the LGN model; error-detecting neuron responses are signified by E_o ; the prediction neuron responses are signified by Y_k , v_{ok} and w_{ok} are kernels which indicate the same synaptic weights for a particular class (k) of neuron, but v_{ok} need to be normalized to make the maximum weight in v_{ok} equal to one and w_{ok} also need to be normalized to make the sum of all weights in w_{ok} equal to one; p represents the total number of the classes of the prediction neurons. In equation 2.4, ϵ_1 is used to determine the baseline responses and the increase rate in the prediction neurons responses, and is set to 1×10^{-5} . In equation 2.3, ϵ_2 is employed to avoid division by zero and control the minimum significant input activity, and is set to 1×10^{-3} . $\oslash$ and $\otimes$ represent element-wise division and

multiplication respectively. G represents a function that clips values at 1 (values which are greater than 1 are set to 1). The operation of convolution is signified by $*$ and the operation of cross-correlation is indicated by $\star$. Equation 2.3 uses the operation of convolution but equation 2.4 employs the operation of cross-correlation to assure that the feedback weights between an error-detecting neuron and a prediction neuron are able to correspond to the feedforward weights between the same two neurons.

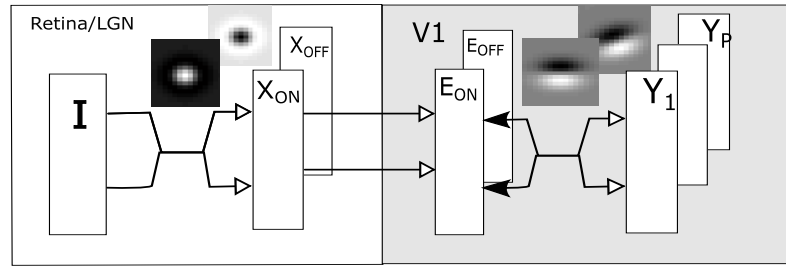


Figure 2-3 The diagram of the PC/BC model of V1.

White rectangles represent the neuron populations. The interconnections between different neuron populations are indicated using arrows. Open (filled) arrows represent feedforward excitatory (feedback inhibition) connections. Many-to-many connectivity between two different neuron populations are indicated by crossed connections. Parallel connections signify a one-to-one mapping between neurons in two populations. The boxes neighbouring crossed connections illustrate the type of weight kernels utilized. Specifically, the LGN model employed Laplacian of Gaussian kernels, and the first order derivative of Gaussian kernels are utilized in the V1 model. The same kernels are used in both feedforward and feedback connections, but scaled differently [28]. The symbols I , X , E , Y are consistent with symbols defined in equation 2.2, 2.3 and 2.4.

The prediction neuron responses Y are set to zero initially. The PC/BC model of V1 is operated recurrently by updating prediction neuron responses Y computed by equation 2.4 substituted into equation 2.3 and equation 2.4 to recursively compute the dynamic prediction neuron responses at each of all 30 iterations. The procedure of the PC/BC model of V1 is illustrated in Figure 2-4. Equation 2.3 is utilized to compute the error-detecting neuron responses E , which is proportional to difference between the predicted input reconstructed from the prediction neuron responses and the actual input from the LGN model to the V1 model. The procedure of updating the prediction neuron responses Y to minimize the error between the actual input X from the LGN model and the predicted input reconstructed from prediction neuron responses Y in the V1 model is defined in equation 2.4. The procedure of the minimization can be achieved because if the predicted input reconstructed from Y is weaker (stronger) than the actual input X in equation 2.3, the value of this error-detecting neuron response in E should be greater (less) than one, which results in the value of this prediction neuron response in the

updated Y can be increased (decreased) in equation 2.4.

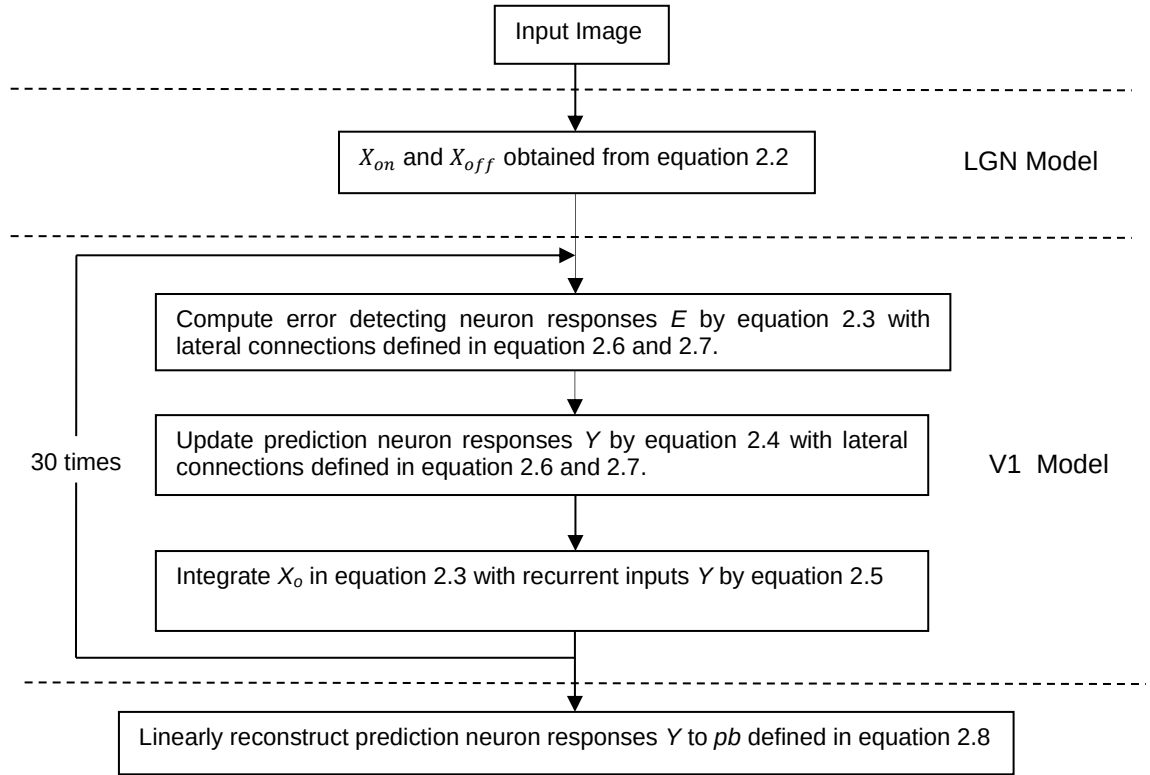


Figure 2-4 The procedure of the PC/BC model of V1

The v_{ok} and w_{ok} represent the same synaptic weights for a particular class, k , of neuron. In this model, they are defined using the first order derivative of Gaussian, which is an odd-symmetric filter that resembles odd-symmetric spatial receptive fields of the simple cells in V1 [92]. The weights to the ON and the OFF channels of the input from the LGN model are defined by dividing the positive and rectified negative values in the first order derivative of Gaussian filters. The length of the first order derivative of Gaussian filters are set to $\sigma_{V1}=3$ pixels, and the width of these filters in the perpendicular direction are set to $\sigma_{LGN}=2$ pixels. The first order derivative of Gaussian filters are defined in the range $[0^\circ, 360^\circ)$ in steps of $\sigma_c = 22.5^\circ$. Hence, there are overall 16 classes of prediction neurons so far.

To improve the performance of the PC/BC model of V1 for contour detection, a set of lateral connections are defined to enhance weak contours that are co-circular with neighbouring strong contours, and suppress the intensity discontinuities which are not real contours but texture. The suppression of texture within this model is achieved by employing another set of prediction neurons to specially represent texture and ignore the responses of this set of neurons in the final result. Therefore, another 16 classes of

prediction neurons are firstly defined with identical receptive fields to the prediction neurons previously defined. Each pair of prediction neurons with the same receptive fields is then utilized to represent real boundaries and texture respectively. To differentiate these two groups of prediction neurons, four types of lateral connections need to be defined: those in which the pre- and post-synaptic prediction neurons both represent boundaries (boundary-to-boundary connections), those in which the pre-synaptic prediction neurons represent texture and post-synaptic prediction neurons represent boundary (texture-to-boundary connections), those in which the pre- and post-synaptic prediction neurons both represent texture (texture-to-texture connections), and those in which the pre-synaptic prediction neurons represent boundary and post-synaptic prediction neurons represent texture (boundary-to-texture connections).

To implement the lateral connectivity in this model, the output of all prediction neuron responses from all channels are utilized as additional inputs to the V1 model to achieve the procedure of recurrent connections, as defined by:

$$X_k = Y_{k-2} \quad \text{for } k=3,4,5\dots34 \quad (2.5)$$

where X_k is substituted into equation 2.3. The PC/BC model of V1 with the mechanism of lateral connections is shown in Figure 2-5. Hence when the lateral connections are utilized in the V1 model, the subscripts for X in equation 2.3 should range from 1 to 34, which correspond to the two input images {ON, OFF} from the LGN and the 32 (16 orientations $\times$ 2 categories of edges (boundary or texture)) classes of prediction neurons. Afterwards, the convolution of those additional inputs with the weight kernel of the lateral connectivity is used to compute the lateral input to each prediction neuron. The convolution operation used in this model is able to effectively duplicate the same property of lateral connectivity between prediction neurons whose receptive fields are identical at all locations. The weight kernel of boundary-to-boundary lateral connections between prediction neuron a of type (orientation) i and prediction neuron b of type j is defined by:

$$W_{ij}^{ab} = Sexp \left\{ -\frac{(d - 2\sigma_D)^2}{2\sigma_D^2} - \frac{\theta^2}{2\sigma_C^2} - \frac{\psi^2}{2\sigma_A^2} \right\} \quad (2.6)$$

where d indicates the Euclidean distance between the coordinates of points (neurons) a and b ; θ signifies the difference in angle between the orientation of filters of type i and j and the tangent to a circle passing through points a and b ; ψ represents the curvature of the circle passing through points a and b ; the extent and specificity of the lateral connections is determined by the parameter σ , and the strength of the lateral

connections is determined by the parameter S . This definition of boundary-to-boundary lateral connections results in a strong connections between prediction neurons that have roughly co-circular receptive fields. The boundary-to-boundary lateral weights are shown on the top-left of Figure 2-6. Equation 2.6 can also be used to define the weight kernels of texture-to-boundary lateral connections. However, the θ need to be off-set by 90 degree and result weight kernels defined in equation 2.6 need to be rotated by 90 degree to ensure that the boundary prediction neurons are able to be activated by texture prediction neurons have roughly perpendicular receptive fields. The texture-to-boundary lateral weights are shown on the top-right of Figure 2-6.

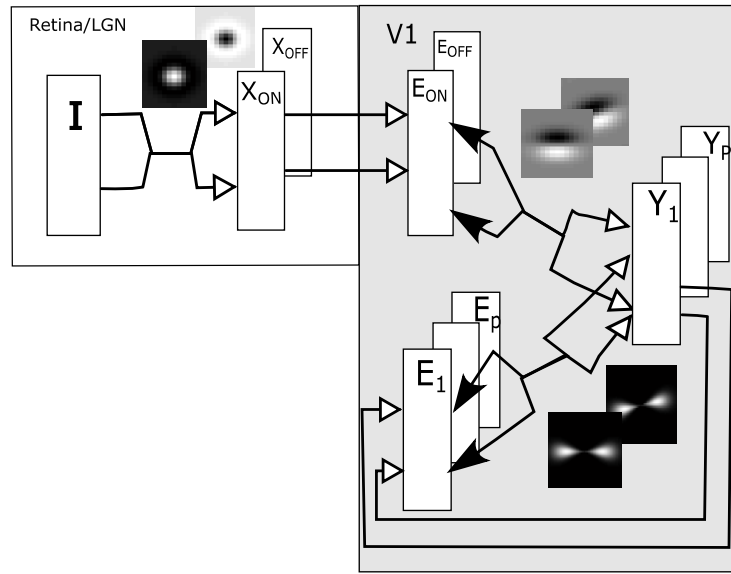


Figure 2-5 The PC/BC model of V1 with the mechanism of the lateral connectivity between the prediction neurons [39].

White rectangles represent the neuron populations. The interconnections between different neuron populations are indicated using arrows. Open (filled) arrows represent feedforward excitatory (feedback inhibition) connections. Many-to-many connectivity between two different neuron populations are indicated by crossed connections. Parallel connections signify a one-to-one mapping between neurons in two populations. The boxes neighbouring crossed connections illustrate the type of weight kernels utilized. Specifically, the LGN model employed Laplacian of Gaussian kernels, and the first order derivative of Gaussian kernels and weight kernels of lateral connectivity defined in equation 2.5 and 2.6 are utilized in the V1 model. The same kernels are used in both feedforward and feedback connections, but scaled differently [39]. The symbols I , X , E , Y are consistent with symbols defined in equation 2.2, 2.3, 2.4 and 2.5.

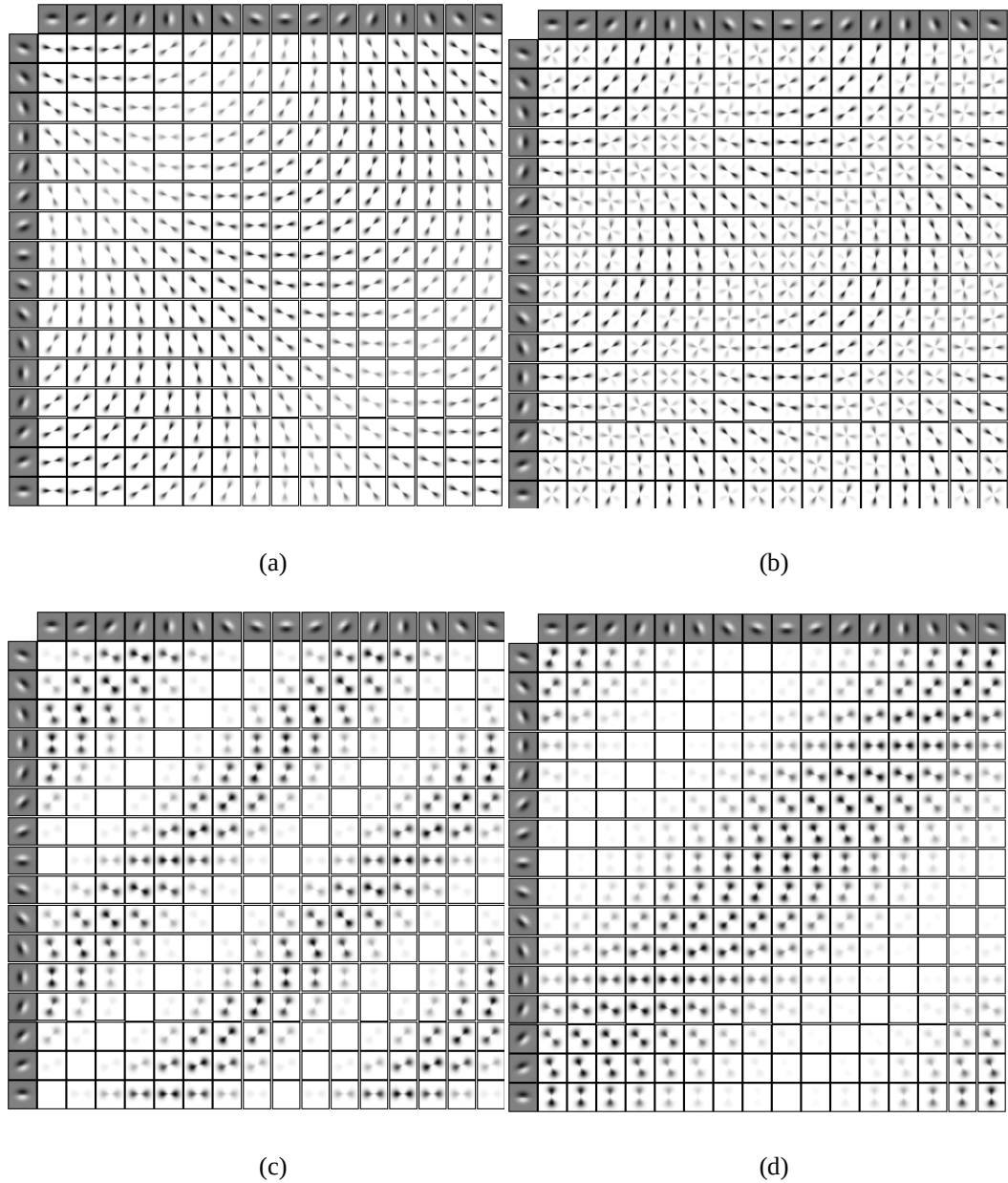


Figure 2-6 Weight kernels of the four types of the lateral connections.

In each sub-figure, the left-hand column indicates the receptive field of the post-synaptic neuron and the top row represents the receptive field of pre-synaptic neuron. Light (dark) pixels represent strong weights to the OFF (ON) channel of the input from the LGN. Each other element within the four sub-figures indicates the spatial distribution of the weight kernels of the lateral connections between the corresponding pre-synaptic and post-synaptic neuron. (a) indicates the lateral connection weights between prediction neurons representing boundary edges. Prediction neurons with co-circular receptive fields are linked. (b) indicates the lateral connection weights from prediction neurons representing texture edges to prediction neurons representing boundary edges. Prediction neurons with perpendicular receptive fields are linked. (c) indicates the lateral connection weights from prediction neurons representing boundary edges to prediction neurons representing texture edges. Prediction neurons with perpendicular receptive fields are linked. (d) indicates the lateral connection weights between prediction neurons representing texture edges. Prediction neurons with parallel receptive fields are linked. The receptive fields of the post-synaptic and pre-synaptic neurons have the size of 21×21 pixels while the size of the lateral connection weights is 51×51 pixels [28].

The weight kernels of texture-to-texture lateral connections between prediction neuron a of type (orientation) i and prediction neuron b of type j is given by:

$$W_{ij}^{ab} = S \exp \left\{ -\frac{(d - 2\sigma_D)^2}{2\sigma_D^2} - \frac{\phi^2}{2\sigma_C^2} - \frac{\omega^2}{2\sigma_A^2} \right\} \quad (2.7)$$

where d represents the Euclidean distance between the coordinates of points (neurons) a and b ; ϕ signifies the difference in angle between the line subtended by points a and b and a line perpendicular to post-synaptic filter; ω indicates the difference in angle between the orientation of filters of type i and filters of type j ; the extent and specificity of the lateral connections is still determined by the parameter σ , and the strength of the lateral connections is still determined by the parameter S . This definition of texture-to-texture lateral connections results in strong connections between prediction neurons that have roughly parallel receptive fields. The lateral weights of texture-to-texture are shown on the bottom-right of Figure 2-6. Equation 2.7 can also be employed to define the weight kernels of boundary-to-texture lateral connection. Whereas ω need to be off-set by 90 degree and the resulting weight kernels defined in equation 2.7 need to be rotated by 90 degree to ensure that texture prediction neurons can be activated by boundary prediction neurons that have roughly perpendicular receptive fields. The lateral weights of boundary-to-texture are shown on the bottom-left of Figure 2-6. The value of S is set to be halved for boundary-to-texture and texture-to-boundary connections compared to boundary-to-boundary and texture-to-texture connections, so that the prediction neurons representing the same class can have more weights for the lateral connections between them. In this model, the σ_D , σ_C , σ_A and S are set to 6, 22.5, 60 and 0.5 respectively. The definitions of all feedback weights v_{ij}^{ab} are equivalent to the definitions of its corresponding lateral weights w_{ij}^{ab} . But all feedback weights v_{ij}^{ab} need to be normalized to make the maximum weight for each prediction neuron equal to one.

Because the receptive fields of V1 neurons are defined by the first derivative of Gaussian filters, which is able to locate boundaries defined by local discontinuities in intensity, the output of boundary prediction neuron responses can be interpreted as the indication of the location and orientation of contours. Therefore, these prediction neuron responses can be employed to linearly reconstruct the contours being detected in an image, such that:

$$pb = \sum_{k=1}^{16} (r_{ok} * Y_k) \quad (2.8)$$

where r_{ok} are kernels signifying straight edge elements; pb represents the probability of boundary at each pixel within an image.

The contour detection results (pb) derived from the PC/BC model of V1 [39] are evaluated by the BSDS300 benchmark [36]. The evaluation results illustrated in Table 2-1 indicate that the PC/BC model of V1 can slightly outperform the leading contour detection method (the Pb algorithm [47]) when this method is also restricted to using only local intensity discontinuities at single scale in grayscale images. The contour detection results on two natural images from the BSDS300 dataset [36] obtained from the PC/BC model of V1 are illustrated in Figure 2-1.

2.6 The global probability of boundary method with the OWT-UCM algorithm

The gPb-owt-ucm [18] uses the brightness, colour and texture gradient method defined in the Probability of Boundary (Pb) algorithm [47] to locate intensity, colour and texture discontinuities within an image. Afterwards, the Pb algorithm is extended to locate these discontinuities across multiple scales [66]. Then the global contour information within the image can be obtained by employing the Normalized Cuts technique [69] and can also be integrated with the result derived from the multi-scale Pb method [66]. The global Probability of Boundary (gPb) [66] is further post-processed by using the oriented watershed transform (OWT) (an improved version of the watershed transform algorithm [67]) to close nonconsecutive contours and over-segment the image into different regions. Last but not the least, these over-segmented regions are merged hierarchically using the ultrametric contour map (UCM) [68]. Each component of gPb-owt-ucm [18] is described in the following sub-sections.

2.6.1 Brightness, colour and texture gradient method at multiple scales defined in the Pb algorithm

The basic component of the Pb contour detector is that the calculation of different oriented gradient signals $G(x, y, \theta)$ from a grayscale image I . The $G(x, y, \theta)$ is computed by placing a uniform circular disc at location (x, y) divided into two equally

half-discs at an orientation θ . Afterwards, the magnitude of the gradient G at location (x, y) is defined by calculating the χ^2 distance between the aggregated intensity values within the two uniform half-disc (g and h):

$$\chi^2(g, h) = \frac{1}{2} \frac{(g - h)^2}{(g + h)} \quad (2.9)$$

Afterwards, the Pb algorithm utilizes Savitzky-Golay filter [47] to smooth the intensity gradient, which is a kind of polynomial regression on these intensity gradients. All pixels located on the boundaries defined by local intensity discontinuities probably have the strong orientated gradient responses.

When a colour image is being processed, the Pb algorithm utilizes the CIELAB colour space to obtain brightness (the L channel) and colour (the a^* and b^* channels) information within the input image. Then the colour oriented gradient signal $G(x, y, \theta)$ is defined identically compared to the brightness gradient method by computing the χ^2 distance using equation 2.9 but only replacing the aggregated intensity values within two half-disc with the aggregated colour values in the a^* or b^* channel of the CIELAB colour space. Hitherto, overall three channels: the L, a^* and b^* are independently processed to obtain the brightness and colour oriented gradient signals $G(x, y, \theta)$ in the Pb algorithm.

The fourth channel in the Pb algorithm is defined to compute the texture oriented gradient signals. The input map to the texture channel is either the intensity values from a grayscale image or the L channel of a colour image represented by the CIELAB colour space. The texture input map is firstly convolved with a filter bank which contains both even- and odd-symmetric Gaussian derivative filters at six evenly spaced orientations. Afterwards, the filtered responses' maps are clustered by applying the K-Means algorithm. Each cluster centre is called a texton. Therefore, each pixel within an image can be assigned its closest texton. In other words, each pixel within an image is represented by the label (from 1 to 64) of its closest cluster centre for further computation. Then the texture oriented gradient signal $G(x, y, \theta)$ is defined similarly compared to the brightness/colour gradient method by calculating the χ^2 distance between g_l and h_l , as defined in:

$$\chi^2(g, h) = \frac{1}{2} \sum_{l=1}^{64} \frac{(g_l - h_l)^2}{(g_l + h_l)} \quad (2.10)$$

where g_l and h_l represent the aggregated number of pixels which are assigned by the texton label l within the two uniform half-disc respectively. The Savitzky-Golay filter [47] is also utilized to smooth the colour and texture gradients. Hitherto, the brightness, colour and texture gradient method are all defined to compute the intensity, colour and texture oriented gradient signals $G(x, y, \theta)$ for four independently processed channels, which include one intensity channel, two colour channels and one texture channel.

To detect contours corresponding to multiple scaled local discontinuities, the brightness, colour and texture oriented gradient signals are processed at three scales. These three kinds of oriented gradient signals $G(x, y, \theta)$ from four channels at three scales are linearly combined into a multiscale orientated signal $mPb(x, y, \theta)$. The Pb algorithm computes all oriented gradient signals at eight evenly spaced orientations (θ) from 0 degree to 180 degree, but only the maximum responses over these eight orientations are kept to generate a probability of boundary at each pixel, as defined in:

$$mPb(x, y) = \max_{\theta} \{mPb(x, y, \theta)\} \quad (2.11)$$

Afterwards, non-maximum suppression [10] can be utilized on $mPb(x, y)$ to generate thinned contours.

2.6.2 The Normalized Cuts algorithm and its integration with the multi-scaled Pb method

To combine multiple scale local cues derived from multi-scale Pb with global contour information within an image, the normalized cuts algorithm inspired by the graph partition problem was adopted. After several steps of the mathematical derivation [68], the solution of the minimum Normalized cut problem can be approximated by the solution of an eigenvalue and eigenvector system. The generalized seventeen eigenvectors correspond to the seventeen smallest eigenvalues derived from this eigenvalue and eigenvector system are the seventeen maps of global contour information integrated with the multi-scaled Pb algorithm [66]. To obtain the global contour information from the seventeen eigenvectors, each eigenvector is treated as an image and convolved with the first order derivative of Gaussians at multiple orientations θ to obtain the oriented signals. Therefore, the global information of contours is defined by linearly combining these seventeen oriented signals. Finally, global contour information is also linearly combined with multiple scale local cues to obtain the result of global probability of boundary (gPb) algorithm.

2.6.3 Closed contours derived from gPb using OWT-UCM

Before using the oriented watershed transform method (OWT) [18], the maximum response of the global oriented signals $gPb(x, y, \theta)$ across eight evenly spaced orientations is derived, as defined in:

$$gPb(x, y) = \max_{\theta} \{gPb(x, y, \theta)\} \quad (2.18)$$

Afterwards, the watershed transform [67] is utilized to flood ‘water’ from regional minimal of $gPb(x, y)$ based on the topographic surface defined by $gPb(x, y)$ to generate an over-segmented image. The modified version of watershed transform [67] called oriented watershed transform (OWT) is proposed to re-define the method of weighting each pixel along boundaries of each over-segmented region. The OWT estimates an orientation at each pixel on a boundary from the local geometry of the boundary itself. These orientations are derived by approximating the watershed boundaries with line segments. Afterwards, each pixel (x, y) is assigned an orientation $o(x, y)$ corresponding to the orientation of line segment it belongs to. The value of $o(x, y)$ is further approximated to the closest θ employed in $gPb(x, y, \theta)$. Finally, the weight of each boundary is defined by using the average value of $gPb(x, y, o(x, y))$ for all pixels on the arc. Finally, these over-segmented regions are recurrently merged to a hierarchy contour map: Ultrametric Contour Map (UCM) [68] by a greedy graph-based merging algorithm. The UCM is a real-valued image derived from weighting of each contour determined by its scale of disappearance. The UCM is then a weighted contour image which can generate a set of closed curves for any threshold, which is the final contour detection result of gPb-owt-ucm [18].

The contour detection results derived from gPb-owt-ucm [18] are evaluated by the BSDS300 [36] and BSDS500 [18] benchmark. The evaluation results given in Table 2-1 indicate that gPb-owt-ucm [18] is a recent near state-of-art contour detection methods. The contour detection results obtained from gPb-owt-ucm on two natural images from the BSDS300 dataset [36] are illustrated in Figure 2-1.

2.7 Application of the sparse reconstruction-based discrimination method applied to refine Canny

Contour detection results derived from the Canny edge detector [8] is refined by a biologically-inspired method: the Sparse Representation-based Discrimination (SRBD) method [48], which is inspired by the neurophysiological evidence for sparse coding in the biological visual system [33]. The sparse representation technique can be summarized as: an input signal x admits a sparse approximation over a dictionary D comprised of many elements, when a linear combination of only a few elements from D can be found that approximates the original signal x . The sparse reconstruction error is defined by the difference between the input signal x and the approximation of the input signal x using the dictionary D . If two classifications of dictionary D can be used to approximate the same input signal, the input signal is given the same classifications as the dictionary that produces the smaller sparse reconstruction error (better approximation). The procedure of the application of SRBD to refine Canny is described in detail in the following paragraphs.

To refine the edge map derived from the Canny edge detector [8], the SRBD method is applied to separate true contour pixels from false contour pixels within the Canny's edge map. The classification of these two types of pixels is based on the sparse reconstruction error of each centre pixel of test image patches. To compute the sparse reconstruction error, two types of sparse representations need to be obtained by encoding the test image patches using true and false contour dictionaries. Hence the true and false contour dictionaries need to be derived from training images first. To train these two types of dictionaries, the true and false contour patches need to be extracted from the set of training images provided with the BSDS300 [36] dataset.

Only if the centre pixel of a patch satisfies the following condition is the surrounding patch extracted: the centre pixel is identified as a contour by the Canny edge detector [8]. Afterwards, the extracted patch is classified as a true contour if the centre pixel of the patch is identified as a contour pixel by at least one human subject [36]. In contrast, the extracted patch is classified as a false contour if the centre pixel is not identified as a contour pixel by all human subjects [36]. 750 patches that satisfied these conditions were selected, at random, from each training image. As there are 200 training images this gave a total of 150000 training patches for each dictionary.

Specifically, each image patch with size $p \times p$ pixels are converted to a column vector with size $n=p^2$ pixels. If those training images are chromatic, image patches extracted from the red, green, and blue channel of the RGB colour space are concatenated to a single column vector with the size $3n$ pixels. It has been found that utilizing multiple sizes of patches and resolutions of images can improve the performance of the refinement. Therefore, the SRBD method used $p=5, 7, 9, 11, 15, 19, 23$ for both full scale and half scale images to refine the Canny edge detector. As a result, there were overall $N=28$ (14 sizes $\times 2$ classes) sets of training patches.

After two types of image patches have been extracted, the K-SVD algorithm [32] is utilized to learn 14 pairs of true and false contour dictionaries with a series of sparsity factors. Furthermore, a discriminative training method is also employed to increase the discriminative power of each pair of dictionaries by maximizing the reconstruction error obtained by encoding true (or false) contour training patches using the false (or true) contour dictionary. The two dictionaries had a fixed number of 256 elements and the number of iterations for the K-SVD algorithm was set to 25.

In order to classify each pixel within a test image as a true or false contour, $N/2=14$ sizes of patches (column vectors) were extracted from each test image at all locations where the centre pixel had been identified as a contour by the Canny edge detector [8] first. And then $N=28$ optimal sparse representations belonging to the true and false contour classes were obtained by encoding these $N/2=14$ sizes of patches (column vectors) using each pair of dictionaries and the OMP algorithm [112].

The sparse reconstruction error was then straightforwardly derived by using test image patches minus the encoding of these image patches using true or false contour dictionaries. Each corresponding pair of sparse reconstruction errors could be used to assign a test patch as a true contour (false contour) if the reconstruction errors of the true contour dictionaries was smaller (bigger) than the false contour dictionaries. However, the classifications produced for different patch sizes need to be combined to produce single, final, classification. To do this, the SRBD method used a series of sparsity factors in the procedure of dictionary learning. Therefore, each classifier can output a pair of curves of reconstruction errors as a function of the series of sparsity constraints. Afterwards, all curves generated by multiple sizes of classifiers were

combined together to generate a feature vector for classification by a logistic regression classifier or a linear SVM. Then this feature vector can be used to compute a confidence value per pixel on the contour generated by Canny.

The contour detection results derived from both Canny [8] with its refined version were evaluated by the BSDS300 [36] benchmark. The evaluation results shown in Table 2-1 indicate that given the poor performance of the Canny method in locating salient boundaries, the refined method still produce results that fall far short of state-of-the-art contour detection methods.

2.8 Discussion

The comprehensive neurophysiologically-inspired model of primary visual cortex (PC/BC) [39] has a good performance of locating intensity discontinuities for contour detection. However, the PC/BC model of V1 cannot process colour information within a colour image at present. Therefore it can be expected that a better performance can be obtained from an extended version of the PC/BC model of V1 inspired by a neurophysiological data regarding colour processing in V1. The PC/BC model of V1 can be interpreted as a sparse coding model, which utilizes sparse representations (prediction neuron responses) to encode an image employing a set of redundant basis or dictionary (first order derivative of Gaussian filters). Therefore, other types of redundant basis may also be introduced in the PC/BC model to generate sparse representations (prediction neuron responses) of different types of visual information. The redundant basis being introduced might resemble the receptive fields of neurons in other visual cortical areas, and then the PC/BC model can be utilized to simulate neuron activity in other visual cortical areas as well as the V1.

The texture gradient method employed in the probability of boundary (Pb) [47] algorithm, which is the first stage of the leading algorithm gPb-owt-ucm [18], locates texture discontinuities by using K-Means clustering of the filtered responses to represent texture information within an input image. However, the K-Means clustering algorithm is a computationally time-consuming technique and is a non-biologically-inspired process. Therefore, if it is possible to accurately represent the structure of texture using the filtered responses without the application of the K-Means algorithm, a simplified and more biologically-plausible version of the texture gradient method may be obtained.

Inspired by the neurophysiological evidence for sparse coding in the human/animal visual system [33], the biologically-inspired method, sparse reconstruction-based discrimination (SRBD) method [34], is applied in [48] to refine the Canny edge detector [8] whose performance is significantly improved after the refinement, as evaluated by the BSDS300 benchmark [36]. Therefore, it can be expected that the performance of other contour detectors including the two recent near state-of-the-art algorithms: gPb-owt-ucm [18] and SCG [49] may also be improved if they are also refined by the SRBD method.

Chapter 3 The Colour Predictive Coding/Biased Competition

Model of V1

3.1 Introduction

Compared to grayscale images, colour images provide more local cues for contour detection. For instance, as shown in Figure 3-1 using two images from the BSDS300 dataset [36], some human-identified contours, which are marked in red, are defined by insignificant local discontinuities in intensity in the corresponding grayscale images. This means that these contours are not located by some previously devised contour detectors which only process luminance information, such as the original PC/BC model of V1 [39]. In contrast, these contours are defined by significant local discontinuities in colour and hence should be easily located by a contour detection algorithm that can process both colour and luminance information within input images. As a result, many boundary detection algorithms that have been originally devised to work on greyscale image are later extended to process colour images. For instance, Perez et al. [113] extended the Canny edge detector [8] to process the hue channel of the HSI colour space. Boundaries were gradually eliminated where the hue changes were small. The eliminated boundaries were generated from confounding cues such as transparency, shading, shadow or highlights. Evans et al. [59] utilized vectors in the RGB colour space to enable the morphological gradient method [11][58] to process colour information. The grayscale image employed in the gradient method devised to locate intensity discontinuities in the probability of boundary (Pb) algorithm [47] was replaced by the a^* and b^* channels of the CIELAB colour space to locate colour discontinuities. The sparse reconstruction-based discrimination (SRBD) method [48] was also able to process colour information by concatenating the three channels of the RGB colour space for a patch into single vector and minimizing the reconstruction errors in each channel.

The predictive coding/biased competition (PC/BC) model of V1 [39] described in section 2.4 has a good performance of detecting contours defined by local intensity discontinuities. In this Chapter, the PC/BC model of V1 is extended to locate boundaries defined by local colour discontinuities. The proposed extension is inspired by neurophysiological data from single neurons in Macaque V1 [32][33]. There has been much neurophysiological research [40][41][114]-[117] on colour information processing in human/animal visual system indicating that the colour signals received



Figure 3-1 Two colour natural images from the BSDS300 benchmark [36] with their grayscale version and ground truth contour maps. Some human-identified contours, which are marked in red within both the original colour images (first row) and the grayscale version of them (second row), are defined by insignificant local discontinuities in intensity but significant local discontinuities in colour

by the three retina cones are later processed in an opponent format in the retina and LGN cells. Hence similar to other contour detection methods [118][119][120] and computer vision algorithms [121]-[123], the red-green and blue-yellow opponent model is adopted in the colour PC/BC model of V1 to simulate neurophysiological activity related to colour signal processing in retina and LGN in the human/animal visual system.

Inspired by [41], Yang et al. [118] have also proposed a biologically-inspired boundary detection method which applied first order derivative of Gaussian filters to process colour-opponent signals to locate boundaries, and then suppressed textures by assuming

texture regions are far from sparse. However in contrast to the colour PC/BC model of V1, [118] did not simulate the neurophysiological data in [41], and also produces slightly worse results on the BSDS300 benchmark [36].

The differences between the proposed colour PC/BC model of V1 and the original PC/BC model of V1 [39] described in section 2.5 include following points. First, the LGN model is extended to be able to process colour information by defining the colour-opponent model inspired by the neurophysiological data [40][41][124]. Second, the PC/BC model of V1 defined in [39] is operated three times to receive input from two independent colour channels and one luminance channel obtained from the LGN model. Third, three probability of boundary maps derived from the three outputs of the PC/BC models applied to the luminance and colour information are linearly combined to generate the final contour detection result. Forth, neurophysiological data [40][41] from colour neurons in V1 is simulated by using the responses of colour neurons obtained from the proposed colour PC/BC model of V1. Fifth, the performance of the contour detection result derived from the colour PC/BC model of V1 is significantly improved compared with the original PC/BC model of V1 [39] when tested with the BSDS300 benchmark [36].

Because the colour PC/BC model of V1 is inspired by the neurophysiological data obtained by Johnson et al. [40][41], section 3.2 briefly reviews this data. Afterwards, section 3.3 describes the colour PC/BC model of V1 in detail. Section 3.4 presents simulations of the neurophysiological data and evaluations of contour detection with the BSDS300 database [36]. The achievement of the proposed method and possible future research directions are described in section 3.5.

3.2 Neurophysiological data from single neurons in macaque V1

There is increasing evidence that many orientation-selective neurons in V1 receive colour input signals [40][41][125]-[131]. Johnson et al. [40][41] probed the properties of individual V1 neurons using drifting sinusoidal gratings of different types: luminance gratings (black and white), equiluminance gratings (red and green), and cone-isolating gratings (L-, M- and S-). According to the magnitude of the spatial frequency responses to the black-white luminance and red-green equiluminance gratings, 167 V1 neurons from 25 monkeys were classified into three groups: luminance, colour-luminance, and

colour [40]. About 100 neurons which significantly preferred luminance gratings belonged to the luminance group. They were further categorized as non-opponent cells [41], because they generated little or no response to red-green equiluminance gratings, and hence, there probably was not an opponent input from cones to these neurons. 48 neurons, which generated similar response to the luminance and red-green equiluminance gratings, belonged to the colour-luminance group [40]. 19 neurons, which significantly preferred the red-green equiluminance gratings, belonged to the colour group [40]. Both the colour-luminance and the colour neurons were further categorized as single-opponent cells or double-opponent cells [41], according to their low-pass or band-pass spatial frequency responses to the red-green equiluminance grating.

A set of double-opponent simple cells in macaque V1 generated maximum responses to both the luminance and equiluminance gratings at a certain spatial frequency and a certain orientation, and cone inputs to them were of opposite sign and not precisely balanced in strength [40][41]. Johnson et al. [41] proposed a model of these orientation-selective double-opponent neurons, which is illustrated in Figure 3-2 a). Other neurophysiological data such as [130][131] is also consistent with this model. The model of single-opponent V1/LGN cells proposed by Johnson et al. [41] is shown in Figure 3-1 b).

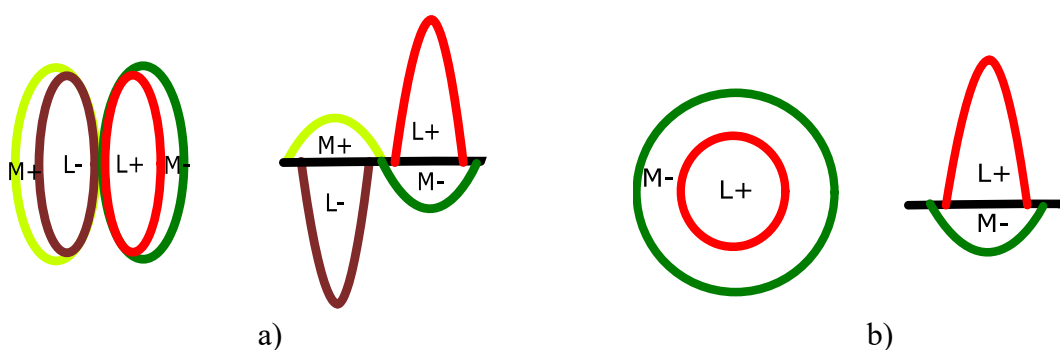


Figure 3-2 Models of double-opponent V1 cells and single-opponent V1/LGN cells [41].
a) The left diagram shows the spatial organization of the two-dimensional receptive field of orientation-selective double-opponent V1 cells, and the right diagram shows their hypothetical sensitivity profile. The spatial receptive field comprises two subregions. Within each subregion, the L- and M- cone inputs are of opposite sign and not precisely balanced in strength. The spatial symmetry resembles the asymmetric or odd-symmetric spatial receptive field of non-opponent cells (luminance cells) [131].
b) The diagrams illustrate the two-dimensional spatial receptive field and profile for one type of single-opponent V1 cell or LGN cell. Single-opponent cells receive inputs from L- and M-cones that are opposite in sign. The spatial receptive field is circular-symmetric and centre-surround.

3.3 Methods

3.3.1 The Photoreceptor Model

The three channels of the RGB colour space are used to simulate the signals received by the three types of retinal cone. L-cone, M-cone and S-cone signals are denoted by $r(x, y)$, $g(x, y)$, and $b(x, y)$, which is the red, green and blue channel of an input colour image. A yellow channel $y(x, y) = (r(x, y) + g(x, y))/2$ is calculated to provide an opponent signal to the S-cone signal. The non-opponent cells (luminance cells) in V1 are required to receive pure luminance information from an input colour image. In this model, the luminance signal is denoted by $l(x, y)$, and is the L channel of the CIELAB colour space. The values in the L channel of the CIELAB colour space can be easily derived from the three channels of the RGB colour space by an intermediary CIEXYZ colour space.

3.3.2 The LGN Model

Because many neurophysiological results [114]-[117] indicate that the LGN cells process colour information in a red-green and blue-yellow opponent format, the input signals from the cones need to be transformed into two opponent maps and a luminance map, which is similar to the colour-opponent model proposed in other contour detection methods [118][119][120] and computer vision algorithms [121][122][123]. This is achieved by:

$$I_{red-green} = r(x, y) - u * g(x, y) \quad (3.1)$$

$$I_{blue-yellow} = b(x, y) - u * y(x, y) \quad (3.2)$$

$$I_{lum} = l(x, y) \quad (3.3)$$

where $I_{red-green}$, $I_{blue-yellow}$, and I_{lum} are all three input maps which are derived from an input colour image. The u is a real number which defines the extent of unbalance in strength for two opponent cone signals, as required by the model of the double-opponent cells in V1 shown in Figure 3-2 a). Since the specific value of u was not determined in [40][41] and it potentially changes depending on the properties of the input visual stimulus [124][126][132], u was a free parameter which was set to a value of 0.8.

Because the receptive field of the LGN cells are circular-symmetric centre-surround (as shown in the Figure 3-2 b)), the three input maps ($I_{red-green}$, $I_{blue-yellow}$, and I_{lum} are

convolved with a Laplacian-of-Gaussian (LoG) filter with standard deviation $\sigma_{LGN}=2$ pixels, as defined by:

$$X_{red-green} = \kappa_{LGN} \left([I_{red-green}]_{\downarrow 2} * LoG \right) \quad (3.4)$$

$$X_{blue-yellow} = \kappa_{LGN} \left([I_{blue-yellow}]_{\downarrow 2} * LoG \right) \quad (3.5)$$

$$X_{white-black} = \kappa_{LGN} (I_{lum} * LoG) \quad (3.6)$$

where $[\cdot]_{\downarrow 2}$ represents down-sampling by a factor 2. In the proposed model, the two colour input maps $I_{red-green}$ and $I_{blue-yellow}$ are half-scale compared with the luminance input map I_{lum} . The number of pixels used as input to the V1 model is proportioned to the number of neurons used in the model of LGN. Therefore, the number of red-green orientation-selective neurons in the V1 model will be significantly smaller than the number of luminance neurons in the model. This is consistent with the neurophysiological data [40][41]. It is possible that there is blue-yellow orientation-selective neurons in V1 [133][134], but there is still no specific neurophysiological data about blue-yellow orientation-selective neurons. Here, they are modelled the same way as red-green orientation-selective neurons. The convolution result is subject to a multiplicative gain (κ_{LGN}). The strength of κ_{LGN} is set to a value of π in this model. The positive and rectified negative responses in $X_{red-green}$, $X_{blue-yellow}$ and $X_{white-black}$ are separated into six images: $X_{red-on/green-off}$, $X_{red-off/green-on}$, $X_{blue-on/yellow-off}$, $X_{blue-off/yellow-on}$, $X_{white-on/black-off}$ and $X_{white-off/black-on}$, simulating the output of cells in retina and LGN with circular-symmetric on-centre/off-surround and off-centre/on-surround receptive fields respectively. Pixels in these six images which are located within a distance $3\sigma_{LGN}$ pixels from the border are set to zero to avoid the strong edges being detected along the border of the input images.

3.3.3 The V1 Model

Three groups of neurons simulated in the colour PC/BC model of V1 including luminance neurons, red-green neurons, and blue-yellow neurons, are processed independently in this model. In other words, the colour PC/BC model of V1 can be interpreted as operating the original PC/BC model of V1 [39] three times by substituting X_o into equation 2.3 and iteratively update equation 2.3 and equation 2.4 for 30 times for each $o \in \{\text{red} - \text{on}/\text{green} - \text{off}, \text{red} - \text{off}/\text{green} - \text{on}\}$ or $\{\text{blue} - \text{on}/\text{yellow} -$

off, blue – off/yellow – on} or {white – on/black – off, white – off/black – on} .
The procedure of the colour PC/BC model of V1 is illustrated in Figure 3-3.

The v_{ok} and w_{ok} used in equation 2.3 and equation 2.4 in the original PC/BC model of V1 [39] are not changed in the colour PC/BC model of V1 for $o \in \{\text{red – on/green – off, red – off/green – on}\}$ or $\{\text{blue – on/yellow – off, blue – off/yellow – on}\}$ or $\{\text{white – on/black – off, white – off/black – on}\}$. In this colour model, they are still defined using the first order derivative of Gaussian, which is an odd-symmetric filter that resembles odd-symmetric spatial receptive fields of the non-opponent (luminance) simple cells in V1 [93][94][135]. Because orientation-selective double-opponent (colour) simple cells in V1 also have odd-symmetric spatial receptive fields [41][130][131], as shown in Figure 3-2 a), the first order derivative of Gaussian filter is also used to simulate the red-green and blue-yellow simple cells. The spatial receptive fields of all three groups of neurons v_{ok} and w_{ok} are defined as:

$$f(x, y) = \frac{1}{\sqrt{2\pi}\sigma_{V1}} \times e^{-\frac{1}{2}\left(\frac{x^2}{\sigma_{V1}^2} + \frac{y^2}{\sigma_{LGN}^2}\right)} \quad (3.7)$$

$$f'(x, y) = f(x, y) \times \frac{-y}{\sigma_{LGN}^2} \quad (3.8)$$

$$w_{ok} = \text{rotate}_k\{f'(x, y), \theta_K^{V1}\} \text{ for } k = 1, 2 \dots 16 \quad (3.9)$$

The Gaussian used to define the first order derivative of Gaussian filters had a length defined by $\sigma_{V1}=3$ pixels, and a width in the perpendicular direction defined by $\sigma_{LGN}=2$ pixels. The first order derivative of Gaussian filters $f'(x, y)$ defined in equation 3.8 are rotated by θ_K^{V1} in the range $[0^\circ, 360^\circ)$ in steps of $\sigma_c = 22.5^\circ$. The positive and rectified negative values of the first order derivative of Gaussian are also separated and used to define the weights to the ON (red – on/green – off, blue – on/yellow – off or white – on/black – off) and the corresponding OFF (red – off/green – on, blue – off/yellow – on, or white – off/black – on) channels of the input.

Similar to the intensity discontinuities, some colour discontinuities within a colour image are not real contours but texture. Therefore, to improve the performance of the colour PC/BC model of V1, the identical set of lateral connections are also defined to enhance weak contours that are co-circular with neighbouring strong contours and suppress those colour or intensity discontinuities that are not real contours but texture for each of the three groups (luminance, red-green and blue-yellow) of neurons. The definition of the lateral connectivity in the colour PC/BC model of V1 are completely

equivalent to the four types of the lateral connection filters defined using equation 2.6 and equation 2.7 in the original PC/BC model of V1 [39] described in section 2.5.

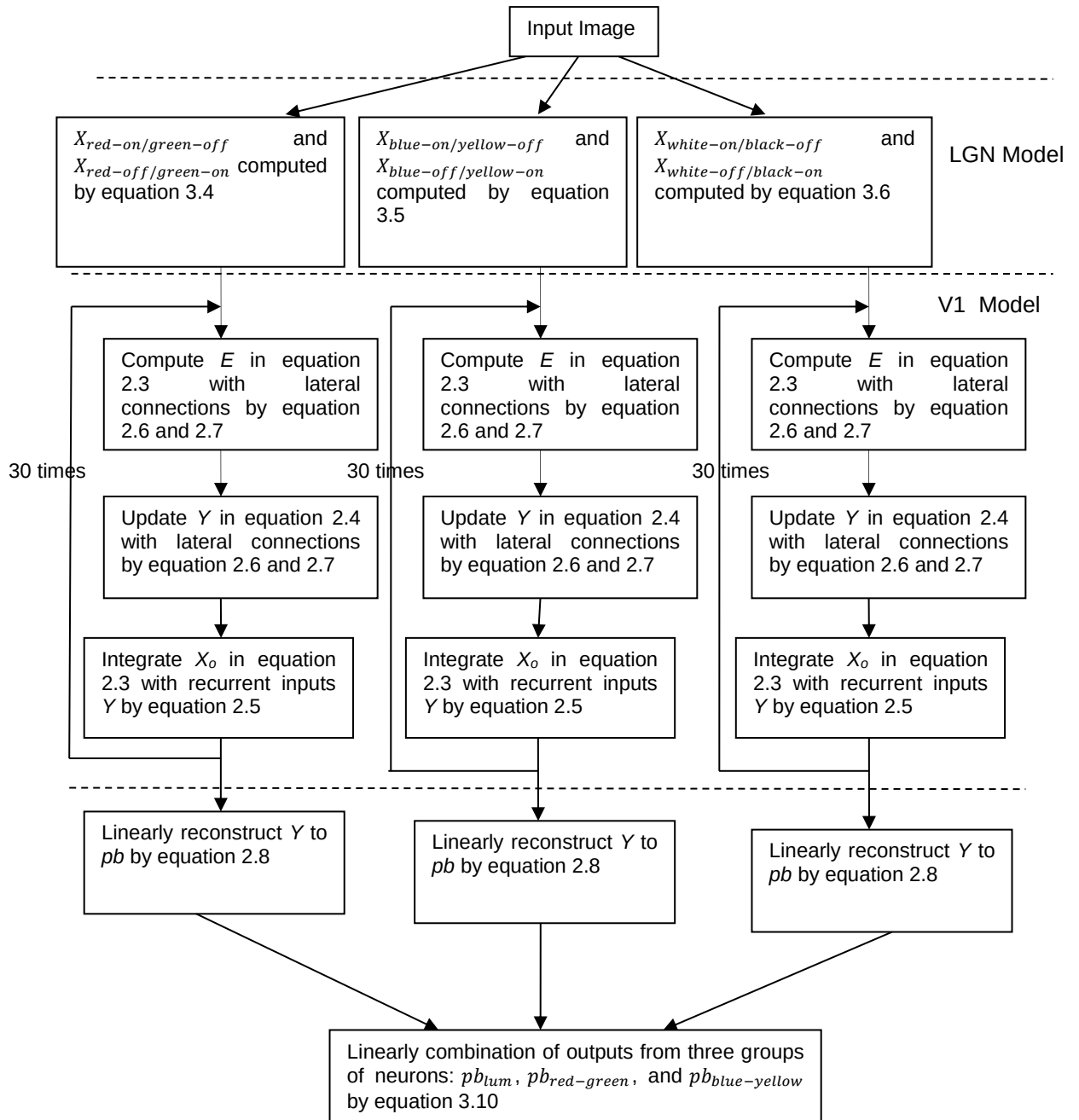


Figure 3-3 The procedure of the colour PC/BC model of V1.

For each of the three independent groups of neuron simulated in the colour PC/BC model of V1 including non-opponent simple cells (luminance neurons) and two types of double-opponent simple cells (red-green neurons and blue-yellow neurons), the $p_{b_{lum}}$, $p_{b_{red-green}}$ and $p_{b_{blue-yellow}}$ can be generated separately first using equation 2.8, and then integrated to generate the final contour detection result $p_{b_{final}}$.

$$p_{b_{final}} = G(wt1 * p_{b_{lum}} + wt2 * p_{b_{red-green}} + wt3 * p_{b_{blue-yellow}}) \quad (3.10)$$

where G is a function that clips at value 1. $wt1$, $wt2$, and $wt3$ are three real numbers that weigh the relative contribution of the output of each group of V1 neurons. $wt1$, $wt2$, and $wt3$ were set to values of 0.75, 1.125, and 1.875 respectively. Using a linear combination of the three output maps is a biologically plausible method and also a computationally efficient method to integrate different local cues. However, other combination methods such as computation of the maximum values or the Euclidean norm values across the three output maps and the application of the logistic regression or SVM method for integration of different local cues may also be used, but they have not been tested.

3.4 Results

3.4.1 Simulation of Neurophysiological Data from colour neurons in Macaque V1

Neurophysiological data [40][41] from a set of double-opponent simple cells in macaque V1 illustrates that this kind of simple cell is selective for a particular spatial frequency and orientation when it was stimulated by drifting sinusoidal gratings of two types: luminance gratings (black and white) and equiluminance gratings (red and green), as shown in Figure 3-4 a) and c). In the proposed colour PC/BC model of V1, a single prediction neuron that belonged to the group which received input from channels of $X_{\text{red-on/green-off}}$ and $X_{\text{red-off/green-on}}$ was used to simulate a single double-opponent neuron in macaque V1 [40][41]. This single prediction neuron in the model generates maximum response to both luminance gratings and equiluminance gratings at a certain spatial frequency and orientation, as shown in Figure 3-4 b) and d), which closely matches the neurophysiological data in Figure 3-4 a) and c) respectively.

The orientation selectivity can also be quantitatively evaluated by three measures: O/P, BW and CV as mentioned in [41]. Specifically, O/P is the ratio of the responses to the preferred orientation and orthogonal-to-preferred orientation, BW is the orientation bandwidth in units of degrees of orientations (half-width at half-height), and CV is the circular variance of the orientation tuning curves proposed in [136] and defined as: $CV = 1 - |R|$ where R is defined by:

$$R = \frac{\sum_k r_k e^{i2\pi\theta_k/180}}{\sum_k r_k} \quad (3.11)$$

Here, θ_k signifies equally spaced orientation angles ranging from 0 to 360 degree, r_k represent the neuron responses at each orientation. Hence, the value of CV is close to 0 if this cell has sharp orientation tuning. Neurophysiological data [41] from this set of double-opponent simple cells indicate that the orientation selectivity for this kind of simple cell is O/P=0.1, BW=22, CV=0.43 when it is stimulated by the equiluminance gratings and O/P=0.13, BW=19, CV=0.5 when it is stimulated by the luminance gratings. Hence, the values of these three quantitative measures are nearly the same for this kind of simple cell when it is stimulated by the equiluminance and luminance gratings. In the proposed colour PC/BC model of V1, the single prediction neuron that belonged to the group that received input from the channels of $X_{\text{red-on/green-off}}$ and $X_{\text{red-off/green-on}}$ also generates similar outcomes when tested with equiluminance and luminance gratings, specifically for this simulation: O/P=0.0018, BW=13, and CV=0.0288 for equiluminance gratings and O/P=0.0021, BW=11, and CV=0.0315 for luminance gratings.

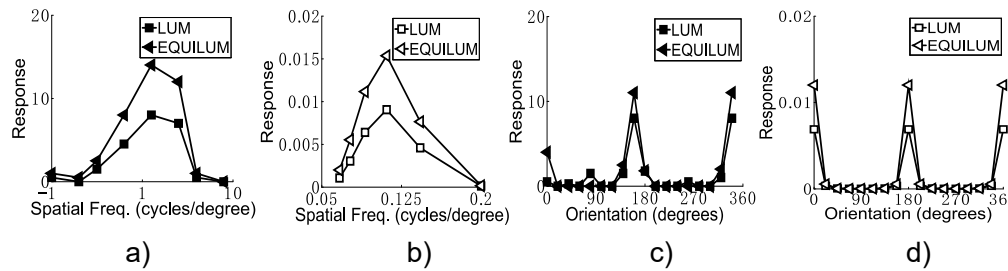


Figure 3-4 Simulation of neurophysiological data [41] from a double-opponent simple cell in macaque V1.

- a) Spatial frequency tuning curves of a single neuron in macaque V1 at the optimal orientation [41].
- b) Spatial frequency tuning curves of a single neuron in the colour PC/BC model of V1 at the optimal orientation.
- c) Orientation tuning curves of a single neuron in macaque V1 at the optimal spatial frequency [41].
- d) Orientation tuning curves of a single neuron in the colour PC/BC model of V1 at the optimal spatial frequency.

Neurophysiological data [40][41] from another set of double-opponent simple cells in macaque V1 illustrates that this kind of simple cell is selective for a particular spatial frequency and orientation when it was stimulated by drifting sinusoidal gratings of equiluminance gratings (red and green) and generated negligible responses when it was stimulated by luminance gratings as shown in Figure 3-5 a) and c). In the proposed colour PC/BC model of V1, a single prediction neuron that belonged to the group which received input from channels of $X_{\text{red-on/green-off}}$ and $X_{\text{red-off/green-on}}$ was used to simulate a single double-opponent neuron in macaque V1 [40][41]. This single

prediction neuron in the model can generate maximum response to equiluminance gratings at a certain spatial frequency and orientation but generate negligible responses to luminance gratings if the value of parameter u in this model is altered from 0.8 to 1, as shown in Figure 3-5 b) and d), which closely matches the neurophysiological data in Figure 3-5 a) and c) respectively.

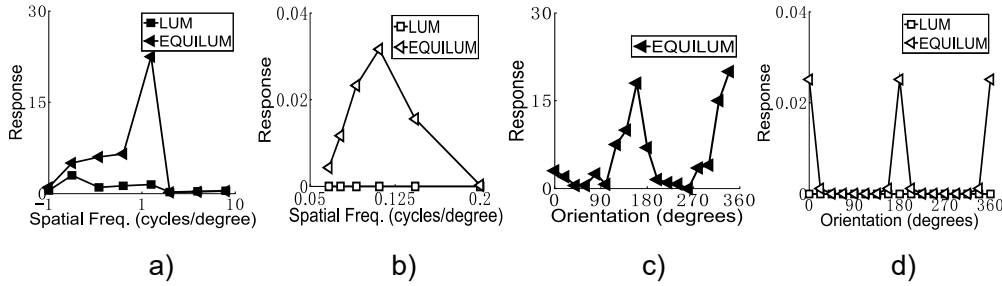


Figure 3-5 Simulation of neurophysiological data [41] from another double-opponent simple cell in macaque V1.

- a) Spatial frequency tuning curves of a single neuron in macaque V1 at the optimal orientation [41].
- b) Spatial frequency tuning curves of a single neuron in the colour PC/BC model of V1 at the optimal orientation.
- c) Orientation tuning curves of a single neuron in macaque V1 at the optimal spatial frequency [41].
- d) Orientation tuning curves of a single neuron in the colour PC/BC model of V1 at the optimal spatial frequency.

Neurophysiological data [40][41] from a complex cell in macaque V1 illustrates that this kind of complex cell is selective for a particular spatial frequency and orientation when it was stimulated by drifting sinusoidal gratings of both equiluminance gratings (red and green) and luminance gratings as shown in Figure 3-6 a) and c). In the proposed colour PC/BC model of V1, the responses of a complex cell is defined by the sum of responses of two prediction neurons at the same location but with anti-phases receptive fields [137]. These two prediction neurons also belonged to the group which received input from channels of $X_{\text{red-on/green-off}}$ and $X_{\text{red-off/green-on}}$ and the sum of responses of them were used to simulate responses of a complex cell in macaque V1. This complex cell in the model can generate maximum response to both equiluminance and luminance gratings at a certain spatial frequency and orientation as shown in Figure 3-6 b) and d), which closely matches the neurophysiological data in Figure 3-6 a) and c) respectively.

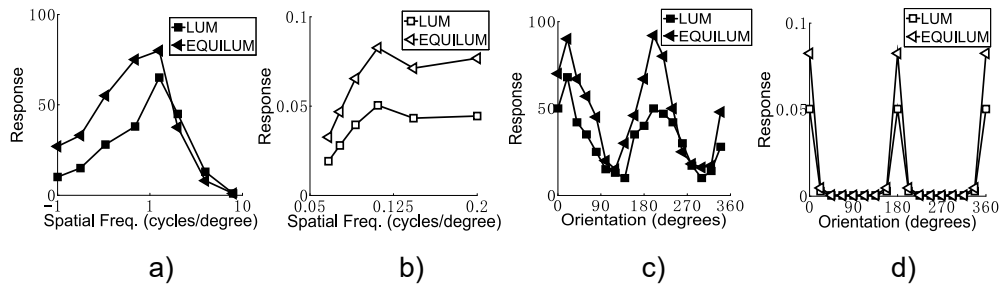


Figure 3-6 Simulation of neurophysiological data [41] from a complex cell in macaque V1.

a) Spatial frequency tuning curves of a complex cell in macaque V1 at the optimal orientation [41].

b) Spatial frequency tuning curves of a complex cell in the colour PC/BC model of V1 at the optimal orientation.

c) Orientation tuning curves of a complex cell in macaque V1 at the optimal spatial frequency [41].

d) Orientation tuning curves of a complex cell in the colour PC/BC model of V1 at the optimal spatial frequency.

3.4.2 Contour Detection Results

The proposed colour PC/BC model of V1 is qualitatively and quantitatively evaluated by the BSDS300 benchmark [36]. With respect to qualitative comparison between the colour PC/BC model of V1 and the original PC/BC model of V1 [39], as shown in Figure 3-7, some human-identified contours, which are marked in red within the four example colour images and their grayscale version from the BSDS300 benchmark [36], cannot be located by the original PC/BC model of V1 [39] because these contours are defined by insignificant intensity discontinuities, but these contours can be easily detected by the colour PC/BC model of V1 because these contours are defined by significant colour discontinuities. Hence the colour PC/BC model of V1 should be able to locate more human-identified contours than the original PC/BC model of V1 [39]. The quantitative evaluation of the colour and original PC/BC model of V1 also produced the same result. As shown in Figure 3-8, the F-score of the colour PC/BC model of V1 (0.67) is significantly better than the original PC/BC model of V1 (0.61). Because the BSDS300 benchmark [36] can also generate the highest F-score for each test image instead of F-score for whole 100 test images as introduced in section 2.4.1, it can be found that the colour PC/BC model of V1 can generate significant higher F-score with increment more than 0.05 (0.10) on 57 (30) test images compared with the original PC/BC model of V1 [39].

The biologically-inspired colour PC/BC model of V1 employs local discontinuities in colour and intensity within an input image to perform boundary detection. Despite only

limited local cues being used in the proposed model, its F-score (0.67) is slightly better than some recently proposed contour detection algorithms, as shown in Figure 3-8. For instance, the Boosted Edge Learning (BEL) algorithm (F-score 0.66) [138], the Sparse Reconstruction-based Discrimination method (SRBD) (F-score 0.66) [48], the Double-Opponency and Spatial Sparseness Constraint Model (DOSSC) (F-score 0.66) [118] and the Probability of boundary (Pb) algorithm (F-score 0.65) [47]. Both the BEL algorithm and the SRBD method require a complicated training procedure and employs at least as many local cues as the proposed model. The DOSSC utilizes the same local cues as the proposed model and has a much lower computational cost. The Pb algorithm employs one more local cue (local discontinuities in texture) compared with the proposed model. The multi-scale Pb [65] algorithm also uses texture cues as well as multiple image scales. Its F-score of 0.68 is slightly higher than the F-score of 0.67 obtained by the colour PC/BC model of V1. Hitherto, the recent near state-of-the-art algorithm gPb-owt-ucm [18] has the F-score of 0.71 which is significantly higher than that of the colour PC/BC model of V1. The gPb-owt-ucm employs local discontinuities in colour, intensity and texture, multiple image scales of these features, and also global information of these features. In contrast, the biologically plausible colour PC/BC model of V1 only employs local discontinuities in colour and intensity.

The value of each parameter in the colour PC/BC model of V1 was determined by trial and error. Specifically, the proposed method was operated on training images with a set of manually defined default parameter values. Afterwards, each parameter was sequentially determined while all other parameter values were kept fixed according to the alteration of F-score on these training images. Compared to the original PC/BC model of V1 [39], the same values of parameters: κ_{LGN} , σ_{V1} , σ_{LGN} , ϵ_1 , and ϵ_2 were used in the colour PC/BC model of V1. The values of other parameters: u , $wt1$, $wt2$, and $wt3$ in the colour PC/BC model of V1 were determined sequentially as previously mentioned. The sensitivity of the results to different parameters was estimated by altering the value of one parameter at a time while other parameters were kept fixed at their optimal values. The sensitivity analysis for parameters: κ_{LGN} , σ_{V1} , σ_{LGN} , ϵ_1 , and ϵ_2 was performed using the original PC/BC model of V1 [39] and the results are reported in [39]. The only new parameters introduced here are: u , $wt1$, $wt2$, and $wt3$. An evaluation of the sensitivity of the results to these parameters is shown in Table 3-1.

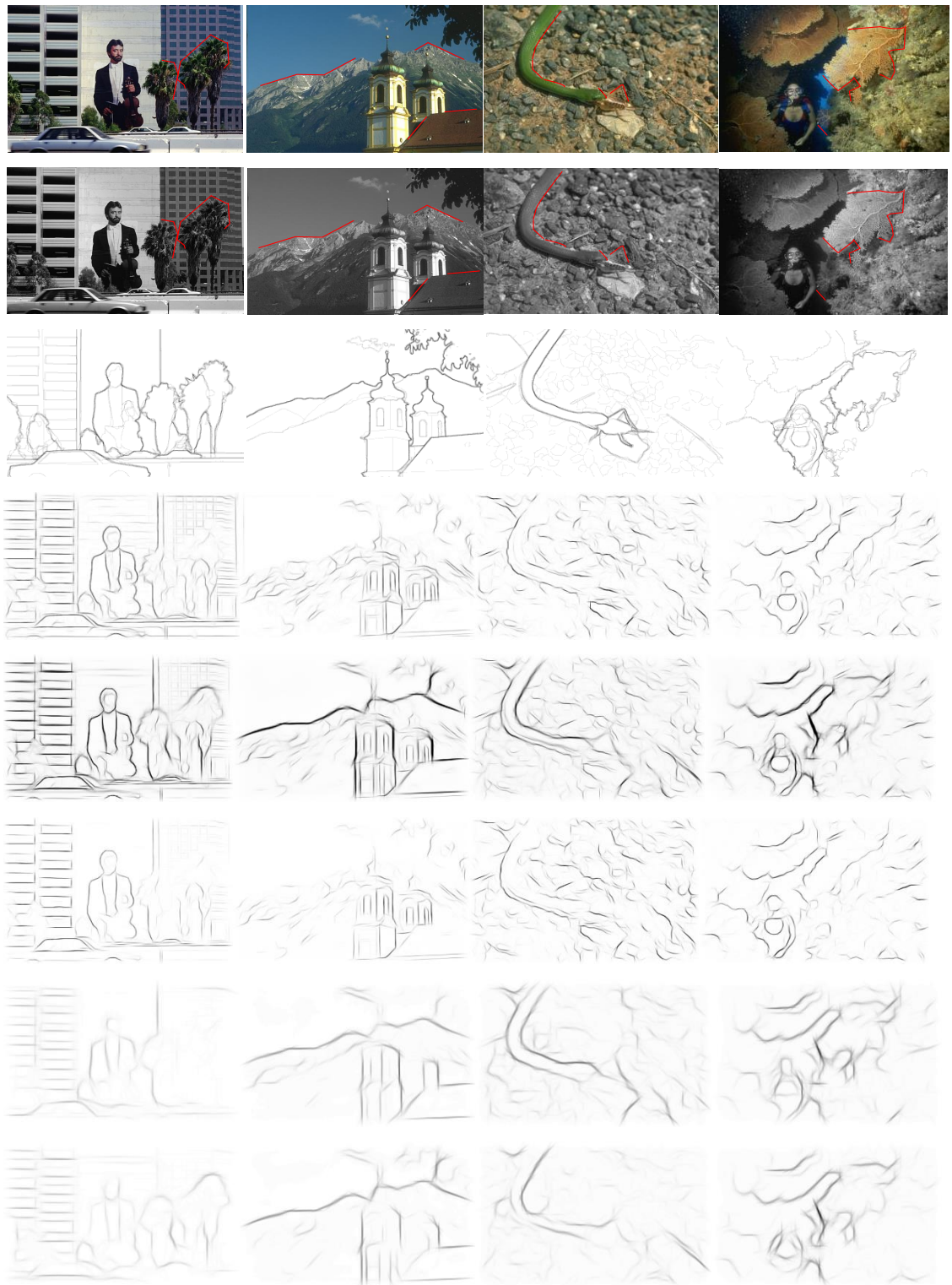


Figure 3-7 Results of contour detections for three colour images with their grayscale version from the BSDS300 dataset [36].

Contour detection results are produced by (from top to bottom): human, the PC/BC model of V1 [39], the colour PC/BC model of V1, and three maps reconstructed from luminance, red-green, and blue-yellow neurons' responses in the colour PC/BC model of V1. Some human-identified contours, which are marked in red in both colour and grayscale images, cannot be detected by the original PC/BC model of V1 [39] but can be located by the colour PC/BC model of V1.

It can be seen that significant changes in three parameters ($wt1$, $wt2$, and $wt3$) have little effect on the contour detection results, whereas the results are more sensitive to the value of parameter u . As shown in Figure 3-7, some of contours detected from each group of neurons in the colour PC/BC model of V1 differ while some overlap, because contours defined by intensity discontinuities may also be defined by the colour discontinuities. The effect of the parameters $wt1$, $wt2$, and $wt3$ is to determine in what proportion the contours reconstructed from three different groups of neurons' responses are mixed to produce the final contour detection result of the proposed model. For instance, if the value of $wt2$ is increased, the proportion of strength of contours reconstructed from the red-green neurons' responses possessed in the final result is also increased. Hence if the value of $wt2$ is much larger than $wt1$ and $wt3$, the final contour detection result should be similar to contours reconstructed from only red-green neurons' responses as shown in the seventh row of Figure 3-7.

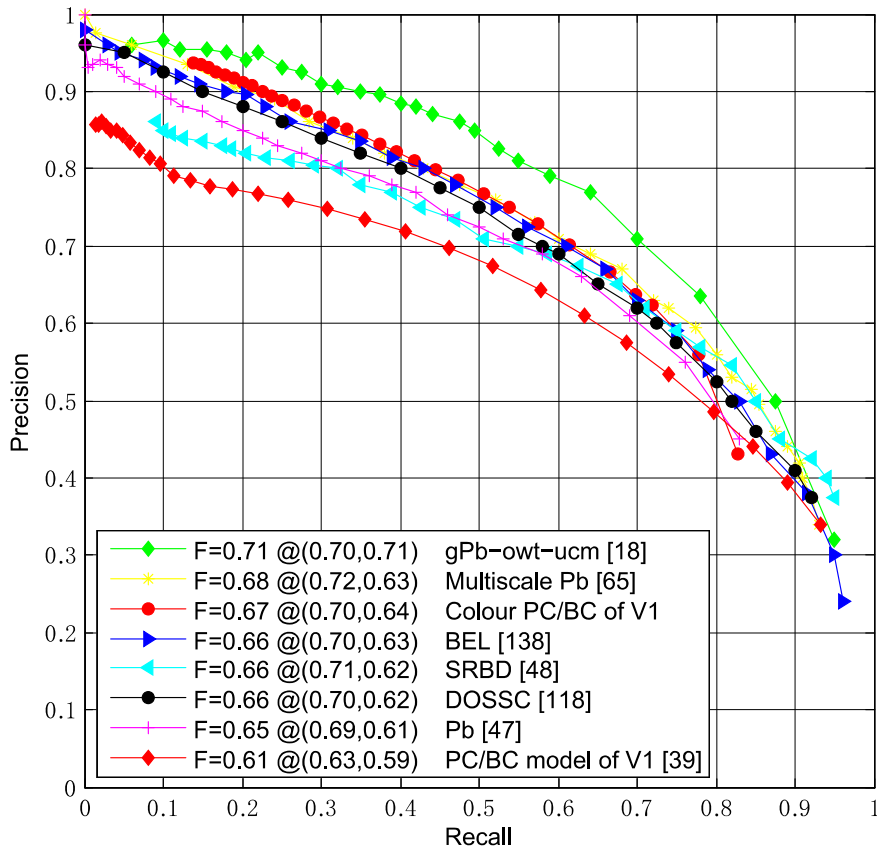


Figure 3-8 Performances of different contour detection algorithms evaluated by the precision-recall curve and the F-score defined in section 2.4.1 from the BSDS300 Benchmark [36].

3.4.3 Failure Cases Discussion

In an additional experiment, the two colour input channels: red-green and blue-yellow

defined in equation 3.1 and 3.2 in the colour PC/BC model of V1 were replaced with the a^* and b^* channels of the CIELAB colour space. The evaluation results indicate that the performance of the colour PC/BC model of V1 utilizing the two new colour channels is slightly worse (F-score 0.66) than using the original two neurophysiologically-inspired colour channels (F-score 0.67). Other colour spaces may also be used in the proposed model but they have not been tested. In another experiment, the two colour input channels defined in section 3.3.2 were kept at full scale instead of being half-scaled as defined in equation 3.4 and 3.5. The evaluation results are also worse (F-score 0.65).

Table 3-1 Effect of altering the values of the four parameters introduced in the colour PC/BC model of V1 on the F-score defined in equation 2.1 from the BSDS300 Benchmark [36].

Parameter	Description	Standard Value	Altered Value	F-score
u	The extent of unbalance in strength for two opponent cone signals	0.8	1 0.6	0.66 0.65
wt1	The weight of the contribution of the output from luminance neurons in the final result.	0.75	0.375 1.5	0.66 0.67
wt2	The weight of the contribution of the output from red-green neurons in the final result.	1.125	0.5625 2.25	0.67 0.66
wt3	The weight of the contribution of the output from blue-yellow neurons in the final result.	1.875	0.9375 3.75	0.67 0.67

3.5 Discussion

In this Chapter, the predictive coding/biased competition model of V1 [39] has been extended to locate boundaries defined by local discontinuities in colour. The result is a significant improvement in the contour detection performance (F-score 0.67) compared with the original PC/BC model of V1 [39] (F-score 0.61). This extension has been inspired by neurophysiological data from single neurons in macaque V1 [40][41]. The resulting model can successfully simulate the response properties of orientation-selective double-opponent neuron in macaque V1 [41]. The biologically-plausible

colour PC/BC model of V1 performs contour detection based on local discontinuities in colour and intensity within an input image. Despite only limited local cues being used in the proposed model, it still slightly outperforms some recently proposed algorithms which use more cues and/or require a complicated training procedure. The good performance of the biologically-plausible colour PC/BC model of V1 indicates it is possible that the human/animal visual system identifies contours defined by colour discontinuities in similar manner to those defined for intensity discontinuities.

Chapter 4 The Predictive Coding/Biased Competition Model of V2

4.1 Introduction

The predictive coding/biased competition (PC/BC) model of V1 [39], which has been introduced in section 2.5, has a good performance of locating boundaries defined by intensity discontinuities. Furthermore, the colour PC/BC model of V1 described in Chapter 3 can also outperform some recently proposed contour detection algorithms.

When the input to the PC/BC model of V1 is a grayscale image, some human-identified contours defined by texture discontinuities (marked in red in Figure 4-1) and some human-identified contours defined by insignificant intensity discontinuities (marked in green in Figure 4-1) can hardly be detected by the PC/BC model of V1 [39]. Hence the PC/BC model of V1 needs to be extended to detect more contours within greyscale images.

Since illusory contours can be interpreted as boundaries defined by texture discontinuities within an image, a PC/BC model of V2 is devised to simulate the illusory contours detection in V2 [42][43] by locating the texture boundaries (red-marked contours shown in Figure 4-1) which cannot be detected by the PC/BC model of V1 [39]. The definition of the PC/BC model of V2 for texture boundary detection is described in section 4.2. Section 4.2 also describes a method of integrating results derived from the PC/BC model of V1 with the PC/BC model of V2 and shows this integrated contour detection results.

The responses of corners in V2 are stimulated by the input of two sides at different orientations [44][45][46]. Therefore, the feedback of corner responses can possibly be used to enhance both sides forming a corner. Based on this hypothesis, another version of the PC/BC model of V2 is devised to detect corners at different orientations and utilize the feedback of prediction neuron responses in V2 to strengthen the prediction neuron responses in V1 which can comprise a corner. It can be anticipated that some weak contours defined by insignificant discontinuities in intensity (green-marked contours shown in Figure 4-1) that can hardly be detected by the PC/BC model of V1 [39] may be enhanced by the feedback from this PC/BC model of V2. The definition of

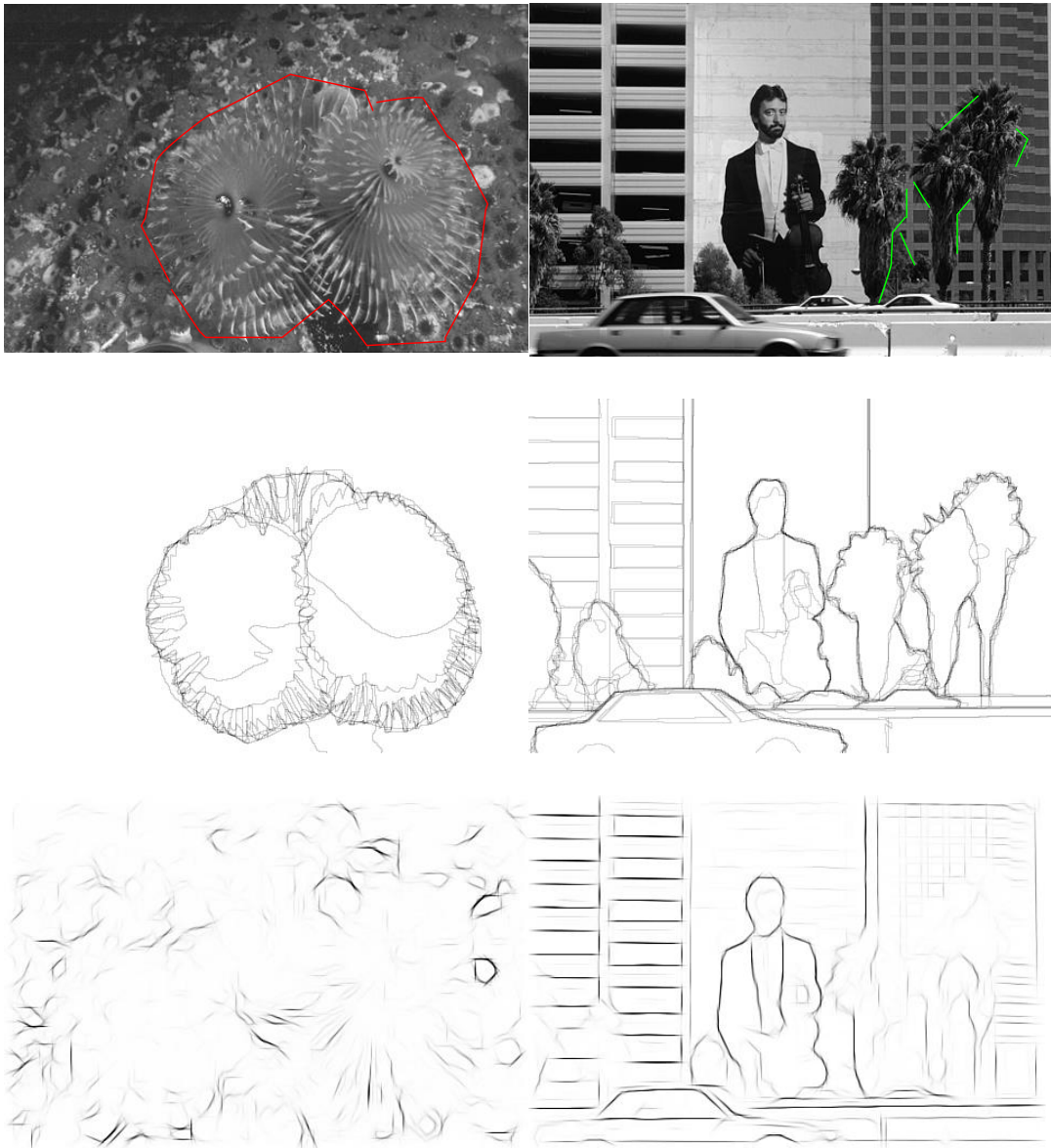


Figure 4-1 Two natural images from the BSDS300 benchmark [36] with the ground truth contour maps and the maps derived from the PC/BC model of V1 [39]. Some human-identified contours, which are marked in red, are defined by texture discontinuities. Some human-identified contours, which are marked in green, are defined by insignificant intensity discontinuities. These human-identified contours, which are marked in red and green, can hardly be detected by the PC/BC model of V1 [39].

the PC/BC model of V2 for corner detection is described in section 4.3. The results obtained from the PC/BC model of V1 integrated with this PC/BC model of V2 are also shown in section 4.3. Section 4.4 discusses some possible reasons why each of the integrated PC/BC model fails to improve the performance of the original PC/BC model of V1 [39] for contour detection.

4.2 The PC/BC model of V2 for texture boundary detection

4.2.1 Methods

As introduced in section 2.4 of Chapter 2, the PC/BC model of V1 [39] defines a set of prediction neurons to represent texture within an input image. The responses of these prediction neurons are ignored in its final contour detection result, because they probably are not edges that would be identified as contours by humans. Although texture is unlikely to be real contours within an image, the discontinuities in texture are usually identified as true contours. Therefore, the PC/BC model of V2 is devised to locate texture discontinuities starting from processing the responses of those prediction neurons representing texture in the PC/BC model of V1 [39]. The PC/BC model of V2 for texture boundary detection integrated with the PC/BC model of V1 [39] is illustrated in Figure 4-2.

Because both the PC/BC model of V1 and the PC/BC model of V2 are implemented by using equation 2.3 and equation 2.4, the terms ‘V1 equation 2.3’, ‘V1 equation 2.4’, ‘V2 equation 2.3’ and ‘V2 equation 2.4’ are used to prevent confusion.

The procedure of the PC/BC model of V2 combined with the PC/BC model of V1 is shown in Figure 4-2. Firstly, the PC/BC model of V1 is operated independently as defined in the section 2.5 of Chapter 2. V1 equation 2.3 and 2.4 are iteratively updated 30 times to generate the responses of two sets of prediction neurons from Y_1^{V1} to Y_{16}^{V1} and Y_{17}^{V1} to Y_{32}^{V1} representing real contours and texture respectively. Only the 16 classes of the prediction neurons (Y_{17}^{V1} to Y_{32}^{V1}) that represent texture are further processed as the input to the PC/BC model of V2 to locate boundaries defined by texture discontinuities. Since the receptive fields of the 16 classes of the prediction neurons are comprised by the first order derivative of Gaussians at two anti-phases for each of eight orientations, the responses of each two classes of prediction neurons whose receptive fields are at the same orientation but in anti-phase are added together first. Hence, the 16 classes of prediction neurons are combined into the 8 classes of prediction neurons. Afterwards, the 8 classes of prediction neurons are further combined into 4 classes using the following method. Firstly, at each orientation the three classes of prediction neurons whose receptive fields are at neighbouring orientations are aggregated to generate eight classes of aggregated prediction neurons. For instance, the responses of prediction neurons whose receptive fields are at 0 degrees are added with the responses of the

prediction neurons whose receptive fields are at 22.5 degrees and 157.5 degrees. Secondly, only 4 classes of the aggregated prediction neurons are kept, these are the ones which are aggregated from the three classes of prediction neurons whose receptive fields are at the orientations (157.5, 0, 22.5), (22.5, 45, 67.5), (67.5, 90, 112.5), and (112.5, 135, 157.5) degrees. Thirdly, the four classes of aggregated prediction neuron responses are down-sampled by a factor 2. The procedure to combine responses across

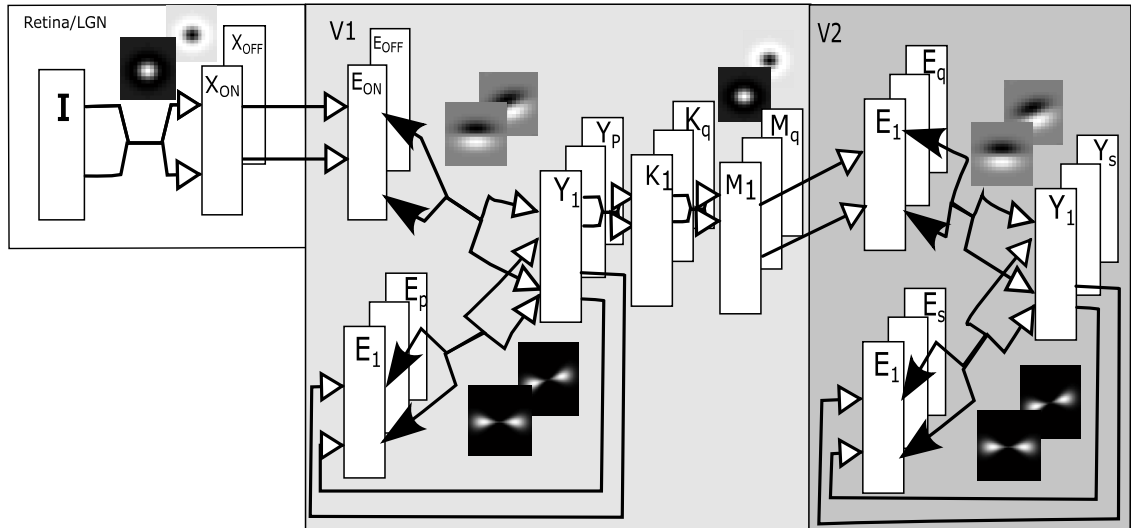


Figure 4-2 The diagram of the PC/BC model of V2 for texture boundary integrated with the PC/BC model of V1 [39].

White rectangles represent the neuron populations. The interconnections between different neuron populations are indicated using arrows. Open (filled) arrows represent feedforward excitatory (feedback inhibition) connections. Many-to-many connectivity between two different neuron populations are indicated by crossed connections. Parallel connections signify a one-to-one mapping between neurons in two populations. The boxes neighbouring crossed connections illustrate the type of weight kernels utilized. Specifically, the LGN model employed Laplacian of Gaussian kernels, and the first order derivative of Gaussian kernels and weight kernels of lateral connectivity defined in equation 2.5 and 2.6 are utilized in both the V1 model and V2 model. The same kernels are used in both feedforward and feedback connections, but scaled differently [39]. The symbols I , X , E , Y , K , M are consistent with symbols defined in V1 and V2 equation 2.2, 2.3, 2.4, 2.5 and equation 4.1.

phases, orientations and spatial locations (through down-sampling) can be interpreted as calculating the responses of complex cells in V1. These combined and down-sampled prediction neuron responses are denoted by K_q .

In computer vision, the procedure of locating intensity or texture boundaries is computationally similar. Typically, both first utilize pixels to represent the structure of

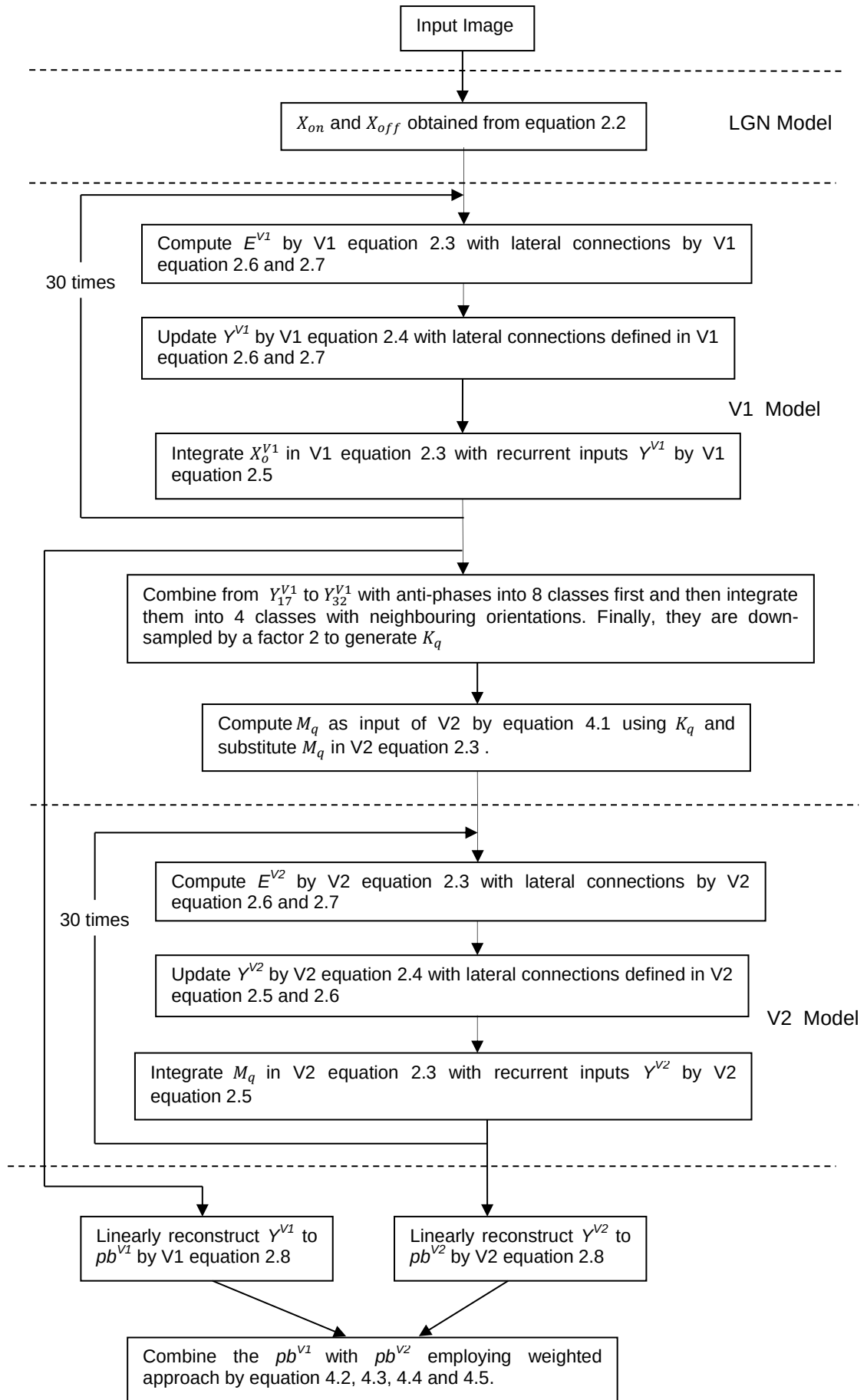


Figure 4-3 The procedure of the PC/BC model of V2 for texture boundary detection combined with the PC/BC model of V1.

the intensity or texture and then employ a kind of linear filter to locate their discontinuities. The same method used to locate intensity discontinuities in the V1 model was used here to locate discontinuities in the V2 model. Each of the $q=4$ classes of aggregated and down-sampled prediction neuron responses (K_q) are also pre-processed by convolution with a Laplacian of Gaussian filter (l) before input to the PC/BC model of V2, as defined by:

$$M_q = \tanh\{\kappa_{V1}(K_q * l)\} \text{ for } q = 1 \dots 4 \quad (4.1)$$

The values of all parameters in equation 4.1 are to the same as parameters defined in equation 2.1. The value of κ_{V1} is equal to κ_{LGN} used in equation 2.1. However, the positive and negative responses in M_q are not separated and directly input to the PC/BC model of V2.

Afterwards, the PC/BC model of V2 is operated independently by substituting M_q into V2 equation 2.3 and iteratively updating V2 equations 2.3 and 2.4 30 times. The synaptic weights v_{qk} and w_{qk} for prediction neurons in the PC/BC model of V2 are also defined by the first order derivative of Gaussian at eight orientations as defined in equation 3.7, 3.8 and 3.9. The values of all parameters regarding the definitions of the synaptic weights for the prediction neurons in V2 are also set equivalently to those used in the V1 model. As previously mentioned, the positive and negative responses in the input channels M_q of the PC/BC model of V2 are not separated, therefore the positive and negative values of the first order derivative of Gaussian are also not separated. Hence, there are overall 8 classes of prediction neurons for V2.

Since some of the discontinuities in texture are also not true contours (i.e. texture boundaries) but still texture edges itself, the PC/BC model of V2 also defined another 8 classes of prediction neurons to represent texture edges. Afterwards, the weight kernels of the lateral connections being identical to the PC/BC model of V1 are also defined to differentiate the two groups of prediction neurons in V2 to operate as texture edge detector and texture boundary detector. Finally, the result of the PC/BC model of V2 are also obtained by using V2 equation 2.8 to generate the probability of texture boundary at each pixel within an image pb^{V2} which are later integrated with the probability of boundary pb^{V1} derived from the PC/BC model of V1.

To generate the overall contour detection result from the integrated PC/BC model, a

straightforward method is to combine pb^{V1} with pb^{V2} by directly adding them together. However, some of the texture boundaries in pb^{V2} and the corresponding intensity boundaries in pb^{V1} are spatially offset which will make the combined contour detection results fuzzy. To avoid fuzzy boundaries, the non-maximum suppression method [10] needs to be applied to the sum of pb^{V1} and pb^{V2} . The non-maximum suppression method [10] can be used to thin fuzzy boundaries along an orientation at each pixel of a boundary map. If such boundary map is comprised by a set of oriented responses, this orientation is defined as the orientation of the maximum responses across different orientations at each pixel, hence, we call this orientation as the ‘maximum orientation’. To apply the non-maximum suppression method [10] to the boundary map of the sum of pb^{V1} and pb^{V2} , that ‘maximum orientation’ needs to be defined at each pixel first.

Since both pb^{V1} and pb^{V2} are computed by the sum of sixteen ($8 \text{ orientations} \times 2 \text{ anti-phases}$) oriented responses as defined in equation 2.7, the boundary map of the sum of pb^{V1} and pb^{V2} is comprised by two sets of oriented responses. These two sets of oriented responses can be combined into one set of oriented responses by summing together the oriented responses from different sets at the same orientation. Afterwards, the ‘maximum orientation’ can be directly defined at each pixel in the sum of pb^{V1} and pb^{V2} . However, some texture boundaries and intensity boundaries are spatially offset but at different orientations, which means the ‘maximum orientation’ defined above between neighbouring pixels can be dramatically different. Hence, applying the non-maximum suppression method using this ‘maximum orientation’ can generate some non-smooth contours in the final result.

To generate a better overall contour detection result from the integrated PC/BC, the following method is utilized to define a new ‘maximum orientation’ in the sum of pb^{V1} and pb^{V2} . First, θ^{V1} and θ^{V2} are the ‘maximum orientation’ defined in the boundary map of the pb^{V1} and pb^{V2} respectively. Then they are processed to generate the ‘maximum orientation’ in the sum of pb^{V1} and pb^{V2} by following equations:

$$WTV1 = \frac{pb^{V1}}{pb^{V1} + pb^{V2}} \quad (4.2)$$

$$WTV2 = \frac{pb^{V2}}{pb^{V1} + pb^{V2}} \quad (4.3)$$

$$\theta^{V1V2} \approx WTV1 \otimes \theta^{V1} + WTV2 \otimes \theta^{V2} \quad (4.4)$$

$$pb^{V1V2} = \text{nonmax}_{\theta^{V1V2}}\{pb^{V1} + pb^{V2}\} \quad (4.5)$$

where $\otimes$ represents element-wise multiplication. The ‘maximum orientation’ θ^{V1} and θ^{V2} at each pixel are combined using the weight values defined in equations 4.2 and 4.3. The combined orientations are approximated to the nearest orientation employed in θ^{V1} and θ^{V2} to define the ‘maximum orientation’ θ^{V1V2} in the sum of pb^{V1} and pb^{V2} at each pixel as defined in equation 4.4. Last but not the least, the final contour detection result pb^{V1V2} derived from the PC/BC model of V1 [39] integrated with the PC/BC model of V2 is generated by applying the non-maximum suppression method [10] using the ‘maximum orientation’ θ^{V1V2} as defined in equation 4.5.

4.2.2 Results

The contour detection result derived from the PC/BC model of V1 [39] integrated with the PC/BC model of V2 for texture boundary detection is qualitatively and quantitatively evaluated by both the BSDS300 benchmark [36] and the Prague texture segmentation benchmark [37]. The value of each parameter in the integrated PC/BC model of V1 for texture boundary was determined by trial and error. Specifically, the integrated PC/BC model was operated on training images from the BSDS300 benchmark [36] with a set of manually defined default parameter values. Afterwards, the sensitivity of the F-score to the exact value of each parameter was tested by sequentially changing the value of each parameter while all other parameter values were kept fixed. All parameters’ values defined in the original PC/BC model of V1 [39] were not changed.

The qualitative evaluation results indicated that some human-identified texture boundaries (marked in red in Figure 4-4) that cannot be detected by the original PC/BC model of V1 [39] can be located by the integrated PC/BC model for texture boundary, as shown in Figure 4-4. However, there are also many texture boundaries that cannot be or are inaccurately located by the integrated PC/BC model, an example is marked by the yellow rectangle in Figure 4-4. The quantitative evaluation results indicate that the performance of the integrated PC/BC model for texture boundary (F-score 0.60) was slightly worse than the original PC/BC model of V1 (F-score 0.61) [39]. However, the employment of the proposed integrated method defined in equation 4.2 to 4.5 instead of straightforwardly combining pb^{V1} and pb^{V2} can increase the performance of integrated PC/BC model from F-score 0.57 to 0.61. To further evaluate the integrated PC/BC model, the Prague texture segmentation [37] was also used. The qualitative results illustrated that synthetic texture boundaries (marked in red in Figure 4-5) that cannot be

located by the original PC/BC model of V1 can be located by integrated PC/BC model for texture boundary, as shown in Figure 4-5. Because both the PC/BC model of V1 and integrated PC/BC model were devised to locate contours within natural images, these two methods can locate many additional contours within each part of a synthetic texture image as well as synthetic texture boundaries. This, results in the poor performance of these two methods when quantitatively evaluated on the synthetic texture images using the F-score. But the quantitative evaluation results also indicate that the integrated PC/BC model (F-score 0.26) can generate better performance for locating synthetic texture boundaries than the original PC/BC model of V1 (F-score 0.22).

As previously mentioned, the quantitatively evaluation result from the BSDS300 benchmark [36] indicates that the performance of the proposed integrated PC/BC model for texture boundary detection is worse than the original PC/BC model of V1 [39]. In other words, the texture boundaries located by the PC/BC model of V2 are not good enough to improve the contour detection result obtained from the integrated PC/BC model. To analyse the deficiency of the integrated PC/BC model, the texture gradient method used in the probability of boundary (Pb) algorithm [47] is employed for comparison in this section.

To locate good texture boundaries, an algorithm needs to represent texture information efficiently and accurately within an image first, and then locate discontinuities in texture based on the texture information which has been represented. Therefore, the intermediately generated map from the integrated PC/BC and the Pb texture gradient which are utilized to represent texture information within the same three images from the BSDS300 dataset [36] are shown in Figure 4-5 for comparison. Specifically, the intermediately generated map from the integrated PC/BC is the sum of the $q=4$ classes of M_q which are the input to the PC/BC model of V2. The intermediately generated map from the texture gradient method used in the Pb algorithm [47] is the map of the filtered responses derived from the convolution of a set of derivative of Gaussian filters with an input image, which are further processed by the K-Means clustering algorithm to generate textons.

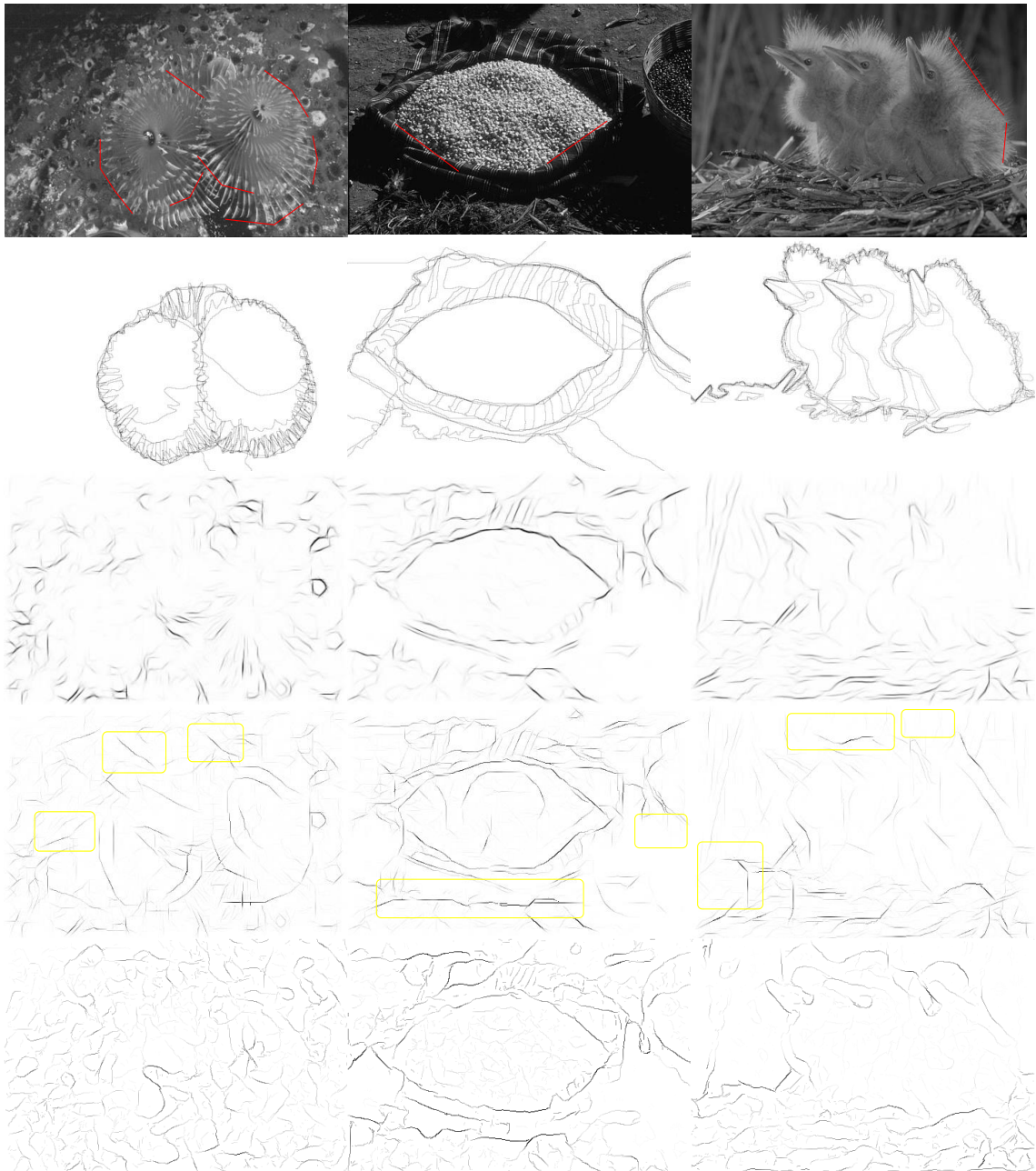


Figure 4-4 Results of contour detections for three images are from BSDS300 [36]. Contour detection results are produced by (from top to bottom): human, the PC/BC model of V1 [39], the integrated PC/BC model, and the texture gradient method used in the Pb algorithm [47]. Some human-identified contours, which are marked in red, are texture boundaries that cannot be detected by the original PC/BC model of V1 [39] but can be detected by the integrated PC/BC model. Some human-identified texture boundaries, which are marked by yellow rectangle, are missed by the integrated PC/BC model or are inaccurately located by the integrated PC/BC model.

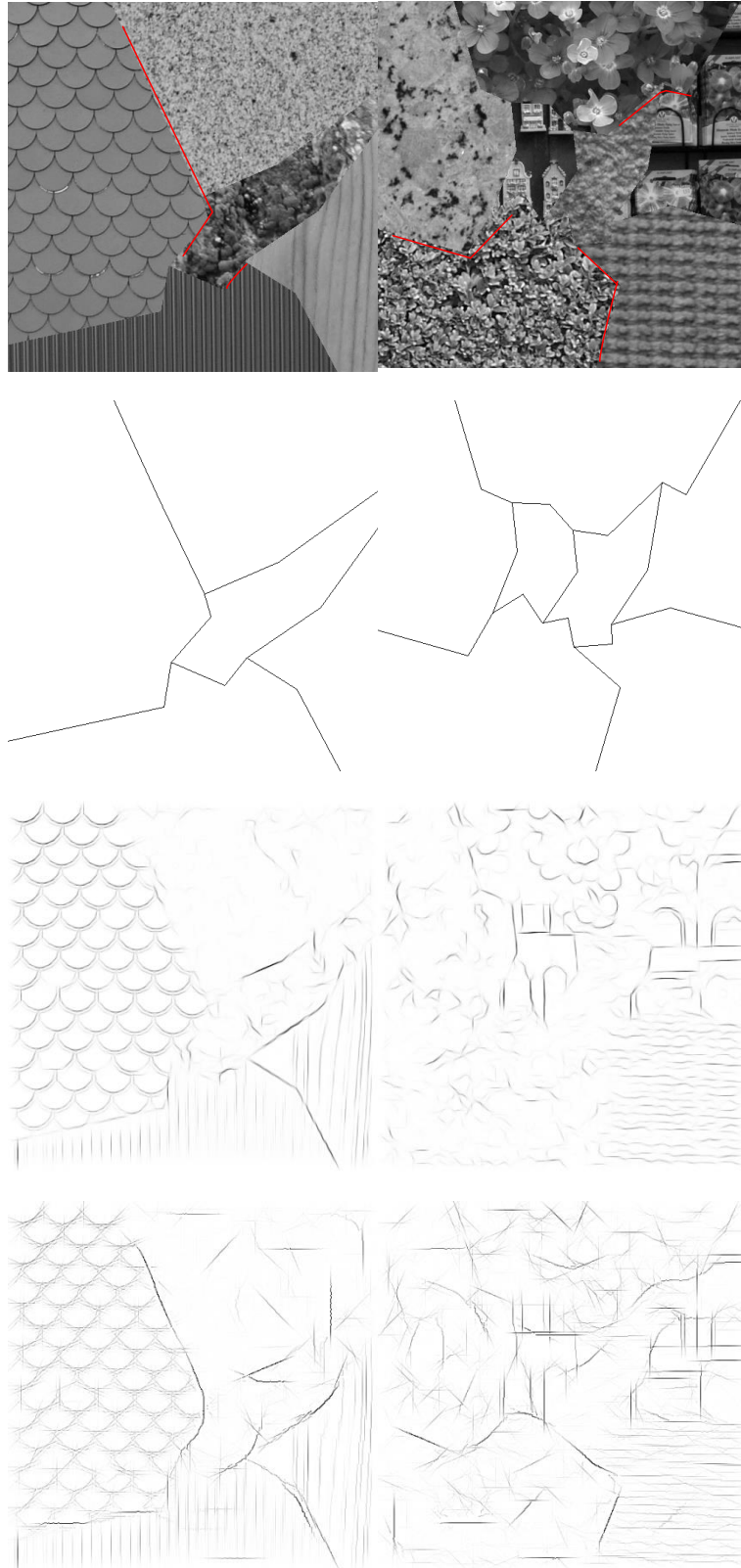


Figure 4-5 Results of contour detections for three images are from Prague benchmark [37].

Contour detection results are produced by (from top to bottom): ground truth, the PC/BC model of V1 [39], and the integrated PC/BC model. Some ground truth contours, which are marked in red, are texture boundaries that cannot be detected by the original PC/BC model of V1 [39] but can be detected by the integrated PC/BC model.

As shown in Figure 4-4, the integrated PC/BC model is able to locate some boundaries defined by texture discontinuities which cannot be located by the original PC/BC model of V1 [39]. However, comparing with the texture gradient method utilized in the Pb algorithm [47], the texture boundaries located by the integrated PC/BC are neither comprehensive nor accurate. In other words, there are still many texture boundaries that cannot be located by the integrated PC/BC model and some texture boundaries located by the integrated PC/BC are not located by both the Pb texture gradient [47] and human observers. Hence the worse performance of the integrated PC/BC model is due to both the deficiency of good texture boundary and the impairment by inaccurate texture boundary. The root cause of this problem is possible that the texture information represented by the intermediately generated map from the integrated PC/BC is insufficient and inaccurate compared to the texture information represented by the intermediately generated map from the Pb texture gradient [47] as indicated in Figure 4-6. The ‘insufficiency’ can be indicated by the fact that some texture information in the intermediately generated maps cannot be found from the integrated PC/BC model but can be found from the Pb texture gradient method [47], such as the stripped textile and basket in the second image, and some feathers of birds in the third image, as shown in Figure 4-6. The ‘inaccuracy’ can be indicated by that some texture information in the intermediately generated maps from the integrated PC/BC model are not represented accurately especially at the location of texture boundaries, such as the boundaries of grass in the second image and the boundaries of nest in the third image, as shown in Figure 4-6. It is reasonable that an algorithm can hardly locate good texture boundaries from an image map that represents insufficient and inaccurate texture information.

As previously mentioned, the PC/BC model of V1 can be interpreted as a sparse coding model, which utilizes sparse representations (prediction neuron responses Y) to encode an image employing a set of redundant basis or dictionary (first order derivative of Gaussian filters). Texture information within an image represented by the sparse representation of this image is possibly not sufficient and accurate enough, because some texture information may be represented by dictionary and missed in the sparse representation. The procedure of the transformation made from Y_p to K_q including down-sampling and combination of phases and orientations is to make the texture information input to the PC/BC model of V2 as sufficient and accurate as possible. However, as shown in Figure 4-6, the texture information represented in the current

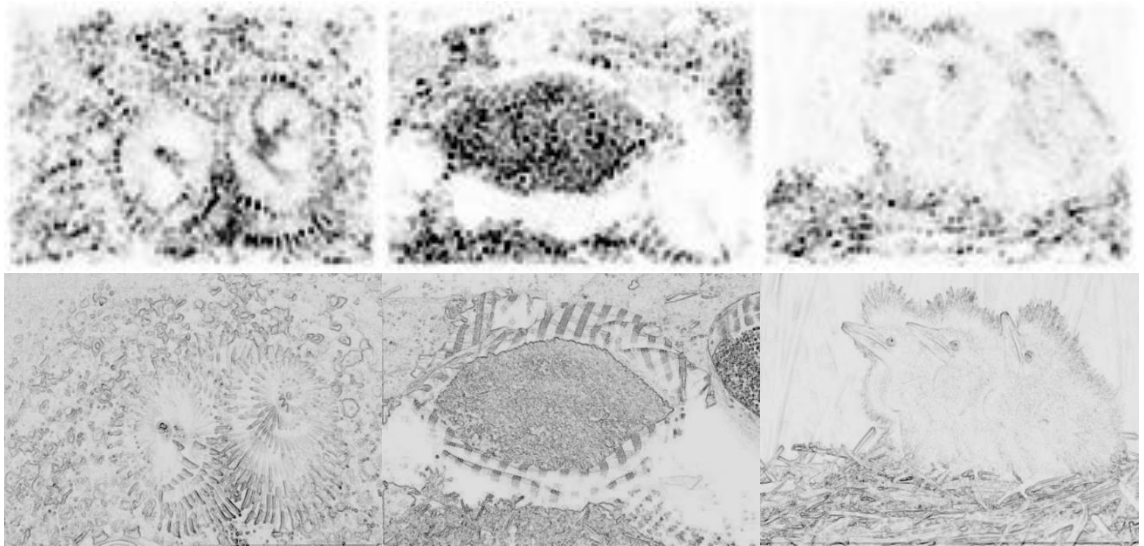


Figure 4-6 The intermediately generated maps generated from the integrated PC/BC model (top row) and the Pb texture gradient [47] (bottom row).

These maps are used to represent texture information within the same three images from the BSDS300 dataset [36]. The texture information represented in the Pb method is more accurate (i.e. boundaries of grass in the second image and the boundaries of nest in the third image) and comprehensive (i.e. the stripped textile and basket in the second image and some feathers of birds in the third image) than the integrated PC/BC model.

integrated PC/BC model is still not good enough. To improve the sufficiency and accuracy of texture information represented in the integrated PC/BC model, a possible solution is to define the receptive fields of the V1 neurons in multiple sizes to represent both boundary edge and texture edge first. The PC/BC model of V1 is then operated independently to generate the responses of the prediction neurons whose receptive fields are in different sizes. Afterwards, the prediction neuron responses that represent texture edges generated in each operation can be integrated to possibly represent texture information more sufficiently and accurately, because the integration of different sparse representations of an image based on different sizes of dictionaries probably have more texture information than the sparse representations based on single dictionary.

4.3 The PC/BC model of V2 for corner detection

4.3.1 Methods

The aim of devising this PC/BC model of V2 is to utilize the feedback of prediction neuron responses representing corners in the PC/BC model of V2 to enhance the weak prediction neuron responses representing boundaries defined by insignificant intensity discontinuities in the PC/BC model of V1 [39]. Therefore, the PC/BC model of V1 and the PC/BC model of V2 need to be operated iteratively, as illustrated in Figure 4-7. The

procedure of the PC/BC model of V2 for corner detections integrated with the PC/BC model of V1 is shown in Figure 4-8.

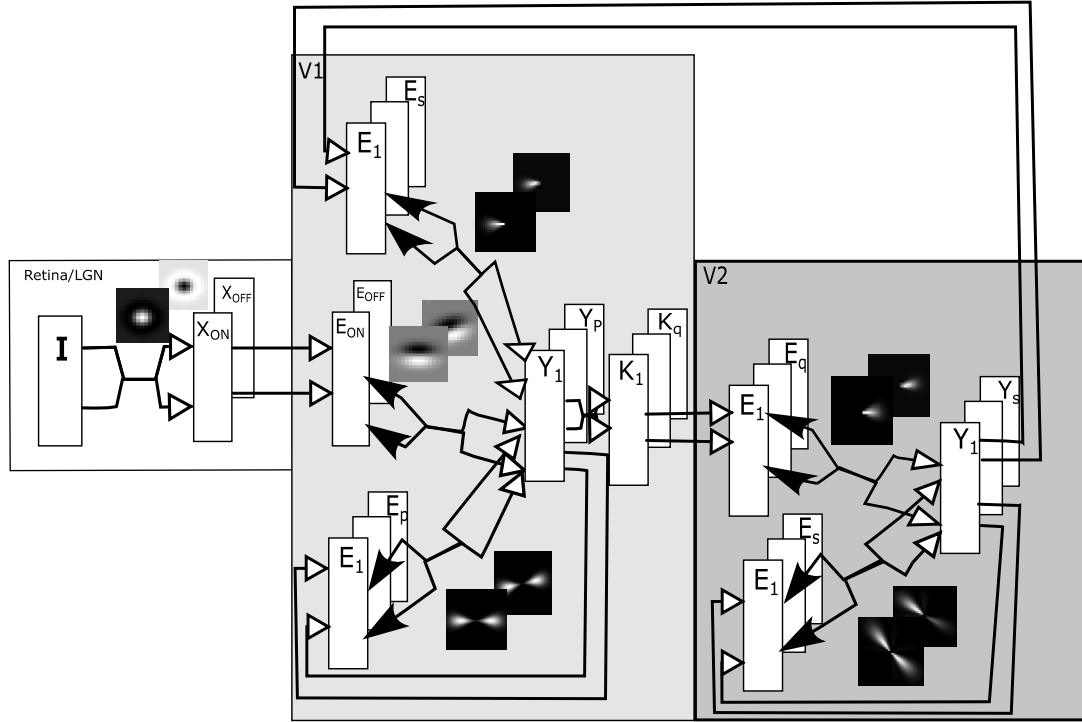


Figure 4-7 The diagram of the PC/BC model of V2 for corner detection integrated with the PC/BC model of V1 [39].

White rectangles represent the neuron populations. The interconnections between different neuron populations are indicated using arrows. Open (filled) arrows represent feedforward excitatory (feedback inhibition) connections. Many-to-many connectivity between two different neuron populations are indicated by crossed connections. Parallel connections signify a one-to-one mapping between neurons in two populations. The boxes neighbouring crossed connections illustrate the type of weight kernels utilized. Specifically, the LGN model employed Laplacian of Gaussian kernels, and the first order derivative of Gaussian kernels and weight kernels of lateral connectivity defined in equation 2.5 and 2.6 are utilized in the V1 model. The same kernels are used in both feedforward and feedback connections, but scaled differently [39]. The symbols I , X , E , Y , K are consistent with symbols defined in V1 and V2 equation 2.2, 2.3, 2.4, 2.5 and equation 4.6.

Because both the PC/BC model of V1 and the PC/BC model of V2 are implemented by using equation 2.3 and equation 2.4, the terms ‘V1 equation 2.3’, ‘V1 equation 2.4’, ‘V2 equation 2.3’ and ‘V2 equation 2.4’ are used to prevent confusion.

In the first iteration of the integrated PC/BC model, the $k=32$ classes of V1 prediction neuron responses Y_k^{V1} are set to zero in V1 equation 2.3. The input X_{k+2}^{V1} is the responses X_o in two channels obtained from equation 2.2 integrated with Y_k^{V1} utilized for lateral connectivity. Afterwards, V1 equation 2.4 is used to update Y_k^{V1} . The receptive fields of each 16 classes of the V1 prediction neurons that represent boundary edges or texture

edges are comprised by the first order derivative of Gaussian filters at two anti-phases for each of eight orientations. The prediction neuron responses representing boundary edges or texture edges whose receptive fields are at the same orientation but two anti-phases are added together first. Hence the $k=32$ classes of prediction neurons Y_k^{V1} representing texture and boundary edges are combined into overall $q=16$ classes of prediction neurons Y_q^{V1C} , which can be interpreted as the neuron activities produced by a set of complex cells in V1. These 16 classes of prediction neurons are pre-processed by a multiplicative gain ($\kappa_{V1} = 2\pi$) before input to the PC/BC model of V2, as defined by:

$$K_q = \tanh\{\kappa_{V1}(Y_q^{V1C})\} \text{ for } q = 1 \dots 16 \quad (4.6)$$

Afterwards, the PC/BC model of V2 is operated. Firstly, s classes of V2 prediction neuron responses Y_s^{V2} representing boundary corners and texture corners are set to zero in V2 equation 2.3. The input K_q to the PC/BC model of V2 is the responses (16 channels) representing boundary and texture obtained from the PC/BC model of V1, together with Y_s^{V2} for lateral connectivity in V2. Then Y_s^{V2} is updated by using V2 equation 2.4. The definitions of weight kernels v_{qs} and w_{qs} in V2 equation 2.3 and V2 equation 2.4 for the V2 prediction neurons Y_s^{V2} and the lateral connectivity in V2 are described later. Before utilizing the V2 prediction neurons to provide feedback to the V1 prediction neurons, Y_s^{V2} is also pre-processed by a multiplicative gain ($\kappa_{V2} = 2\pi$), as defined by:

$$X_s^{V1} = \tanh\{\kappa_{V2}(Y_s^{V2})\} \quad (4.7)$$

From the second iteration of the integrated PC/BC model, X_s^{V1} is integrated with the X_{k+2}^{V1} as the new input substituted into V1 equation 2.3. Y_k^{V1} substituted into V1 equation 2.3 are calculated by V1 equation 2.4 in the last iteration. Y_k^{V1} is then updated by V1 equation 2.4 in this iteration. The weight kernels corresponding to the new input channels X_s^{V1} are defined equivalently to the weight kernels of the V2 prediction neurons employed in V2 equation 2.3 and V2 equation 2.4 except they are rotated by 180 degree to ensure the correct correspondence between the feedforward in V2 and feedback in V1. Afterwards, the Y_k^{V1} is combined into the Y_q^{V1C} and then transformed to K_q as the input substituted into V2 equation 2.3. Y_s^{V2} substituted into V2 equation 2.3 are calculated by V2 equation 2.4 in the last iteration. Y_s^{V2} is then updated by V2 equation 2.4 in this iteration, which are later transformed to X_s^{V1} integrated with the X_{k+2}^{V1} as the input substituted into V1 equation 2.3. The PC/BC model of V1 integrated with the PC/BC model of V2 is then operated sequentially and iteratively for 60 times.

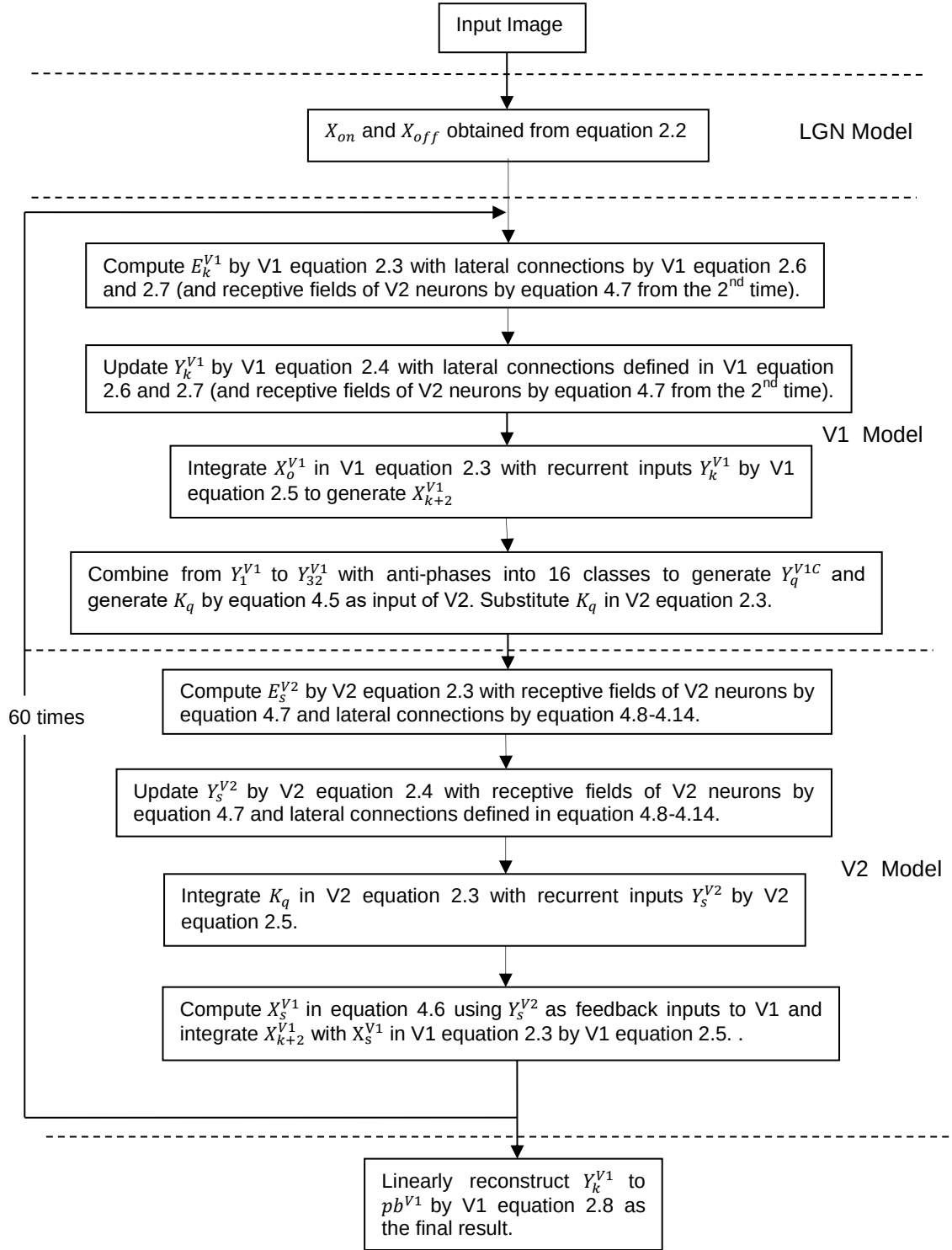


Figure 4-8 The procedure of the PC/BC model of V2 for corner detection combined with the PC/BC model of V1

The weight kernels w_{qs} of the prediction neurons in the PC/BC model of V2 is utilized to locate corners within the sixteen input maps transformed from prediction neuron responses representing boundary and texture at eight orientations in the PC/BC model of V1 [39]. The corners to be located are defined in the range $[22.5^\circ, 180^\circ]$ in steps of $\sigma_c = 22.5^\circ$. Hence there are overall eight kinds of corners for the eight angles being defined. Furthermore, each kind of the corner is rotated in the range $[0^\circ, 360^\circ)$ in

steps of $\sigma_c = 22.5^\circ$ to represent corners of the same angle but at 16 orientations. However, the 180 degrees corner only need to be represented at 8 orientations. Therefore, there are 120 (8 kinds of corners $\times$ 16 orientations – 8 redundant orientations) classes of the prediction neurons to represent corners. Since V1 prediction neuron responses representing boundary and texture are both input to the PC/BC model of V2, two identical sets of V2 prediction neurons need to be defined to represent boundary corners and texture corners respectively by receiving the input from the corresponding eight input maps at eight orientations. Hence there are overall $s=240$ classes of the prediction neurons defined in the PC/BC model of V2.

The weight kernel of the V2 prediction neuron b that represents a corner comprised by two sides at orientations θ_i and θ_j and receives the input of V1 prediction neurons a from the k^{th} input map at orientation θ_k is given by:

$$W_{\theta_i\theta_j\theta_k}^{ab} = \exp \left\{ -\frac{(d - 1/2\sigma_D)^2}{2\sigma_D^2} - \frac{\min_{co=i,j} \psi_{\theta_{co}}^2}{2\sigma_C^2} - \frac{\psi_{\theta_k}^2}{2\sigma_A^2} \right\} \quad (4.8)$$

where d is the Euclidean distance between the coordinates of points (neurons) a and b ; ψ_θ is the difference in angle between the line subtended by points a and b and a line co-linear with the orientation θ ; the σ values are parameters that control the shape of the receptive fields of the prediction neurons in the PC/BC model of V2. The definition of the receptive fields of V2 prediction neurons results in a V2 prediction neuron b being preferably stimulated by the input of V1 prediction neuron a if the orientation of a is roughly co-linear with either of the orientations of the two sides of the corner represented by b . Some receptive fields of V2 prediction neurons representing corners of different angles but at one orientation are illustrated in Figure 4-9. In this V2 model, σ_D , σ_C , σ_A are set to 4, 15, 20 respectively. All feedback weights $V_{\theta_i\theta_j\theta_k}^{ab}$ are defined to be identical to the corresponding $W_{\theta_i\theta_j\theta_k}^{ab}$, but are normalized so that the maximum weight is equal to one.

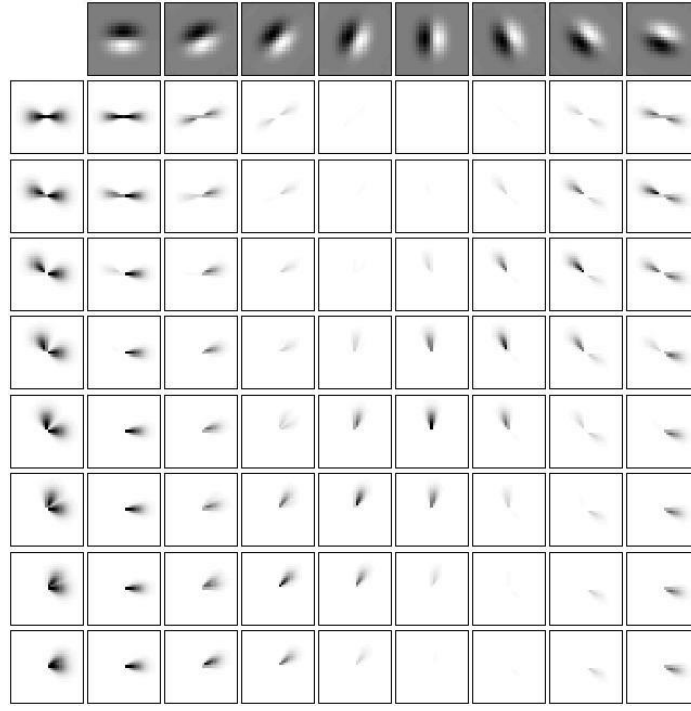


Figure 4-9 Some receptive fields of the prediction neurons defined in the PC/BC model of V2.

The left-hand column shows the aggregated receptive field of the V2 prediction neurons. The top row shows the receptive field of the V1 prediction neurons whose responses provide the eight input maps to the V2 model. Each other element shows the spatial distribution of the receptive field of the V2 prediction neurons.

Two types of the lateral connectivity are defined in the PC/BC model of V2 to increase the differentiation between boundary corners and texture corners represented by the two identical sets of V2 prediction neurons. These lateral connections are defined between pre- and post-synaptic prediction neurons that both represent boundary corners (boundary-to-boundary corner connections), and between pre- and post-synaptic prediction neurons that both represent texture corners (texture-to-texture corner connections). The weight kernel of boundary-to-boundary corner lateral connection between the V2 post-synaptic prediction neuron e that represents a boundary corner comprised of two sides at orientations θ_l and θ_m and the V2 pre-synaptic prediction neuron f with orientations θ_n and θ_o is given by:

$$W_{\theta_l \theta_n}^{ef} = S \exp \left\{ -\frac{(d - 2\sigma_E)^2}{2\sigma_E^2} - \frac{(\theta_n - \theta_l)^2}{2\sigma_F^2} - \frac{\psi_{\theta_l}^2}{2\sigma_G^2} \right\} \quad (4.9)$$

$$W_{\theta_m \theta_o}^{ef} = S \exp \left\{ -\frac{(d - 2\sigma_E)^2}{2\sigma_E^2} - \frac{(\theta_o - \theta_m)^2}{2\sigma_F^2} - \frac{\psi_{\theta_m}^2}{2\sigma_G^2} \right\} \quad (4.10)$$

$$W_{\theta_l \theta_m \theta_n \theta_o}^{ef} = \max \left(W_{\theta_l \theta_n}^{ef}, W_{\theta_m \theta_o}^{ef} \right) \quad (4.11)$$

where d is the Euclidean distance between the coordinates of neurons e and f ; ψ_θ is the difference in angle between the line subtended by points e and f and a line co-linear with

the orientation θ ; the σ values are parameters that control the extent and specificity of the lateral connections, and S controls the strength of the lateral connections. Only the pre- and post- prediction neurons both representing corners of 90 or larger degrees have the lateral connectivity defined in equations 4.9, 4.10 and 4.11, because only the different corners of 90 degrees or larger can probably form a smooth curve. The definition of the boundary-to-boundary corner connection leads to a strong connection between a pre- and post-synaptic V2 prediction neuron that represents corners that can roughly a smooth curve. Therefore, the pair of θ_l and θ_n in equation 4.9 and the pair of θ_m and θ_o in equation 4.10 are not selected randomly but based on the orientation of the two corners with their four sides represented by the pre- and post-synaptic neurons that can roughly form a smooth curve as indicated in Figure 4-10. Some lateral weights of the boundary-to-boundary corner connection are illustrated in Figure 4-11.

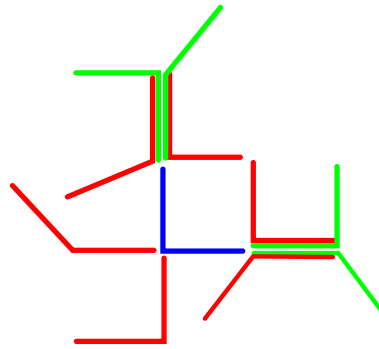


Figure 4-10 The boundary-to-boundary corner lateral connections are only defined between the V2 prediction neurons representing corners that can roughly form a smooth curve.

In this figure, there are strong (no) lateral connections between a post-synaptic prediction neuron represented by a blue corner of 90 degrees and some pre-synaptic prediction neurons represented by some green (red) corners of 90 or larger degrees.

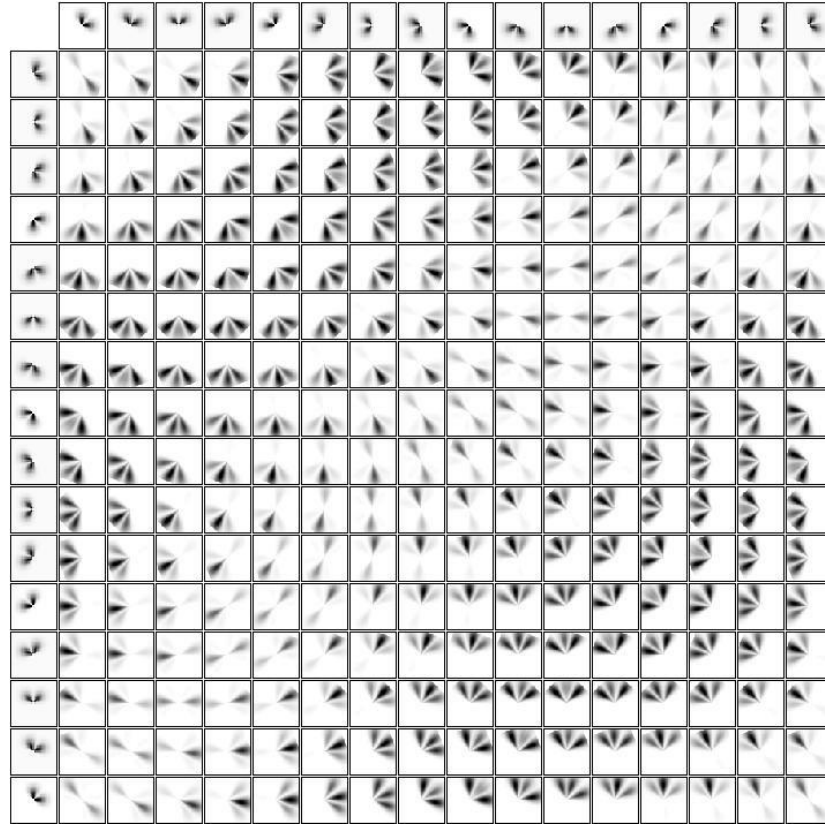


Figure 4-11 Some weights of the boundary-to-boundary corner lateral connections in the PC/BC model of V2.

The left-hand column shows the aggregated receptive field of the post-synaptic neurons representing boundary corners of 90 degrees at different orientations. The top row shows the aggregated receptive field of the pre-synaptic neuron also representing boundary corners of 90 degrees at different orientations. Each other element shows the spatial distribution of the lateral weights between the pre-synaptic neurons and the post-synaptic neurons. They link neurons that represent corners roughly that can roughly form a smooth curve.

The weight kernel of texture-to-texture corner lateral connection between the V2 post-synaptic prediction neuron e that represents a texture corner at θ_{po} degrees comprised of two sides at orientations θ_l and θ_m and the V2 pre-synaptic prediction neuron f at θ_{pr} degrees with orientations θ_n and θ_o is given by:

$$W_{\theta_l \theta_n}^{ef} = S \exp \left\{ -\frac{(d - 2\sigma_E)^2}{2\sigma_E^2} - \frac{(\theta_n - \theta_l)^2}{2\sigma_F^2} - \frac{\phi_{\theta_l}^2}{2\sigma_G^2} - \frac{\omega^2}{2\sigma_H^2} \right\} \quad (4.12)$$

$$W_{\theta_m \theta_o}^{ef} = S \exp \left\{ -\frac{(d - 2\sigma_E)^2}{2\sigma_E^2} - \frac{(\theta_o - \theta_m)^2}{2\sigma_F^2} - \frac{\phi_{\theta_m}^2}{2\sigma_G^2} - \frac{\omega^2}{2\sigma_H^2} \right\} \quad (4.13)$$

$$W_{\theta_{bpo} \theta_{bpr}}^{ef} = S \exp \left\{ -\frac{(d - 2\sigma_E)^2}{2\sigma_E^2} - \frac{(\theta_{bpo} - \theta_{bpr})^2}{2\sigma_F^2} - \frac{\psi_{\theta_{bpo}}^2}{2\sigma_G^2} - \frac{\omega^2}{2\sigma_H^2} \right\} \quad (4.14)$$

$$W_{\theta_l \theta_m \theta_n \theta_o}^{ef} = \max \left(W_{\theta_l \theta_n}^{ef}, W_{\theta_m \theta_o}^{ef}, W_{\theta_{bpo} \theta_{bpr}}^{ef} \right) \quad (4.15)$$

where $\theta_{bpo} = \frac{(\theta_l + \theta_m)}{2}$, $\theta_{bpr} = \frac{(\theta_n + \theta_o)}{2}$, $\omega = \theta_{po} - \theta_{pr}$, d is the Euclidean distance

between the coordinates of neurons e and f ; ϕ_θ is the difference in angle between the line subtended by points e and f and a line perpendicular with the orientation θ ; ψ_θ is the difference in angle between the line subtended by points e and f and a line co-linear with the orientation θ ; the σ values are parameters that control the extent and specificity of the lateral connections, and S controls the strength of the lateral connections. The definition of the texture-to-texture corner connection leads to a strong connection between a pre- and post-synaptic V2 prediction neuron that represents two corners with sides at similar angles and similar orientations. Therefore, the pair of θ_l and θ_n in equation 4.12 and the pair of θ_m and θ_o in equation 4.13 are not selected randomly but based on the similarity of the orientation of two sides from the two corners represented by the pre- and post-synaptic as shown in Figure 4-12. Some lateral weights of the texture-to-texture corner connection are illustrated in Figure 4-13. In this V2 model, the σ_E , σ_F , σ_G , σ_H and S are set to 8, 22.5, 30, 45 and 0.5 respectively. All feedback weights $V_{\theta_l\theta_m\theta_n\theta_o}^{ef}$ are defined to be identical to its corresponding $W_{\theta_l\theta_m\theta_n\theta_o}^{ef}$, but are normalized so that the maximum weight is equal to one.

The contour detection result of the integrated PC/BC model is the pb obtained from linearly reconstructing the V1 prediction neurons representing boundaries as defined in equation 2.7.

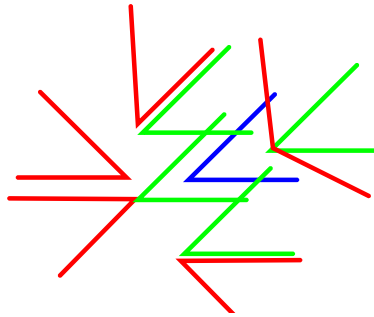


Figure 4-12 The texture-to-texture corner lateral connections are only defined between the V2 prediction neurons that represents corners with sides at similar angles and similar orientations.

In this figure, there are strong (weak or no) lateral connections between a post-synaptic prediction neuron represented by a blue corner of 45 degrees and some pre-synaptic prediction neurons represented by some green (red) corners of 45 or other degrees.

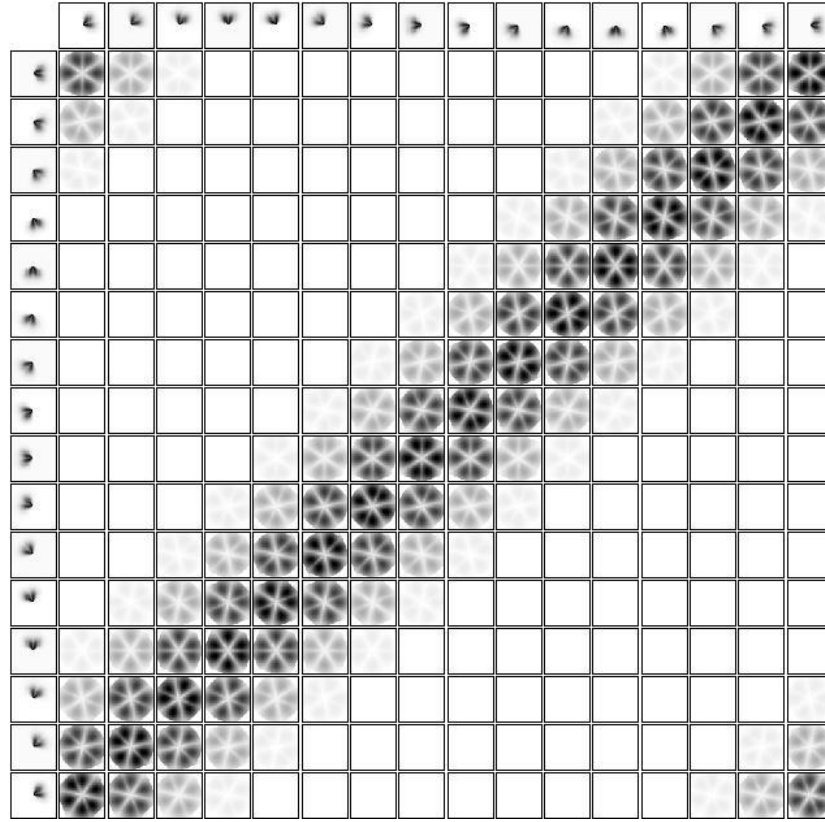


Figure 4-13 Some weights of the texture-to-texture corner lateral connections in the PC/BC model of V2.

The left-hand column shows the aggregated receptive field of the post-synaptic neurons representing a 45 degrees texture corner at different orientations. The top row shows the aggregated receptive field of a pre-synaptic neuron also representing a texture corner of 45 degrees at different orientations. Each other element shows spatial distribution of the lateral weights between the pre-synaptic neurons and the post-synaptic neurons. They link neurons that represent corners with similar angles and at similar orientations.

4.3.2 Results

The contour detection results derived from the PC/BC model of V1 [39] integrated with the PC/BC model of V2 for corner detection are qualitatively and quantitatively evaluated by using the BSDS300 benchmark [36]. The value of each parameter defined in the integrated PC/BC model of V1 for corner detection was determined by trial and error. Specifically, the integrated PC/BC model was operated on training images from the BSDS300 benchmark [36] with a set of manually defined default parameter values. Afterwards, each parameter was sequentially determined while all other parameter values were kept fixed according to the alteration of F-score on these training images. All parameters values defined in the original PC/BC model of V1 [39] were not changed.

When the integrated PC/BC model tested with the BSDS300 benchmark [36], the qualitative evaluation of the corner detection results indicate that the corners detected by the integrated PC/BC model is similar to the Harris corner detection method [139], as shown in Figure 4-14. Furthermore, the qualitative evaluation of the contour detection results indicate that some human identified contours (marked in green in Figure 4-15) defined by insignificant intensity discontinuities and that are weakly detected by the original PC/BC model of V1 [39] can be enhanced by the integrated PC/BC model for corner detection. Whereas, many texture edges which are not human-identified contours (indicated by the yellow rectangle in Figure 4-15) are also enhanced by the integrated PC/BC model. The quantitative results also indicate that the performance of the integrated PC/BC model (F-score 0.51) is significantly worse than the original PC/BC model of V1 [39] (F-score 0.61).

Therefore, the feedback from the prediction neuron responses representing corners in the PC/BC model of V2 to the prediction neurons representing both boundary and texture in the PC/BC model of V1 do not perform as well as previously expected. As shown in Figure 4-15, it is obvious that the bad performance of the integrated PC/BC model is mainly due to the textures within the image not being suppressed as well as in the original PC/BC model of V1 [39]. In the last two images shown in Figure 4-17 and the first four images shown in Figure 4-16, it can be found that some texture corners comprised by some texture edges are represented by the V2 boundary corner neurons and these texture corners can roughly form a smooth curve. Therefore, it is reasonable to claim that the texture suppression mechanism defined in the original PC/BC model of V1 [39] becomes inefficient due to the effect of the boundary-to-boundary corner lateral connections defined by equations 4.9-4.11 in the PC/BC model of V2. However, as shown in Figure 4-16, there is no significant difference between the map of prediction neuron responses representing boundary corners or the map of the reconstruction of these boundary corner neuron responses with and without lateral connections defined in the PC/BC model of V2. Hence the lateral connections defined in the PC/BC model of V2 do not provide negative effect on the texture suppression problem in the integrated PC/BC model. Moreover, as shown in Figure 4-16 and Figure 4-17, many adjacent boundary corners can form a smooth curve and many adjacent texture corners are with similar angles and at similar orientations, which is consistent with the lateral connections defined in equation 4.12-4.15 in the PC/BC model of V2, despite some texture corners are not accurately differentiated from the boundary corners.

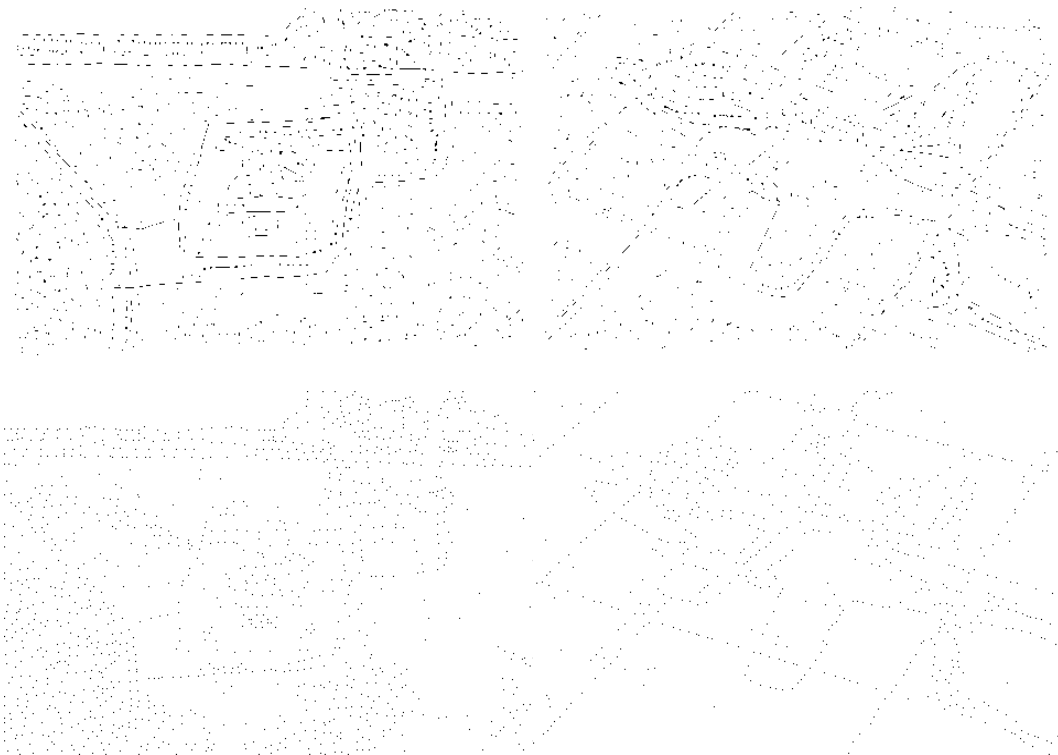
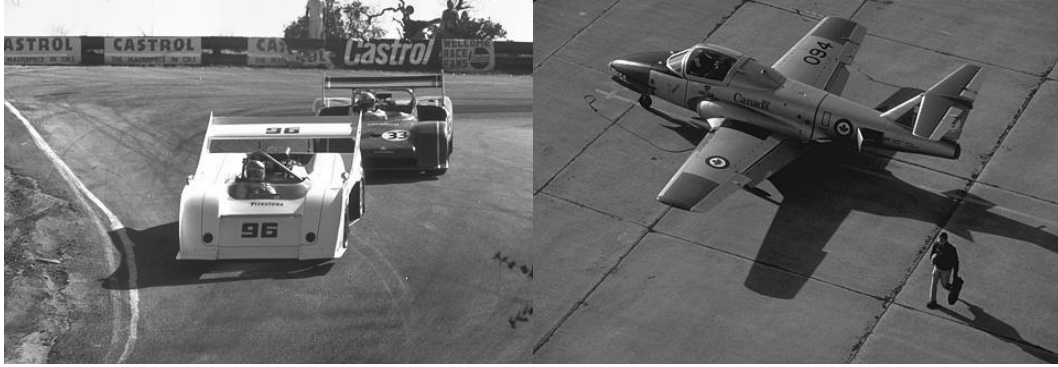


Figure 4-14 Corner detection maps for two images from BSDS300 dataset [36]. These maps include the sum of all V2 neuron responses across 240 channels in the integrated PC/BC model (second row) and the Harris corner detection results [139] (third row). The corners detected by the integrated PC/BC model are similar to those found by Harris corner detection method [139]



Figure 4-15 Results of contour detection for two images from BSDS300 dataset [36]. Contour detection results are produced by (from top to bottom): human, the PC/BC model of V1 [39], and the integrated PC/BC model. Some green-marked human identified contours defined by insignificant intensity discontinuities and weakly detected by the original PC/BC model of V1 [39] can be enhanced by the integrated PC/BC model of corner detection. Some yellow-rectangle marked contours are texture which is not human identified contours but also being enhanced by the integrated PC/BC model for corner detection.

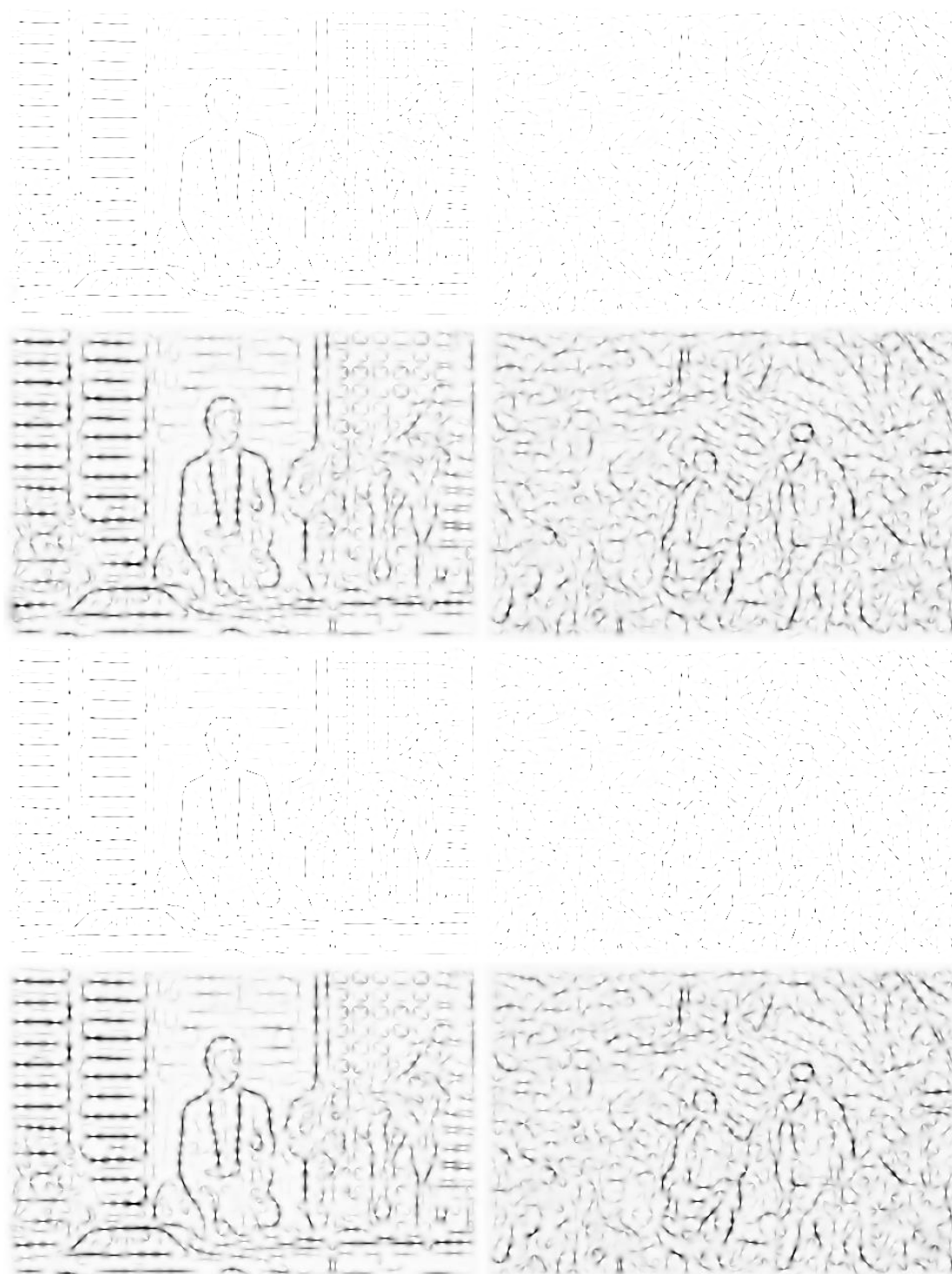


Figure 4-16 Some maps generated by the integrated PC/BC model with and without lateral connections for the two images from BSDS300 [36].

These maps include the sum of V2 boundary corners responses across 120 channels with (without) lateral connections in the first (third) row and the sum of the reconstruction of V2 boundary corners responses across eight orientations with (without) lateral connections in the second (fourth) row. These two different groups of summed boundary corners responses and summed reconstruction of boundary corners responses which are derived from the integrated PC/BC model with and without lateral connections are almost same. Hence the lateral connections defined in the PC/BC model of V2 do not provide negative effect on the texture suppression problem in the integrated PC/BC model.

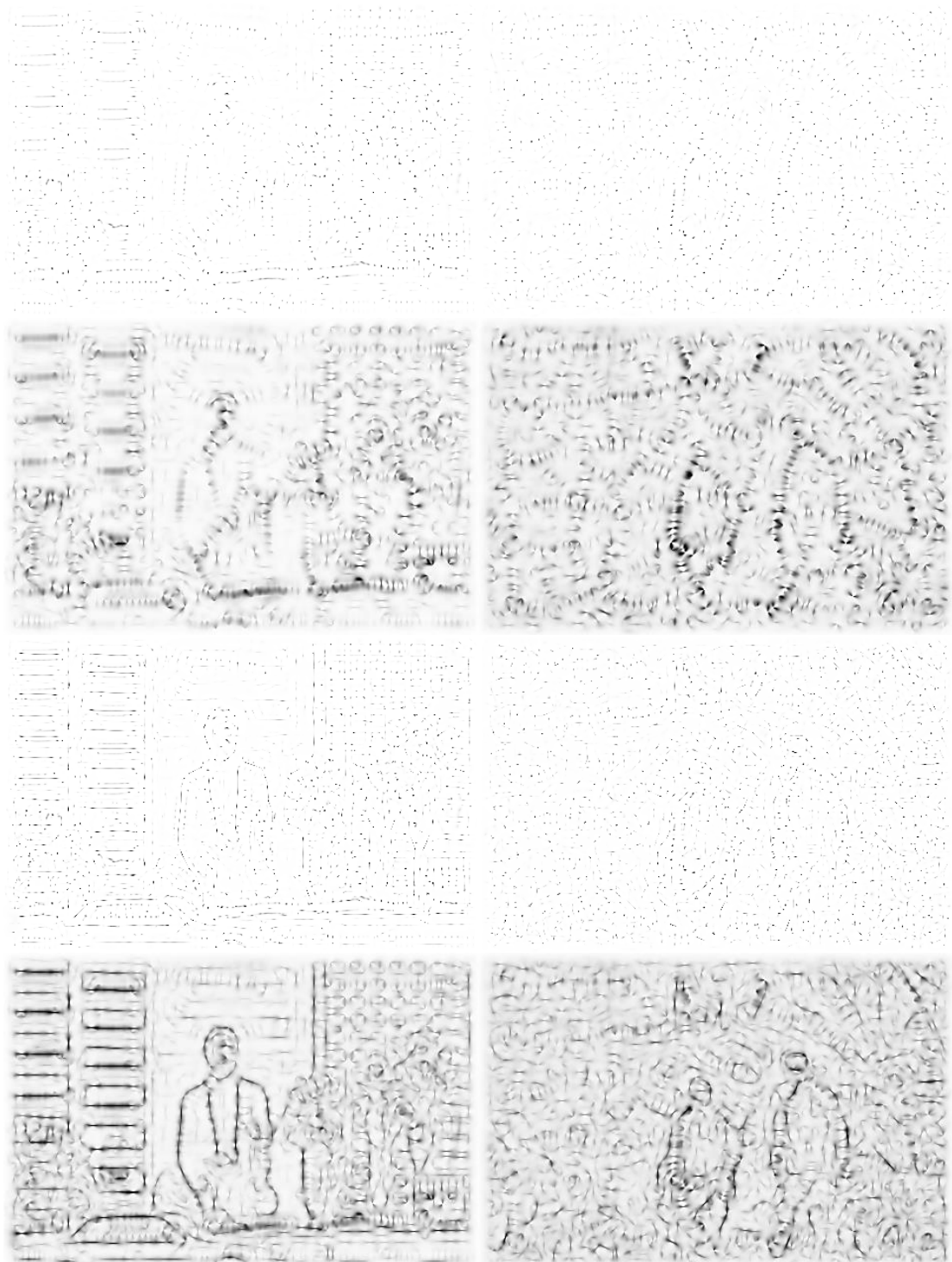


Figure 4-17 Some maps generated by the integrated PC/BC model with lateral connections for the two images from BSDS300 [39].

These maps include (from top to bottom): the sum of V2 texture corners responses across 120 channels, the sum of the reconstruction of V2 texture corners responses across eight orientations, the sum of all V2 neuron responses across 240 channels, and the sum of the reconstruction of all V2 neuron responses across eight orientations. There is no significant difference between different smooth curves formed by the adjacent boundary corners, adjacent texture corners, and the adjacent boundary corners with texture corners

The reason that the texture suppression mechanism defined in the original PC/BC model

of V1 [39] becomes inefficient in the integrated model is mainly due to the feedback from the prediction neurons representing boundary and texture corners in the PC/BC model of V2 to the prediction neurons representing boundary and texture in the PC/BC model of V1. During the recurrent operation of the integrated PC/BC model, some V1 prediction neurons responses representing weak texture edges that can form a corner are enhanced by the feedback from the V2 prediction neurons. Some of these enhanced V1 neuron responses then become more easily collinearly linked with other adjacent V1 neuron responses representing texture edges or boundary edges, which results in many texture edges become represented by the V1 boundary prediction neurons in the final result of the integrated PC/BC model because of the effect of the boundary-to-boundary lateral connections defined in the PC/BC model of V1 [39]. Therefore, the problem of the inefficiency of the texture suppression mechanism in the integrated PC/BC model can hardly be solved by only defining a new set of lateral connections in the PC/BC model of V2 which can precisely differentiate the boundary corners from texture corners, because there is no significant difference that can be recognized even by the eyes between different smooth curves formed by the adjacent boundary corners, adjacent texture corners, and the adjacent boundary corners with texture corners as shown in the last four images of Figure 4-17. Hence a completely new texture suppression mechanism would need to be defined in both the PC/BC model of V1 and the PC/BC model of V2 to successfully suppress texture edges and to improve the performance of the integrated PC/BC model, despite some of the weak contours defined by insignificant intensity discontinuities can be strengthened in the current model.

4.4 Discussion

In section 4.2, the PC/BC model of V2 devised to locate boundaries defined by discontinuities in texture is integrated with the original PC/BC model of V1 [39], which is expected to be able to improve contour detection derived from the PC/BC model of V1 [39] by locating more contours within an image. However, the integrated PC/BC model performed worse than the PC/BC model of V1 [39]. Despite some texture boundaries within natural images [36] and synthetic texture images [47] which cannot be located by the PC/BC model of V1 can be located by the PC/BC model of V2. Whereas, the comprehensiveness and accuracy of the texture boundaries located by the PC/BC model of V2 is not good enough to improve the performance of the integrated PC/BC model. The main reason for the bad performance of the PC/BC model of V2 for texture boundary detection is probably that the texture information represented by the

current transformation of the V1 prediction neuron responses is insufficient and inaccurate for the PC/BC model of V2 to locate comprehensive and accurate texture boundaries. Hence it is possible that the performance of the integrated PC/BC model can be improved compared to the original PC/BC model of V1 [39] only if a new method can be proposed to sufficiently and accurately represent texture information using the prediction neuron responses generated from the framework of the PC/BC model. A possible solution is to define the receptive fields of V1 prediction neurons in multiple sizes first. Then the PC/BC model of V1 is operated independently to generate the responses of prediction neurons whose receptive fields are in different sizes. Afterwards, the prediction neuron responses that represent texture edges generated in each operation can be integrated to represent texture information more sufficiently and accurately.

In section 4.3, the PC/BC model of V2 devised to detect corners is integrated with the original PC/BC model of V1 [39]. This was anticipated to be able to enhance some weak contours defined by insignificant intensity discontinuities by employing the feedback from prediction neuron responses in the PC/BC model of V2 to prediction neuron responses in the PC/BC model of V1, improving the performance of the contour detection result. However, the integrated PC/BC model performed worse than the original PC/BC model of V1 [39]. Despite the corners detected by the integrated PC/BC model is similar to the Harris corners detection result [139] and some of the weak contours defined by insignificant intensity discontinuities can be enhanced in the result derived from the integrated PC/BC model, there is a new problem occurring in the integrated PC/BC model that corrupts the performance of the contour detection result. Specifically, the texture suppression mechanism is not efficient in the integrated PC/BC model, which causes many false contours (texture edges) existing in the contour detection result. The reason for the inefficiency of the texture suppression mechanism defined in the PC/BC model of V1 [39] is mainly due to some V1 prediction neurons responses representing weak texture edges that can form a corner are enhanced by the feedback from the V2 prediction neurons. Some of these enhanced V1 neuron responses then become more easily colinearly linked with other adjacent V1 neuron responses representing texture edges or boundary edges, which leads to many texture edges becoming represented by the V1 boundary prediction neurons in the final result because of the effect of the boundary-to-boundary lateral connections defined in the PC/BC model of V1 [39]. Therefore, the problem of the inefficiency of the texture suppression

mechanism in the integrated PC/BC model can hardly be solved by only defining a new set of lateral connections in the PC/BC model of V2 which can precisely differentiate the boundary corners from texture corners, because there is no significant difference that can be recognized even by the eyes, between different smooth curves formed by the adjacent boundary corners, adjacent texture corners, and the adjacent boundary corners with texture corners. Hence a completely new texture suppression mechanism would need to be defined in both the PC/BC model of V1 and the PC/BC model of V2 to successfully suppress textures for the integrated PC/BC model.

Chapter 5 A Simplified Texture Gradient Method

5.1 Introduction

To locate boundaries defined by texture discontinuities in computer vision, an algorithm needs to firstly represent the structure of texture within an image, and then work out which pixels are located at boundaries between two different structures of texture. Therefore, the effectiveness of a texture boundary detection algorithm highly depends on how it represents the structure of texture. For instance, [12] utilized two-sample test of independence, [13] employed G statistics of the distribution of local binary patterns, [62] used a Gaussian window on Gabor filtered responses, [63] utilized statistical distributions of Gabor filtered responses, and [64] employed a Markov chain to represent the structure of texture within an image.

The texture gradient method used in the probability of boundary (Pb) algorithm [47] used the K-Means algorithm to cluster the responses of Gaussian derivative filters to represent the structure of texture. This produced good performance in locating boundaries defined by texture discontinuities. However, the K-Means algorithm is computationally time-consuming and a non-biologically-inspired method. Therefore, if it is possible to accurately represent the structure of texture using the filtered responses without the application of the K-Means algorithm, the computation time of the texture gradient method used in the Pb algorithm [47] could be reduced. Furthermore, this texture gradient method can also become a biologically-inspired method, because the filtered responses utilized to represent the structure of texture can be interpreted as the simulation of neuron responses in V1 [93] and the computation of the difference of filtered responses between neighbouring regions to locate the boundaries between two different structures of texture can be interpreted as the simulation of illusory contours detection in V2 [42][43].

In this Chapter, a simplified and biologically-inspired version of the Pb algorithm's texture gradient method [47] is developed to locate texture boundaries within images. The performance of this algorithm at locating texture boundaries is compared to the Pb algorithm using the BSDS300 benchmark [36] and the Prague texture segmentation dataset [37]. The proposed simplified biologically-inspired texture gradient method directly use filtered responses to represent structure of texture (V1 model) and then applied oriented half Gaussian filters to compute the difference of filtered responses in

order to locate the boundaries between two different structures of texture (V2 model). Because the proposed method for locating boundaries is based on local discontinuities in texture alone, it cannot detect boundaries defined by intensity discontinuities. The PC/BC model of V1 [39] is able to locate boundaries defined by intensity discontinuities. It has been shown that the PC/BC model of V1 outperforms the Pb algorithm [47] when this algorithm is also restricted to using local discontinuities in intensity (brightness) at single-scale in grayscale images. Hence, to better evaluate the effectiveness of the proposed method for locating texture boundaries within grayscale images, boundaries defined by intensity discontinuities derived from the PC/BC model of V1 are integrated with boundaries defined by discontinuities in texture derived from the proposed method. This combined contour detection result enables boundaries defined by both of these local discontinuities to be located.

Because the proposed texture boundary detection method is inspired by the texture gradient method used in the Probability of Boundary algorithm [47], section 5.2.1 first reviews this algorithm. Afterwards, section 5.2.2 describes the proposed texture gradient method in detail, highlighting the differences compared with the Pb algorithm and the complexity advantage of each difference. Section 5.2.3 details how the proposed algorithm is combined with the PC/BC model of V1. The contour detection results derived from the proposed method and the integrated method are given in section 5.3. Section 5.4 discusses the achievement and the rationality of the proposed method.

5.2 Methods

5.2.1 The Texture Gradient Method Used in the Probability of Boundary Algorithm

To locate boundaries defined by texture discontinuities in computer vision, an algorithm needs to firstly represent the structure of texture within an image, and then work out which pixels are located at boundaries between two different structures of texture. The texture Gradient method used in the Probability of Boundary Algorithm [47] utilized the ‘texton’ defined as the K-Means clustering of the filtered responses to represent the structure of texture and then defined uniform half discs to compute the difference of ‘texton’ between neighbouring regions to locate boundaries between two different structures of texture.

The filter bank used to generate filtered responses contains both even-symmetric and odd-symmetric filters (second order derivative of Gaussians and their Hilbert transforms). These filters are defined at two scales (with a standard deviation of 1 and $\sqrt{2}$ pixels) and at six evenly spaced orientations. The sizes of these filters at the two scales are 13×13 and 19×19 pixels respectively. This filter bank is shown in Figure 5-1. Afterwards, 24 filtered responses maps are produced by the convolution of the filter bank with the original image. The ‘texton’ is computed by applying the K-Means algorithms to cluster these response maps. Therefore, each pixel within an image can be assigned its closest texton. In other words, each pixel within an image is represented by the label of its closest cluster centre to represent different structures of texture for further computation. In fact, in order to derive better texture boundary results, the Pb algorithm [47] does not apply the K-Means clustering on each new image, but firstly records all filter bank responses from all training images in the BSDS300 benchmark [36]. The K-Means clustering is applied to all responses from all images in the training set to generate 64 universal textons. When processing each new image, each pixel is assigned the label of the closest universal texton to represent different structures of texture. The texton labels are represented using 64 binary maps.

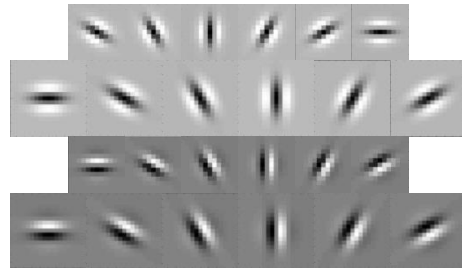


Figure 5-1 Filter bank for the texture gradient method used in the Pb algorithm [47].

After the different structures of texture have been represented by the label of the closest universal texton for each pixel, the boundaries between two different structures of texture are located by computing the texture gradients, which are defined by the difference in the number of pixels with the same texton labels within two neighbouring semi-circular regions. Hence, a second set of filters are defined which consist of two uniform half discs at eight evenly spaced orientations to be convolved with the 64 binary texton label maps to count the number of pixels with the same label within each semi-circular region. All the values in the two half discs are the same, and the sum of values in each half disc have been normalized to one. The size of each half disc is 23×23 pixels. These half disc filters are shown in Figure 5-2. Afterwards, the difference in the number of pixels with the same texton label between each half of one disc at one

of eight orientations is then computed. Therefore, 64 binary label maps yield 64 differences at each orientation. The sum of these 64 differences is defined as the texture gradient at one orientation. There are thus eight texture gradients, one at each orientation. The Pb algorithm utilizes Savitzky-Golay filter [47] to smooth these eight texture gradients, which is a kind of polynomial regression on these texture gradients. However, only the maximum texture gradient at each pixel is kept across the eight orientations. Four examples of the maximum texture gradient images are shown in Figure 5-3. Finally, non-maximum suppression [10] is applied to thin the maximum texture gradients to texture boundaries being the final texture boundary result.



Figure 5-2 Half disc filters defined in the Pb algorithm

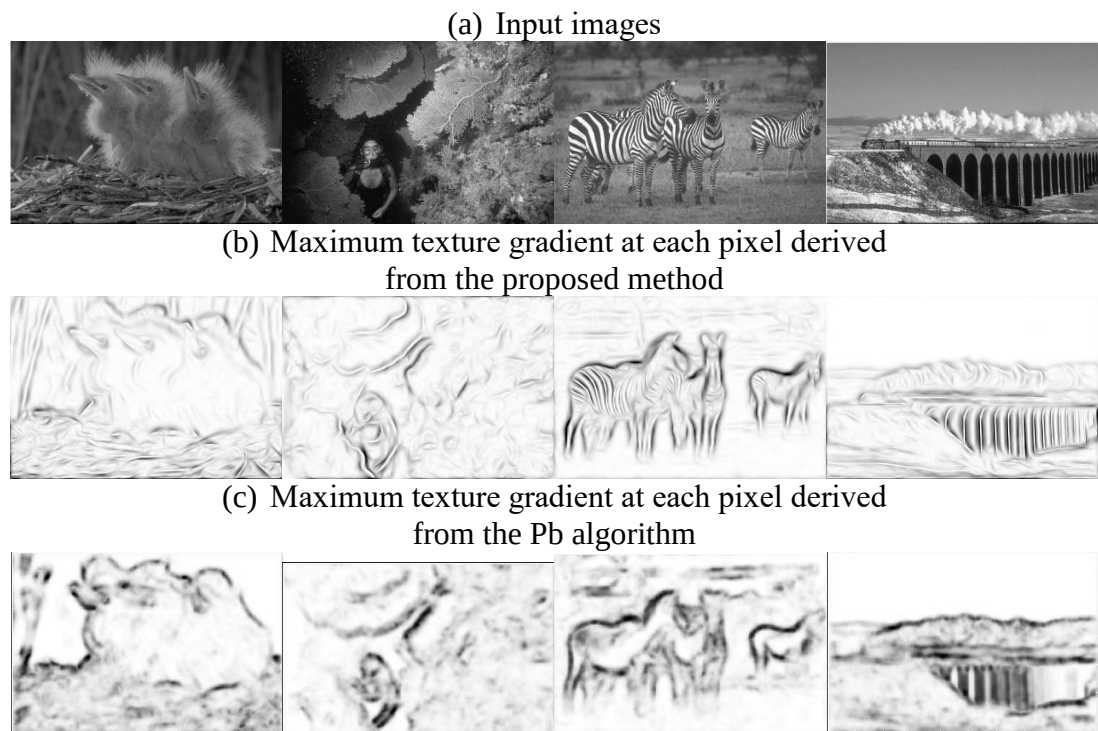


Figure 5-3 Four original images from BSDS300 dataset [36] with their maximum texture gradients at each pixel generated by the proposed method and the Pb algorithm [47].

Similar texture boundaries can be located by the maximum texture gradients derived from the proposed method and the Pb algorithm [47].

5.2.2 Proposed Texture Gradient Method

To locate boundaries defined by texture discontinuities in computer vision, an algorithm needs to firstly represent the structure of texture within an image, and then work out

which pixels are located at boundaries between two different structures of texture. The proposed biologically-inspired and simplified texture gradient method directly utilizes filtered responses to represent the structure of texture, which can be interpreted as the simulation of neuron responses in V1 [93]. Then half Gaussian filters are defined to compute the difference of the sum of filtered responses within neighbouring Gaussian semi-circular regions to locate boundaries between two different structures of texture, which can be interpreted as the simulation of illusory contour detection in V2 [42][43]. The procedure of the proposed biologically-inspired texture gradient method is illustrated in Figure 5-4. The differences between the proposed method and the Pb algorithm's texture gradient method are highlighted.

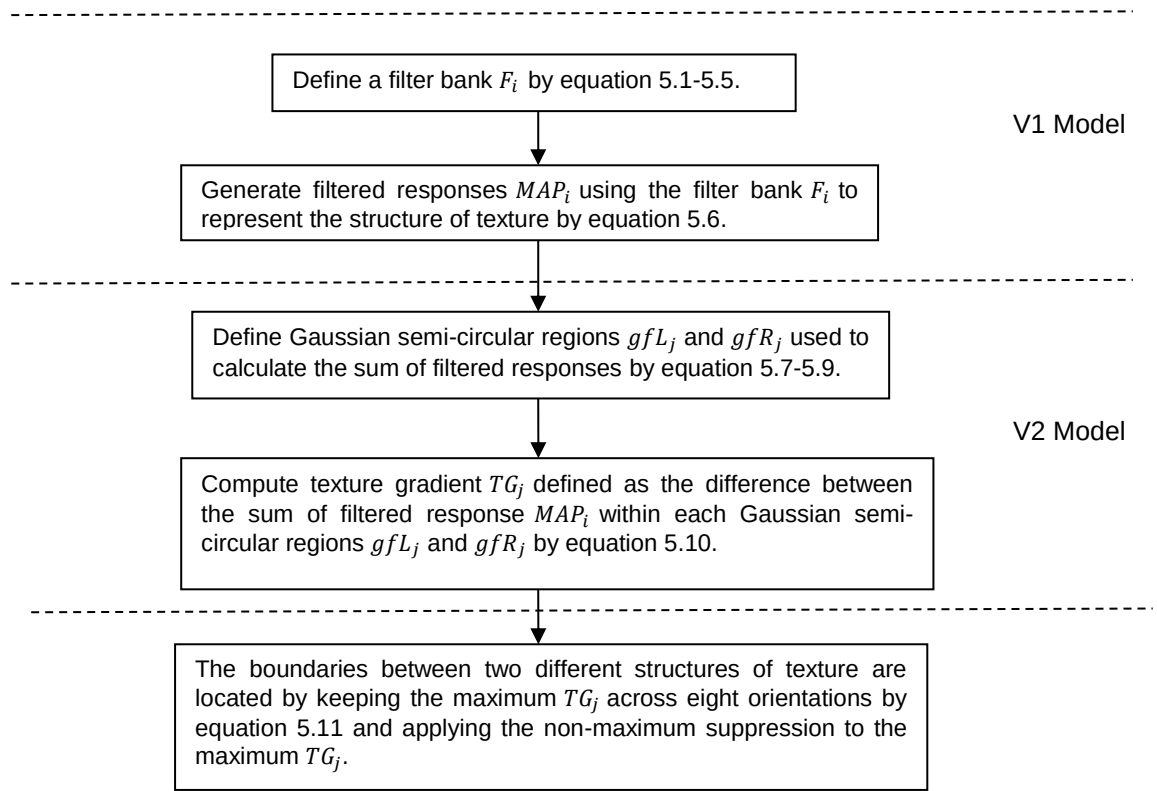


Figure 5-4 The procedure of the proposed texture gradient method.

5.2.2.1 V1 Model

Because the receptive fields of neurons in V1 are similar to first and second order derivative of Gaussian filters [93], the absolute value of filtered responses used to represent the structure of texture can be interpreted as neuron responses in V1. Specifically, the filter bank utilized to produce filtered responses is comprised of both odd-symmetric and even-symmetric filters (first and second order derivatives of a Gaussian), as defined in:

$$f(x,y) = \frac{1}{\sqrt{2\pi}\sigma_{fb}} \times e^{-\frac{1}{2}\left(\frac{x^2}{(\sigma_{fb} \times EL)^2} + \frac{y^2}{\sigma_{fb}^2}\right)} \quad (5.1)$$

$$f'(x,y) = f(x,y) \times \frac{-y}{\sigma_{fb}^2} \quad (5.2)$$

$$f''(x,y) = f(x,y) \times \frac{y^2 - \sigma_{fb}^2}{\sigma_{fb}^4} \quad (5.3)$$

$$F_i = \text{rotate}\{f'(x,y), \theta_i\} \text{ for } i = 1, 2 \dots 8 \quad (5.4)$$

$$F_i = \text{rotate}\{f''(x,y), \theta_i\} \text{ for } i = 9, 10 \dots 16 \quad (5.5)$$

where F_i represents the filters in the filter bank and is defined at one scale with a standard deviation of $\sigma_{fb}=0.3$ pixels and aspect ratio of Gaussian envelope $EL=3$ pixels. The sizes of these filters F_i are 7×7 pixels. The first order derivative of Gaussian filters $f'(x,y)$ and the second order derivative of Gaussian filters $f''(x,y)$ defined in equation 5.2 and 5.3 respectively are rotated by θ_i in the range $[0^\circ, 360^\circ)$ in steps of 22.5° to generate these filters F_i in the filter bank. Hence these filters F_i are defined at eight evenly spaced orientations ($FBORIENT=8$) as shown in Figure 5-5. Afterwards, these filters F_i in the filter bank are convolved with input images to generate 16 maps of absolute value of filtered responses to represent the structure of texture, as defined in:

$$MAP_i = |I \star F_i| \quad \text{for } i=1,2,3,\dots,16 \quad (5.6)$$

where I is the original image, F_i is the i^{th} filter in the filter bank defined in equation 5.4 and 5.5, $\star$ is the convolution operation, and MAP_i is the i^{th} response map to the i^{th} filter. The number and size of the filters determines the computation time when they are convolved with the same input image. The proposed method using 16 filters of 7×7 pixels has a significant advantage over the Pb algorithm [47] with 24 filters of 13×13 and 19×19 pixels. The comparison of computation time is shown in Table 5-1.

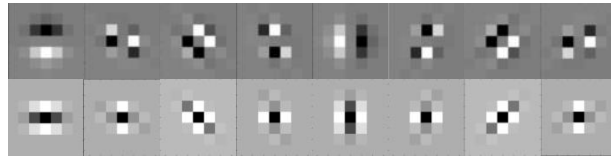


Figure 5-5 Filter bank of the proposed texture gradient method

As previously mentioned, the representation of structure of texture in the Pb algorithm [47] can be implemented either by obtaining universal textons from training images, or by applying the K-Means clustering algorithm to generate textons for each input image individually. In the former case, the training can be done off-line (and the time taken will depend on the size of training image set) and only the classification of the filter

responses needs to be performed on-line. In the latter case, both training and classification are performed on-line. The computation time for applying the K-Means clustering algorithm off-line and on-line is shown in Table 5-1. In contrast to the Pb algorithm, the proposed method does not apply the K-Means clustering algorithm, instead texture gradients are calculated directly from the filter response maps. Hence, there is an additional computation time for Pb compared to the proposed algorithm.

5.2.2.2 V2 Model

After the structure of texture represented by the filtered response from the V1 model, the proposed biologically-inspired method locates boundaries between two different structures of texture by computing the difference of the sum of filtered responses between neighbouring Gaussian semi-circular regions. The normalized half Gaussian filters are defined by:

$$f(x,y) = \frac{1}{\sqrt{2\pi}\sigma_G} \times e^{-\frac{1}{2}\left(\frac{x^2}{\sigma_G^2} + \frac{y^2}{\sigma_G^2}\right)} \quad (5.7)$$

$$gfL_j = \underset{j}{\text{rotate}}\{f(x,y), \theta_j\} \text{ for } j = 1, 2 \dots 8 \text{ and } y \leq 16 \quad (5.8)$$

$$gfR_j = \underset{j}{\text{rotate}}\{f(x,y), \theta_j\} \text{ for } j = 1, 2 \dots 8 \text{ and } y > 16 \quad (5.9)$$

where the standard deviation of the Gaussian filters $f(x,y)$ is $\sigma_G=6$ pixels, and the size of each half Gaussian filter is 31×31 pixels. The Gaussian filter $f(x,y)$ defined in equation 5.7 is bipartition along the line $y=16$, and then two Gaussian semi-circular regions are rotated by θ_j in the range $[0^\circ, 360^\circ)$ in steps of 22.5° to generate the gfL_j or gfR_j at overall $TGORIENT=8$ orientations. These half Gaussian filters are shown in Figure 5-6. In contrast to the Pb algorithm, employing filters that are half Gaussians instead of half uniform discs avoids the need to use Savitzky-Golay filter [47] smoothing that is used in the Pb algorithm, and also provides a better trade-off between comparing large regions of the image and localizing boundaries.

Afterwards, the half Gaussian filters gfL_j or gfR_j are convolved with 16 filtered responses maps derived from equation 5.6 to calculate 16 maps of the sum of filtered responses within each Gaussian semi-circular region. And 16 differences between the 16 maps of the sum of filtered responses within each Gaussian semi-circular region at each orientation are then computed. The sum of these 16 differences is the texture gradient TG_j at the j^{th} of $TGORIENT=8$ orientations, as defined by:

$$TG_j = WT \sum_{i=1}^{16} \frac{(tgL_{ij} - tgR_{ij})^2}{tgL_{ij} + tgR_{ij}}$$

where $\begin{cases} tgL_{ij} = MAP_i \star gfl_j \\ tgR_{ij} = MAP_i \star gfr_j \end{cases}$ for $j=1,2,\dots,8$ (5.10)

and where MAP_i is the i^{th} response map to the i^{th} filter in the filter bank; $\star$ is convolution operation, gfl_j or gfr_j is a half Gaussian filter at the j^{th} orientation; tgL_{ij} and tgR_{ij} compute the sum of responses from the i^{th} response maps at each pixel within each half Gaussian region at the j^{th} orientation; WT determines the range of texture gradient values, and is used in the combined algorithm in section 5.2.3. Because the size of half Gaussian filter in the proposed method (31×31 pixels) is bigger than the size of half disc in the Pb algorithm (23×23 pixels), there is not a clear complexity advantage at this step for the proposed method. However, the proposed method uses a smaller number of input maps, 16, compared to the 64 label maps used by the Pb algorithm [47]. The computation times of the texture gradient operation for the proposed method and the Pb algorithm are shown in Table 5-1.

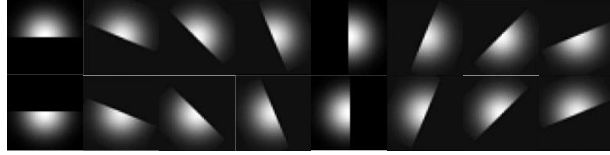


Figure 5-6 Half Gaussian filters defined in the proposed method

The proposed method only keeps the maximum texture gradients MTG at each pixel across $TGORIENT=8$ orientations as defined in:

$$MTG = \max_j \{TG_j(x, y)\} \quad (5.11)$$

where $TG_j(x, y)$ represents the texture gradient at overall $TGORIENT=8$ orientations. Finally, the non-maximum suppression [10] is applied to MTG to thin the texture boundaries. These thinned texture boundaries are the final texture boundary result derived from the proposed method.

In summary, the main differences between the proposed texture gradient method and the texture gradient method used in the probability of boundary algorithm [47] are in the following three aspects.

- **First, the filter bank which is convolved with the original images.** The proposed method defines odd-symmetric and even-symmetric filters at only one scale at eight evenly spaced orientations; in contrast, both even-symmetric and odd-symmetric

filters at two scales and six evenly spaced orientations are defined in the Pb algorithm. The proposed method has the complexity advantage at this stage because of the significantly smaller number and size of the filters in the filter bank.

- **Second, the sources that are used to compute the texture gradient.** The proposed method directly uses the absolute values of the filter outputs to compute texture gradients instead of employing (universal) texton labels. The proposed method has a complexity advantage at this step, because of the significantly additional computation time of applying the K-Means clustering algorithm (either off-line or on-line) and assigning a texton label to each pixel.
- **Third, the second set of filters which are defined to compute the sum of sources within semi-circular regions, and hence to compute the texture gradient.** The proposed method defines filters that are half Gaussians instead of filters that are uniform half discs. Despite the half Gaussian filters being larger, the proposed method still has the complexity advantage at this step due to the significantly smaller number of input maps with which the filter is convolved.

From the differences listed above it can be seen that the proposed method is considerably simpler than the texture gradient method used in the Pb algorithm. Furthermore, the proposed method is also more biologically-plausible, because the non-biologically-inspired K-Means clustering process is removed. To quantitatively evaluate the effect of these differences on the computational complexity between the texture gradient method used in the Pb algorithm [47] and the proposed method, the computation time of each step is compared using a computer with an Intel® Core™ i7-3770 CPU@3.40GHZ CPU. The computation times for these two different methods, which are shown in Table 5-1, are the times taken to process the 100 test images from the BSDS300 dataset [36]. These results indicate that the proposed method is approximately 2 (or 6) seconds per image faster than the texture gradient method used in the Pb algorithm which is implemented by applying the K-Means clustering algorithm off-line (or on-line). Although the computation time for the Pb algorithm may become shorter if the Pb algorithm is implemented by an optimized method, the reported computation time for the Pb algorithm is obtained using the code provided by the authors of that algorithm [47].

Table 5-1 The computation time of different stages in the proposed method and the texture gradient method used in the Pb algorithm [47] when both of the two methods are operated on 100 natural images from the BSDS300 benchmark [36].

Step(s)	Method(s)	Computation time (seconds)
Convolution operation	Proposed	3.15
	Pb	17.76
Obtaining (universal) texton and assigning texton label	Proposed	0
	Pb (off-line)	52.81
	Pb (on-line)	433.44
Texture gradient operation	Proposed	303.68
	Pb	415.40
Overall steps	Proposed	338.63
	Pb (off-line)	538.32
	Pb (on-line)	928.35

5.2.3 Combination with Boundaries Defined by Intensity Discontinuities

Derived from the PC/BC Model of V1

The following three modifications are required to integrate the boundaries derived from the proposed model with the boundaries derived from the PC/BC model of V1 [39].

The absolute values of the filtered responses which are derived from the proposed method are the intensity discontinuities in quite small regions within an input image. The proposed method utilizes these responses to represent the structure of texture and then compute texture gradient to locate texture boundaries. Therefore, it is unavoidable that parts of these responses do not represent texture but the boundaries defined by local intensity discontinuities. Using these parts of the filtered responses in the texture gradient method makes these local intensity boundaries become fuzzy. In order to avoid these fuzzy local intensity boundaries located by the proposed method contaminating the boundaries derived from the PC/BC model of V1, the boundaries derived from the PC/BC model of V1 are subtracted from the 16 response maps generated in the proposed method, as defined in:

$$MAP_i \leftarrow |MAP_i - V1R| \quad (5.12)$$

where MAP_i is the i^{th} of 16 response maps, $V1R$ is the boundary responses derived from the PC/BC model of V1 based on local intensity discontinuities.

Second, the range of values for the texture gradient will influence the final combined algorithm's boundary result. Hence, the WT defined in equation 5.10 is set to a value of 2.

Third, in step 5, after the maximum texture gradient is found at each pixel across the eight orientations, the maximum texture gradient is added to the boundaries derived from the PC/BC model of V1 [39] to yield a combined map. Afterwards, non-maximum suppression [10] is applied to thin the boundaries in this combined map. The combined boundaries are the final result and contain boundaries defined by local discontinuities in both texture and intensity.

5.3 Results

The proposed biologically-inspired texture gradient method is qualitatively and quantitatively evaluated by the BSDS300 benchmark [36] and the Prague texture segmentation dataset [37]. When the texture gradient method is tested with the BSDS300 benchmark [36], the qualitative evaluation results indicate that most of the human identified texture boundaries that are located by the texture gradient method used in the Pb algorithm [47] can also be detected by the proposed method, as shown in Figure 5-7. Although the proposed texture gradient method is a considerable simplification compared to the texture gradient method used in the Pb algorithm, the quantitative evaluation results indicate that two methods generate the same F-score 0.58 [47], as shown in Figure 5-8. When the proposed texture gradient method is tested with the Prague dataset [37], the qualitative results indicate that most of the synthetic texture boundaries that are located by the texture gradient method used in the Pb algorithm can also be detected by the proposed method, as shown in Figure 5-9. Because both the texture gradient used in the Pb algorithm [47] and the proposed texture gradient method are designed to locate discontinuities in texture within natural images, these two methods can locate many additional texture boundaries within each part of the synthetic texture images as well as the synthetic texture boundaries, which results in poor performance for these two methods on synthetic texture images when quantitatively evaluated using the F-score. However, the proposed method still have slightly better result (F-score 0.23) compared with the Pb algorithm (F-score 0.21).

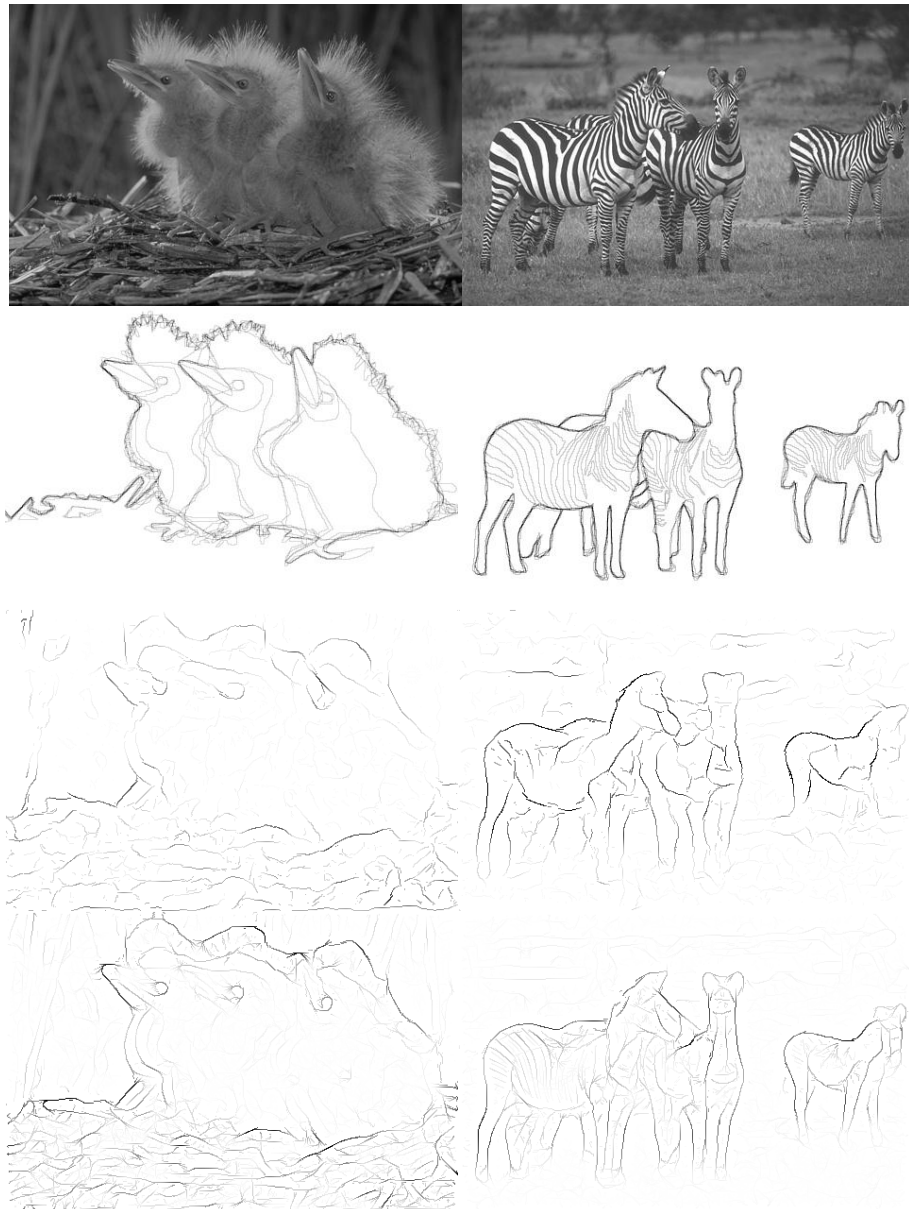


Figure 5-7 Results of texture boundary detections for two images are from BSDS300 dataset [36].

Texture boundary detection results are produced by (from top to bottom): human, the texture gradient method used in the Pb algorithm [39], and the proposed biologically-inspired texture gradient method. Most of the human-identified texture boundaries being located by the Pb algorithm [39] can also be located by the proposed method.

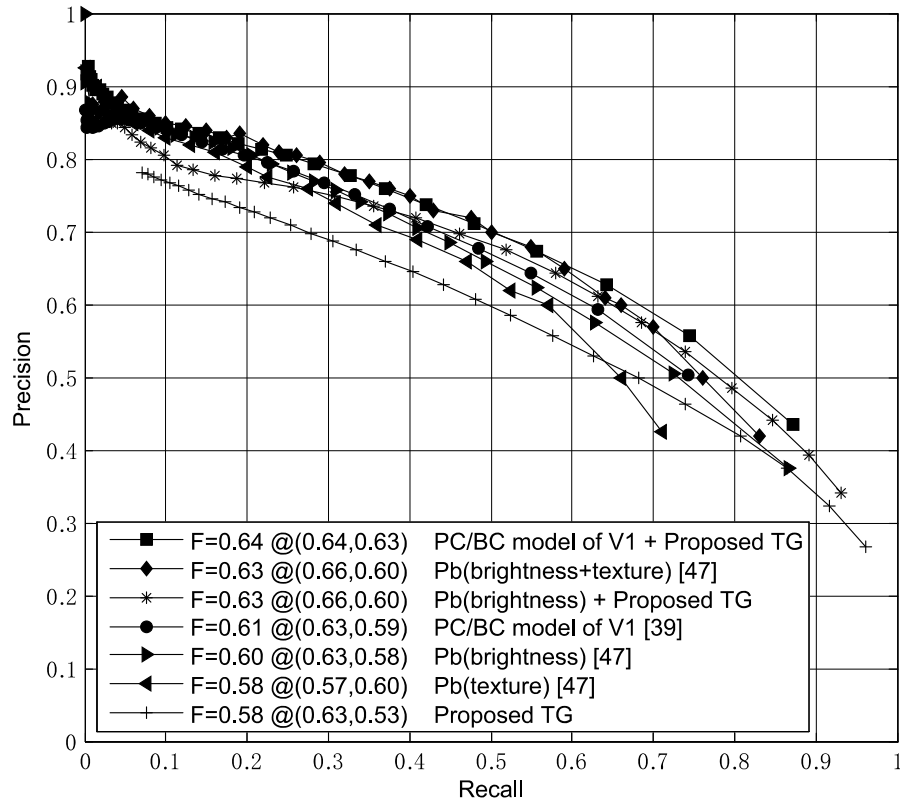


Figure 5-8 Performances of different contour detection algorithms evaluated by the precision-recall curves and F-score from the BSDS300 Benchmark [36]. These algorithms only use single-scale local cues within grayscale images.

Furthermore, as shown in Figure 5-8, the performance obtained by combining the brightness gradient method used in the Pb algorithm with the proposed method is the same as the performance obtained by combining the brightness and texture gradient methods used in the Pb algorithm (F-score 0.63). Hence, the proposed simplifications do not affect performance. The performance obtained by integrating the proposed method with the PC/BC model of V1 (F-score 0.64) is slightly better than the performance obtained by the Pb algorithm (F-score 0.63) [47]. The reason that the combined algorithm, outperforms the Pb algorithm is that the PC/BC model of V1 produces better performance in detecting boundaries defined by local intensity discontinuities (F-score 0.61) [39] than the brightness gradient method used in the Pb algorithm (F-score 0.60) [40].

The two recent near state-of-the-art algorithms for contour detection are SCG [49] and gPb-owt-ucm [18]. The gPb-owt-ucm algorithm [18] is an extended version of the Pb algorithm [47]. As shown in Table 5-2, these two algorithms have the same F-score of

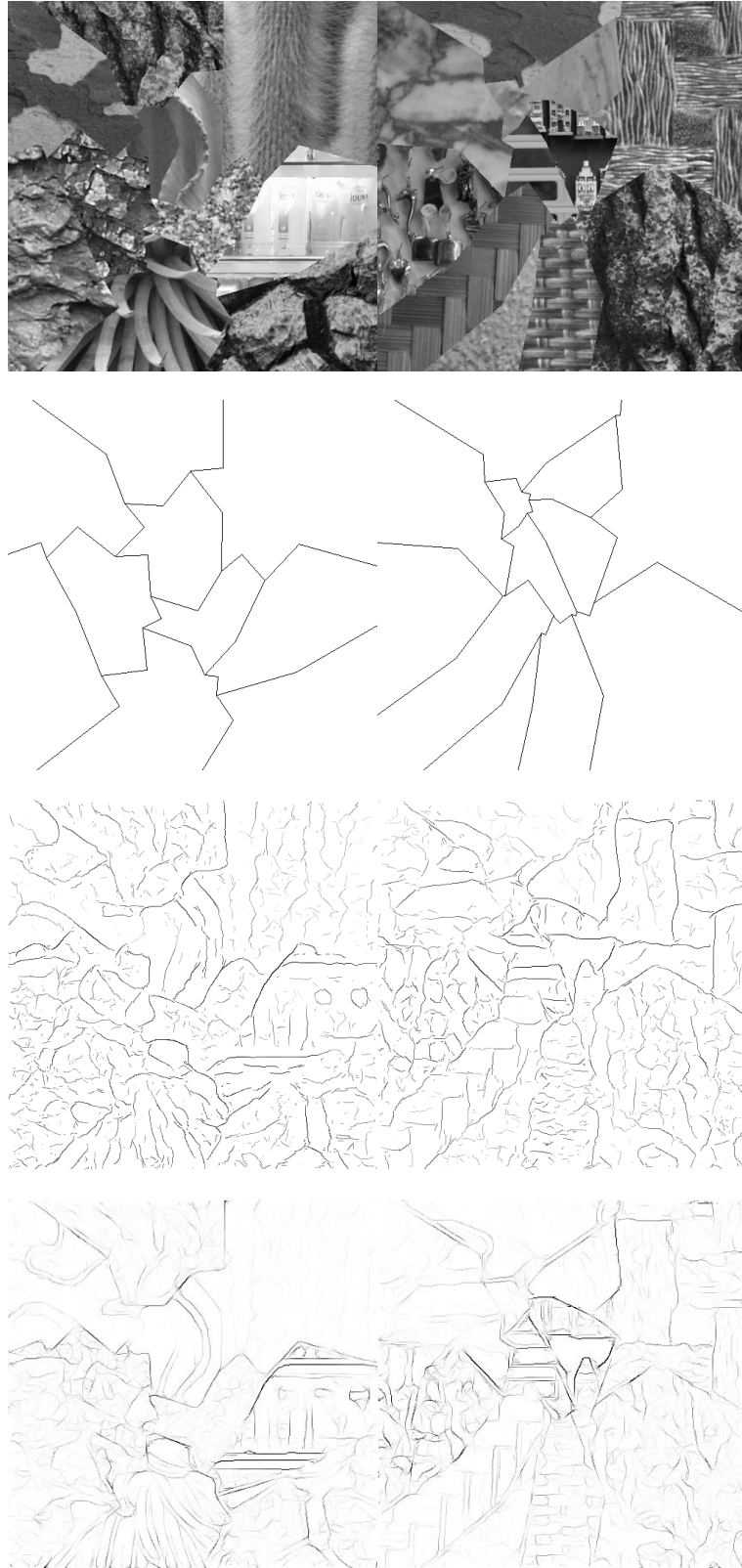


Figure 5-9 Results of texture boundary detections for three images from Prague benchmark [37].

Texture boundary detection results are produced by (from top to bottom): ground truth, the texture gradient method used in the Pb algorithm [39], and the proposed biologically-inspired texture gradient method. Most of the synthetic texture boundaries being located by the Pb algorithm [39] can also be located by the proposed method.

0.71, which is significantly higher than that of the proposed method integrated with the brightness method used in the Pb algorithm (0.63). SCG [49] and gPb-owt-ucm [18] both employ multiple image features and image scales. These multiple image features include local discontinuities in colour, intensity, texture and also global information of these features within an image. In contrast, the integrated algorithm only employs local intensity texture discontinuities at a single-scale. Therefore, the computation time of the proposed method integrated with the brightness gradient method used in the Pb algorithm is significantly lower than that of the recent near state-of-the-art algorithms when tested on 100 test images from the BSDS300 dataset [36], which is shown in Table 5-2. The computation times for the recent near state-of-the-art algorithms were obtained using the code provided by the authors of those algorithms [18][49]. The computation time of the PC/BC model of V1 is significantly higher than other contour detection methods, because there are many convolution operations in this model. The two current state-of-the-art contour detection algorithms are HED [104] and CEND [105]. These two methods are both implemented by deep convolutional neural networks (CNNs) and executed on graphic processing units (GPU). Furthermore, the computation time of the two state-of-the-art methods are significantly lower than other contour detection methods operated in central processing units (CPU), as illustrated in Table 5-2. The computation time of the two state-of-the-art contour detection algorithms are mentioned in [104][105]. Therefore, if both the proposed texture gradient method and the texture gradient method used in the Pb algorithm [47] are implemented in GPU instead of CPU, the computation time of the both methods should be significantly reduced, and the advantage of the proposed method in computation time probably becomes negligible.

The value of each parameter in the proposed texture gradient method was determined by trial and error. Specifically, the proposed method was operated on training images with a set of manually defined default parameter values. Afterwards, each parameter was sequentially determined while all other parameter values were kept fixed according to the alteration of F-score on these training images. The sequence of the justification of each parameter value is from parameters related to obtaining filtered responses to computing the difference of sum of these filtered responses. Experiments were conducted to assess the effect of altering the values of six key parameters in the proposed algorithm and the proposed method integrated with the PC/BC model of V1 [39]. These results are shown in Table 5-3. It can be seen that significant changes in

FBORIENT, *TGORIENT*, and *WT* have little effect on the texture boundary detection result (F-score no lower than 0.57), but the results are more sensitive to the parameter: σ_{pb} , EL , and σ_G . It can be seen that the proposed algorithm integrated with the PC/BC model of V1 [39] is robust to changes in these parameters, with the F-score remaining 0.63 or 0.64 for all combinations of parameters tested. Hence, the good performance of the integrated algorithm is not the result of using parameters with a set of special values. In other words, the proposed texture gradient method combined with the PC/BC model of V1 is robust.

Table 5-2 The computation time and F-score (defined in equation 2.1) of different methods when tested on 100 images from the BSDS300 benchmark [36].

Method	F-score	Computation time (seconds)
Proposed TG	0.58	338.63
Pb (texture) [47]	0.58	538.32/928.35
Proposed TG + Pb (brightness)	0.63	440.39
Pb (brightness+texture) [47]	0.63	664.72
PC/BC [39] + proposed TG	0.64	96221.52
SCG [49]	0.71	11688.64
gPb-owt-ucm [18]	0.71	5858.34
CEDN [104]	0.78	30
HED [105]	0.78	40

5.4 Discussion

Inspired by the texture gradient method used in the Pb algorithm [47], a simplified and biologically-inspired texture gradient method based on local discontinuities in textures is proposed to locate texture boundaries within grayscale images. The proposed method is a considerable simplification compared to the Pb algorithm. First, the proposed method uses odd-symmetric and even-symmetric filters at only one scale rather than at two scales. Second, the proposed method directly employ the filter responses to compute the texture gradient rather than using labels of universal textons generated by applying the K-Means clustering algorithm to all filter responses from all images in the training set as used in the Pb algorithm. Third, uniform half disc filters are used in the Pb algorithm to compute texture gradient; instead, the proposed method employed half Gaussian filters in order to avoid the need to use Savitzky-Golay filter [47] smoothing on texture gradients and bring a better trade-off between using large regions to calculate

the texture gradient while maintaining good localization of boundaries. The proposed method is also a biologically-inspired method, because the filtered responses used to represent the structure of texture can be interpreted as the simulation of neuron responses in V1 [93] and the location of texture boundaries achieved by computing the difference of sum of filtered responses between two semi-circular Gaussian regions can be interpreted as the simulation of illusory contour detection in V2 [42][43].

Table 5-3 Effect of altering the value of six key parameters used in the proposed algorithm (integrated with the PC/BC model of V1 [39]) on the F-score (defined in equation 2.1) from the BSDS300 benchmark [36].

Parameter	Description	Standard Value	Altered Value	F-score
σ_{fb}	Standard deviation of second order derivatives of Gaussian filters.	0.3	0.6	0.57 (0.64)
			0.1	0.56 (0.63)
EL	Aspect ratio of Gaussian envelop for second order derivatives of Gaussian filters	3	5	0.57 (0.64)
			1	0.56 (0.64)
FBORIENT	Number of evenly spaced orientations for first and second order derivatives of Gaussian filters	8	12	0.57 (0.64)
			6	0.57 (0.64)
σ_G	Standard deviation of half Gaussian filters used to compute texture gradient	6	12	0.57 (0.63)
			3	0.55 (0.63)
TGORIENT	Number of evenly spaced orientations of half Gaussian filters	8	12	0.57 (0.64)
			6	0.57 (0.64)
WT	Range of value of texture gradient in the proposed method	2	4	0.58 (0.63)
			1	0.58 (0.63)

Despite the proposed simplifications, the performance of the proposed method in detecting texture boundaries is the same as the texture gradient method used in the Pb algorithm when tested with the BSDS300 benchmark [36]. Furthermore, a combination of the proposed texture gradient method with the PC/BC model of V1 [39] enables boundaries defined by local discontinuities in both intensity and texture to be located.

When tested with the BSDS300 benchmark, the combined algorithm slightly outperforms the recent near state-of-the-art contour detection method (the Pb algorithm) when this method is also restricted to using only local discontinuities in intensity and texture at a single-scale within grayscale images.

The proposed simplified texture gradient method is also reasonable and effective. To locate texture boundaries, an algorithm needs to firstly represent the same type of texture within an image, and then work out which pixels are located at boundaries between two different types of texture. The Pb algorithm [47] utilizes textons, which are the cluster centre of filtered responses across multiple orientations, to represent the same type of texture. In other words, pixels that have similar filtered responses across multiple orientations within an image are supposed to belong to the same type of texture. In comparison, the proposed algorithm directly uses filtered responses at one orientation within the image to represent the type of texture. This works because in circumstances where a smooth (non-textured surface) neighbours a non-smooth (textured area), the proposed method will generate weak responses at all orientations on one side of the boundary and strong responses at one orientation at least on the other side of the boundary. Second, in circumstances where two textured areas meet, different types of texture have different dominant orientations, and the proposed method will generate strong filtered responses at two different orientations to represent the two types of texture.

Chapter 6 Contour Detection Refined by A Sparse Reconstruction-based Discrimination Method

6.1 Introduction

Inspired by the neurophysiological evidence for sparse coding in the human/animal visual system [33], sparse representation techniques based on the hypothesis that natural images allow a sparse decomposition in some redundant basis (or dictionary), have been widely used for different image processing tasks, such as image denoising [141], image restoration [142], texture classification [143], face recognition [144] and contour detection [48][49]. As described in section 2.7 and [48], the sparse reconstruction-based discrimination (SRBD) method [34] has been applied in [48] to refining contour detection results derived from the Canny edge detector [8] producing results that are significantly improved compared to those of the unrefined Canny detector.

In this Chapter, a modified version of the SRBD method is applied to refining the contour detection results produced by three other previously proposed algorithms. These three algorithms are the Colour PC/BC model of V1 (c-PC/BC) proposed in Chapter 3, gPb-owt-ucm [18] described in section 2.6, and SCG [49]. Section 6.2 describes the procedure used to apply the modified version of SRBD to refining the results derived from the three contour detectors, highlighting the modified steps compared to [48]. The Results section evaluates the performance of each of the three refined contour detection methods using the BSDS300 [36] and BSDS500 [18] datasets. Section 6.4 discusses the achievement of the modified version of SRBD and possible future research directions.

6.2 Methods

The sparse representation technique can be summarized as: an input signal x admits a sparse approximation over a dictionary D comprised of many elements, when a linear combination of only a few elements from D can be found that approximates the original signal x . The sparse reconstruction error is defined by the difference between the input signal x and the approximation of the input signal x using the dictionary D . The dictionary D can be learnt from image patches. If two classifications of dictionary D are used to approximate the same input signal, the input signal is given the same classification as the dictionary that produces the smaller reconstruction error (the better approximation).

In order to refine the three previously proposed contour detection methods: c-PC/BC, gPb-owt-ucm [18] and SCG [49], the modified version of SRBD is employed to suppress the pb values of contour pixels it classifies as false contours. The classification of true and false contour pixels within these three contour detection results is based on the comparison between two classes of sparse reconstruction error of the centre pixel of each test image patch within input images. To calculate the two classes of sparse reconstruction error, two classes of sparse representations need to be derived from encoding each test image patch using the two classes of dictionaries belonging to the true and false contour. The two classes of dictionaries are derived from learning image patches extracted from the training images in the BSDS300 benchmark [36] whose centre pixels belonging to the true and false contours identified by human beings. Hence the two classes of training image patches belonging to the true and false contour need to be extracted first. In summary, the application of SRBD to refine the three previously proposed contour detection methods can be described in four stages: training patches extraction, dictionary learning, sparse reconstruction error computation, and false contours suppression. The procedure of the modified version of SRBD is illustrated in Figure 6-1.

6.2.1 Training Patches Extraction

Only if the centre pixel of a patch satisfies the following condition is the surrounding patch extracted. The centre pixel is identified as an ‘effective’ contour by the previously proposed boundary-detection algorithm that is being refined. Afterwards, the extracted patch x^T is classified as a true contour if the centre pixel of the patch is selected as a contour pixel by at least one human subject [36]. In contrast, the extracted patch x^F is classified as a false contour if none of the $r \times r$ pixels surrounding the centre pixel are selected as a contour pixel by any of the human subjects [36]. Because c-PC/BC, gPb-owt-ucm and SCG all produce a continuous value of probability of boundary (pb) at each pixel, ‘effective’ contours are defined as those pixels with pb values larger than a threshold $T_1=0.01$. 1000 patches that satisfied these conditions were selected, at random, from each training image. As there are 200 training images this gave a total of 200000 training patches for each dictionary.

Each training image patch x^T or x^F belonging to the true or false contour with size $p \times p$ pixels was converted to a column vector with size $n=p^2$ pixels. If those training images were chromatic, image patches extracted from the red, green, and blue channel of the RGB colour space were concatenated to a single column vector with the size $3n$ pixels. It has been found that utilizing multiple sizes of patches and resolutions of images can improve the performance of the refinement [34]. Mairal et al. [48] used $p=5, 7, 9, 11, 15, 19, 23$ for both full scale and half scale images to refine the Canny edge detector [8]. However, in this modified version, only some of these sizes of patches were used. $p=7, 9, 11, 15, 19$ pixels with $r=5$ pixels for full scale images and $p=7, 9, 11$ pixels with $r=3$ pixels for half scale images were used with the three algorithms. As a result, there were overall $N=16$ (8 sizes $\times$ 2 classes) sets of training patches in the proposed method.

6.2.2 Dictionaries Learning

After $N=16$ sets of training patches belonging to 2 classes has been extracted, the K-SVD algorithm [32] is then applied to learning the $N/2=8$ sets of dictionary D^T and D^F with a fixed number of $k=256$ elements and a fixed sparsity factor $L=12$, using $N/2=8$ sets of x^T and x^F patches (column vectors). The number of iterations for the K-SVD algorithm was set to 1000. The problem of dictionary learning is to work out an optimal dictionary D^T or D^F which can minimize the reconstruction error given a fixed sparsity constraint, as described by the following equation:

$$\min_{\alpha^T, D^T} \sum_{l=1}^{200000} \|x_l^T - D^T \alpha_l^T\|_2^2 \text{ s.t. } \|\alpha_l^T\|_0 \leq L \quad (6.1)$$

$$\min_{\alpha^F, D^F} \sum_{l=1}^{200000} \|x_l^F - D^F \alpha_l^F\|_2^2 \text{ s.t. } \|\alpha_l^F\|_0 \leq L \quad (6.2)$$

where $\|\cdot\|_2$ represents the Frobenius l^2 -norm; x_l^T and x_l^F represents a training image patch converted to a column vector with the size n pixels belonging to the true and false contour class; D^T and D^F represents the dictionary to be learnt with the size $n \times k$ belonging to the true and false contour class. The k -element column vector α_l^T and α_l^F is the sparse representation for the l -th patch using the dictionaries D^T and D^F belonging to the true and false contour class; and $\|\cdot\|_0$ denotes the number of non-zero elements in a vector.

When SRBD was applied to refining the Canny edge detector by Mairal et al. [48], all 28 (14 sizes $\times$ 2 classes) dictionaries belonging to the true and false contour classes were learnt by using both the K-SVD algorithm [32] and the discriminative training method [48]. The dictionaries used in [48] had a fixed number of 256 elements. The number of iterations for the K-SVD algorithm was set to 25 and a series of sparsity factors ranging from 1 to 15 were used.

6.2.3 Sparse Reconstruction Error Computation

After 8 sets of dictionary D^T and D^F have been learnt from the training images in the BSDS300 benchmark [36], the sparse reconstruction error then needs to be determined. Hence, the 8 sets of test image patches need to be extracted from each test image at all locations where the centre pixel had been identified as an ‘effective’ contour by the previously proposed boundary-detection algorithm that was being refined. The 8 sets of test image patches have the same 8 sizes as defined in the procedure of extracting training images patches. Afterwards, the 8 sets of dictionaries D^T and D^F learnt from equation 6.1 and 6.2 as well as 8 sets of test image patches (column vectors x) become inputs to the OMP algorithm [112], which is applied to generate 8 sets of sparse representations α^{T*} and α^{F*} of the test image minimizing the reconstruction errors using the same sparsity factor $L=12$. This procedure is as follows:

$$\alpha^{T*}(x, D^T) \equiv \arg \min_{\alpha} \|x - D^T \alpha\|_2^2, s. t. \|\alpha_l\|_0 \leq L \quad (6.3)$$

$$\alpha^{F*}(x, D^F) \equiv \arg \min_{\alpha} \|x - D^F \alpha\|_2^2, s. t. \|\alpha_l\|_0 \leq L \quad (6.4)$$

where $\|\cdot\|_2$ represents the Frobenius l^2 -norm; x represents a test image patch converted to a column vector with the size n pixels; D^T and D^F represents the dictionary with the size $n \times k$ belonging to the true and false contour class learnt from equation 6.1 and 6.2. The k -element column vector $\alpha^{T*}(x, D^T)$ and $\alpha^{F*}(x, D^F)$ is the sparse representation for the l -th patch using the dictionary D^T and D^F belonging to the true and false contour class; $\|\cdot\|_0$ denotes the number of non-zero elements in a vector.

Afterwards, 8 sets of sparse reconstruction error $R^{T*}(x, D^T, \alpha^{T*})$ and $R^{F*}(x, D^F, \alpha^{F*})$ can be straightforwardly obtained by:

$$R^{T*}(x, D^T, \alpha^{T*}) \equiv \|x - D^T \alpha^{T*}(x, D^T)\|_2^2 \quad (6.5)$$

$$R^{F*}(x, D^F, \alpha^{F*}) \equiv \|x - D^F \alpha^{F*}(x, D^F)\|_2^2 \quad (6.6)$$

where $R^{T*}(x, D^T, \alpha^{T*})$ and $R^{F*}(x, D^F, \alpha^{F*})$ represents sparse reconstruction error obtained from computing the difference between the test image patch and the encoding of this test image patch using the dictionary D^T or D^F with its corresponding sparse representations $\alpha^{T*}(x, D^T)$ and $\alpha^{F*}(x, D^F)$ belonging to the true and false contour class.

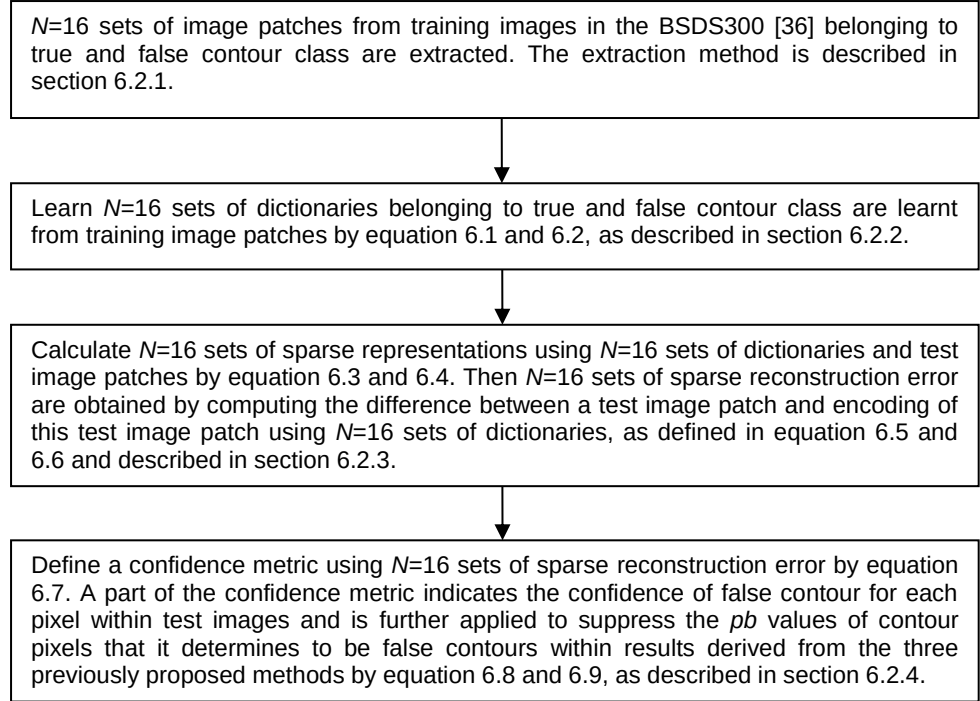


Figure 6-1 The procedure of the modified version of SRBD used to refine the three previously proposed contour detection methods.

6.2.4 False Contours Suppression

After $N/2=8$ sets of the sparse reconstruction error R^{T*} and R^{F*} have been computed on each test image patch, the modified version of SRBD was used to suppress the pb value of false contours within the results derived from the three previously proposed contour detection methods. The first step of this procedure was to define a confidence metric (CM):

$$CM = \frac{R^{T*} - R^{F*}}{R^{T*} + R^{F*}} \quad (6.7)$$

where R^{T*} and R^{F*} represent the sparse reconstruction errors obtained by encoding test patches using the true and false contour dictionaries. There are overall $N/2=8$ CM values obtained from multiple sizes of patches and resolutions of the image. All values of CM continuously range from -1 to 1. Specifically, positive values correspond to the belief that the centre pixel of the patch is a false contour. In contrast, negative values in CM indicate the confidence that the centre pixel of the test patch is a true contour. Only

positive values in CM were kept and negative values in CM were set to 0, to generate a confidence of false contour metric (CFM), as defined in:

$$v(x, y) = \begin{cases} u(x, y), & u(x, y) > 0 \\ 0, & u(x, y) \leq 0 \end{cases}$$

for $\forall v(x, y) \in CFM$, and $\forall u(x, y) \in CM$ (6.8)

where $u(x, y)$ represents each value in CM and $v(x, y)$ represents the corresponding value with same location in CFM . Afterwards, $N/2=8$ CFM were summed up and normalized by a function G , which clipped values $w(x, y)$ within the aggregated CFM at a threshold T_2 and then normalized these values by T_2 , as defined in:

$$G(w(x, y), T_2) = \begin{cases} w(x, y)/T_2, & w(x, y) < T_2 \\ 1, & w(x, y) \geq T_2 \end{cases}$$

for $\forall w(x, y) \in \sum_{i=1}^8 CFM_i$ and $\forall G(w(x, y), T_2) \in NACFM$ (6.9)

where $w(x, y)$ represents each value in the aggregated CFM ; $NACFM$ whose values are the normalized values in the aggregated CFM by the function G still indicates the confidence of false contour and is used to suppress the pb value of false contour pixels within an image. The value of T_2 was set to 0.15, 0.65, and 1.15 for c-PC/BC, gPb-owt-ucm, and SCG respectively.

Finally, $1 - NACFM$ was multiplied by the corresponding output (pb) of the boundary-detection algorithm that was being refined. The proposed version of SRBD thus acts to suppress the pb value of contour pixels it determines to be false contours. When gPb-owt-ucm and SCG were being refined, the output that was suppressed was the final result generated by these two algorithms. However, for c-PC/BC a slightly different approach was used. The PC/BC algorithm simulates neural activity in V1. The output of c-PC/BC that was suppressed was the prediction neuron responses across multiple orientations which indicated discontinuities in colour and luminance. Afterwards, the prediction neuron responses were used to linearly reconstruct the final contour detection result as defined in equation 2.7.

In contrast, because Mairal et al. [48] used a series of sparsity factors in the procedure of dictionary learning, a series of dictionaries was obtained for each different size of image patches based on the series of sparsity factor. Hence, the SRBD [48] can output a pair of curves belonging to the true and false contour class for each different size of image patches plotted by a series of reconstruction errors using the series of dictionaries as a function of the series of sparsity constraint. Afterwards, all curves generated by

multiple sizes of reconstruction errors were combined together to generate a feature vector for classification of true and false contours by a logistic regression classifier or a linear SVM.

In summary, the main difference between the SRBD method [48] used to refine Canny [8] and the modified version of the SRBD method used to refine the three previously proposed methods are in the following aspects:

- **The multiple sizes of image patches extracted from multiple resolution of training and test images.** The SRBD [48] used $p=5, 7, 9, 11, 15, 19, 23$ for both full scale and half scale images to refine the Canny edge detector [8]. Whereas, the proposed modified version of SRBD only employed some of these sizes of patches including $p=7, 9, 11, 15, 19$ pixels for full scale images and $p=7, 9, 11$ pixels for half scale images.
- **The sparsity constraint used in the dictionary learning procedure.** The SRBD [48] used to refine Canny defined a discriminative dictionary training method and used a series of sparsity constraint to learn a series of dictionaries for each different size of training image patches. In contrast, the proposed modified version of SRBD is simpler, as it does not use the discriminative dictionary learning method and only uses one sparsity constraint to generate one dictionary for each different size of training image patch.
- **The sparse reconstruction error employed to separate false contours from true contours within test images.** The SRBD [48] used to refine Canny can output a pair of curves for each different size of image patches plotted by two series of reconstruction errors using the corresponding series of dictionaries belonging to the true and false contour class as a function of the corresponding series of sparsity constraint. Afterwards, all curves generated by multiple sizes of reconstruction errors were combined together to generate a feature vector for the separation of false contours from true contours by a logistic regression classifier or a linear SVM. In contrast, the proposed modified version of SRBD defined a confidence metric calculated from multiple sizes of sparse reconstruction errors. The positive part of the confidence metric indicates the confidence of a false contour at each pixel within a test image and is employed to suppress the pb value of contour pixels that it determines to be false contours.

6.3 Results

6.3.1 Contour Detection Results

Contour detection results derived from c-PC/BC, gPb-owt-ucm [18], and SCG [49] were all generated using the code provided by the authors of these algorithms. Afterwards, the result derived from each algorithm was refined by the modified version of the SRBD method proposed in section 6.2. The contour detection results produced from the three previously proposed methods with and without refinement by the proposed modified version of SRBD are both qualitatively and quantitatively evaluated by the BSDS300 [36] and BSDS500 [18] benchmark.

As shown in Figure 6-2, Figure 6-3 and Table 6-1, the ODS and OIS of c-PC/BC is slightly increased using our proposed SRBD method, and the AP is significantly increased. The result derived from a recent near state-of-the-art algorithms, gPb-owt-ucm [18], is slightly improved after the refinement, with the ODS increased from 0.71 to 0.72 and the AP increased from 0.73 to 0.76, when evaluated by the BSDS300 benchmark; however, only the AP is significantly increased from 0.73 to 0.77 for the BSDS500 benchmark. The performance of the other leading algorithm, SCG [49], is also slightly improved in its refined version, with the ODS increased from 0.71 to 0.72, the OIS increased from 0.73 to 0.74 and the AP increased from 0.75 to 0.76, when evaluated by the BSDS300 benchmark. Only the AP is slightly increased from 0.77 to 0.78 for the BSDS500 benchmark.

With respect to the reason why the performance of the three previously proposed contour detection methods can be improved (as the previous quantitative evaluation results illustrate), a qualitative evaluation of the results produced by the colour PC/BC model of V1 with and without the refinement by the proposed SRBD method for four image from the BSDS300 dataset [36] are used as an example for explanation. As shown in Figure 6-4, some contours marked by the yellow rectangle within the results derived from c-PC/BC are contour pixels that are not identified by human observers and the *pb* value of which are accurately suppressed by the proposed SRBD method. In other words, the *pb* value of some contour pixels within the result derived from c-PC/BC or other methods that are determined as highly confident false contour pixels by the proposed SRBD should be significantly decreased in the refined contour detection result. Moreover, Figure 6-4 also illustrates that the pixels identified as false contour

pixels by the proposed SRBD are almost exclusively not identified as true contours by human observers. Hence, the proposed SRBD has no influence on most of true contour pixels identified by humans. As a result, the *pb* value of all true contour pixels produced by c-PC/BC or other methods should almost have no change in the refined result.

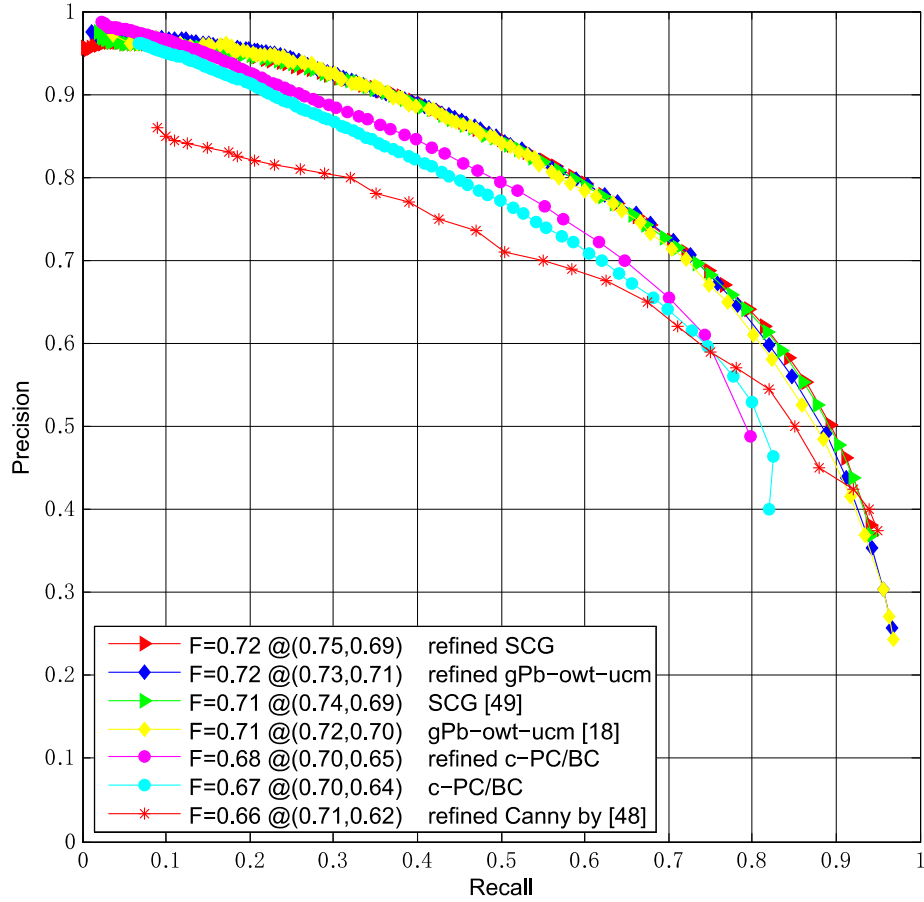


Figure 6-2 Performances of contour detection algorithms with and without refinement by the proposed SRBD evaluated by the precision-recall curve of the F-score defined in section 2.4.1 from the BSDS300 Benchmark [36].

As previously mentioned in section 2.4.1, the BSDS300/500 benchmark applies a set of thresholds ranging from 0 to 1 to convert the contour detection result into a binary map and then output a precision-recall curve at different threshold values. The F-score/ODS or OIS of the method being tested is the highest F-score value in a fixed threshold for all images or a set of optimal thresholds for each image. Because the *pb* value of some false contour pixels (most of true contour pixels) within the contour detection results derived from c-PC/BC or other methods being refined are significantly decreased (unchange) by the proposed SRBD, the precision (recall) values at the majority of threshold values should be increased (should almost have no change) compared with the

result without refinement. In other words, some false contour pixels are no longer treated as a contour pixel at the majority of threshold values and most of the true contour pixels are unaffected by the proposed SRBD. As shown in Table 6-1, the quantitative evaluation results are consistent with the observation. Specifically, the average precision (AP) values are all increased for the comparison between each of the three previously proposed methods with and without refinement. Also, the recall values have only a slight or no decrease at each threshold value for each of the three methods. However, the recall value may be decreased or increased in the highest F-score of different refined results because of optimal threshold values may have been changed. Therefore, the F-score/ODS and OIS are all increased for the comparison between each of the three previously proposed methods with and without refinement, although some of these increments are negligible.

The value of each parameter in the proposed modified version of SRBD was determined by trial and error. Specifically, the modified SRBD was operated on training images with a set of manually defined default parameter values. Afterwards, each parameter was sequentially justified while all other parameter values were kept fixed according to the alteration of F-score on these training images. The sequence of the justification of each parameter value is from parameters related to training image patches extraction, dictionary learning, sparse reconstruction error computation, to suppression of contour pixels if the modified SRBD determines as false contours within results derived from the three previously proposed contour detection methods.

6.3.2 Failure Cases Discussion

As described in section 6.2.1, image patches are extracted from the red, green and blue channel of the RGB colour space if the training images are chromatic. The CIELAB colour space and the neurophysiologically-inspired colour space defined in equations 3.1, 3.2 and 3.3 in Chapter 3 have also been tried. The quantitative evaluation results indicate that the RGB colour space is the best one for the proposed SRBD method (refined gPb-owt-ucm [18] ODS=0.72) compared with the CIELAB colour space (refined gPb-owt-ucm [18] ODS=0.71) and the neurophysiologically-inspired colour space (refined gPb-owt-ucm [18] ODS=0.71). Instead of randomly selecting these 1000 patches in section 6.2.1, other selection methods were also tried. Specifically, all patches that satisfied those conditions were firstly sorted in a descending order according to the *pb* value (contour detection results derived from the three previously

proposed method) of its centre pixel, then 1000 patches classified as true contour and false contour were selected from either the first or the last 1000 patches in this descending order. The quantitative evaluation results indicated that the proposed refinement on each of the three contour detection algorithms was not improved by selecting 1000 patches either of these ways.

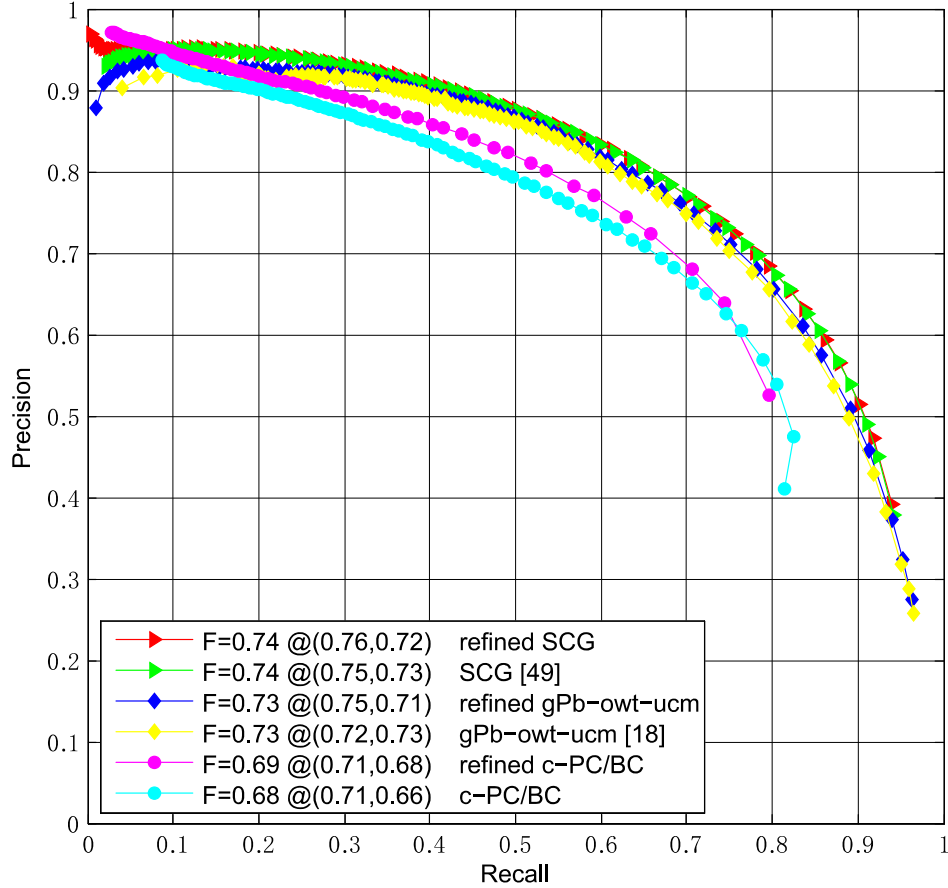


Figure 6-3 Performances of contour detection algorithms with and without refinement by the proposed SRBD evaluated by the precision-recall curve of the ODS (F-score) defined in section 2.4.1 from the BSDS500 Benchmark [36].

Furthermore, in section 6.2.4, the confidence of false contour metric (CFM) obtained from multiple sizes of patches and resolution of the image have also been modified by only keeping the maximum values across multiple CFMs instead of summing up these multiple CFMs and then normalized by a function G at a threshold T_2 . Then $1 -$ the maximum CFM can be directly utilized to multiply with the corresponding output of the contour detection algorithm that was being refined. However, the refinement result of our proposed SRBD method integrating multiple CFMs in this new straightforward way (refined gPb-owt-ucm [18] ODS=0.71) are worse than the originally proposed

Table 6-1 The ODS, OIS and AP of different contour detection methods.

The quantitative evaluation measures include: the F-score (defined in equation 2.1) with a fixed threshold for all test images (ODS), the F-score with an optimal threshold per image (OIS), and the average precision (AP) for the full range of recall values.

Methods	BSDS300			BSDS500		
	ODS	OIS	AP	ODS	OIS	AP
Human	0.79	0.79	—	0.80	0.80	—
Refined SCG	0.72	0.74	0.76	0.74	0.76	0.78
SCG [49]	0.71	0.73	0.75	0.74	0.76	0.77
Refined gPb-owt-ucm	0.72	0.74	0.76	0.73	0.76	0.77
gPb-owt-ucm [18]	0.71	0.74	0.73	0.73	0.76	0.73
Refined c-PC/BC	0.68	0.69	0.63	0.69	0.71	0.64
c-PC/BC	0.67	0.68	0.59	0.68	0.70	0.58



Figure 6-4 Results of contour detection derived from the colour PC/BC model of V1 with and without refinement by the proposed SRBD for four images from BSDS300 [36].

Contour detection results are produced by (from top to bottom): human, c-PC/BC, and refined c-PC/BC. Some contours, marked in the yellow rectangle within the result obtained from c-PC/BC, are the contour pixels that are not identified by human observers and the pb value of which are accurately suppressed by the proposed modified version of SRBD in the refined c-PC/BC result. Most true contour pixels also identified by human within the result derived from c-PC/BC are unaffected by the proposed modified version of SRBD.

combination method defined in step 4 in section 6.2 (refined gPb-owt-ucm [18] ODS=0.72) when evaluated by the BSDS300 benchmark [36].

6.4 Discussion

Inspired by the neurophysiological evidence for sparse coding in the human/animal visual system [33], the sparse reconstruction-based discrimination (SRBD) method was previously proposed by Mairal et al. [34] and utilized two dictionaries to place image patches into two classes: edge and non-edge. Afterwards, Mairal et al. [48] applied SRBD to improve the performance of the Canny edge detector [8]. However, given the poor performance of the Canny method in locating salient boundaries, the refined method still produced results that fell far short of state-of-the-art contour detection methods.

In this Chapter, a modified version of the SRBD method is applied to refining the contour detection results generated by three different boundary-detection algorithms: the colour Predictive Coding/Biased Competition Model of V1 proposed in Chapter 3 and the two recent near state-of-the-art algorithms: gPb-owt-ucm [18] and Sparse Coding Gradients method [49]. Evaluation using the BSDS300 [36] and BSDS500 [18] datasets indicate that for all three algorithms boundary detection is at least slightly improved by the proposed refinement.

Because the source code of the discriminative training method [48] is not available, this method is not able to be utilized to increase the discriminative power of each pair of dictionaries in the proposed version of the SRBD. Therefore, it is possible that the contour detection results obtained from different boundary-detection algorithms refined by the proposed version of the SRBD can be further improved if the discriminative training method [48] can be employed in the dictionary learning stage.

As previously mentioned, the PC/BC model of V1 [39] can be interpreted as a sparse coding model, which uses sparse representations (prediction neuron responses) to encode an image employing a set of redundant basis or a set of elements in a dictionary (first order derivative of Gaussian filters). Therefore, some elements from other types of the dictionaries may also be introduced in the PC/BC model to generate sparse representations (prediction neuron responses) of the corresponding types of visual information. If the proposed version of SRBD method is applied to refining the contour

detection results obtained from the original PC/BC model of V1 [39], some elements in the dictionaries belonging to the true and false contour classes generated in section 6.2.2 can be introduced in the PC/BC model of V1 [39] to generate prediction neuron responses that represent the true contours and texture edges (false contours) which are not accurately distinguished by the mechanism of the lateral connections defined in [39]. Afterwards, the performance of the PC/BC model of V1 may be improved because the new version of the PC/BC model of V1 can distinguish the prediction neuron responses representing true contours from texture edges more accurately.

Chapter 7 Conclusions

7.1 Summary of Findings

The first aim of the research was to extend a comprehensive neurophysiologically-inspired method: the PC/BC model of V1 [39] devised to locate boundaries defined by intensity discontinuities to improve its performance of contour detection by locating more contours that may be recognized by humans in an image. Such extensions should also be inspired by and/or simulate some neurophysiological activities. Such improvement was expected to be indicated when evaluated by the BSDS300 benchmark [36]. The colour PC/BC model of V1 proposed in Chapter 3, which is an extended version of the PC/BC model of V1 [39], is able to locate boundaries defined by local discontinuities in both colour and intensity, and hence, obtain better contour detection results compared to the original PC/BC model of V1 [39]. This extension is inspired by neurophysiological data from single neurons in macaque V1 [40][41]. The colour PC/BC model of V1 can successfully simulate the response properties of orientation-selective double-opponent neuron and complex cells in macaque V1 [40][41]. Furthermore, when evaluated by the BSDS300 benchmark [36], the performance is slightly better than some recently proposed contour detection algorithms which use more local cues (not only boundaries defined by local discontinuities in intensity and colour) and/or require a complicated training procedure.

Inspired by the neurophysiological activity related to texture boundaries [42][43] and corners [44][45][46] in V2 in the human/animal visual system, two extended versions of the PC/BC model of V1: the PC/BC model of V2 for texture boundary detection and corner detection, were devised in Chapter 4. The two versions of the PC/BC model of V2 were integrated with the PC/BC model of V1 respectively to generate an integrated contour detection result expected to improve the performance of contour detection under the framework of the PC/BC model when evaluated by the BSDS300 benchmark [36]. The integrated model for texture boundary detection can locate some boundaries defined by texture discontinuities, but these boundaries are neither accurate nor sufficient enough to improve the performance. The root cause of the worse result of the contour detection derived from this integrated model is possibly that the structure of texture represented in this integrated PC/BC model is not good enough. The integrated model for corner detection can enhance some weak contours that are defined by insignificant intensity discontinuities, because the V1 neuron responses consistent with

a corner can be strengthened by the feedback from the V2 neuron responses. However this feedback also leads to the inefficiency of texture suppression mechanism in this integrated model, which results in the worse contour detection results compared to the original PC/BC model of V1 [39].

The second aim of the research was to simplify the first stage of a recent near state-of-the-art contour detection algorithm to a biologically-inspired method, which can reduce the computation time of the first stage of this algorithm without generating a worse result. In Chapter 5, a simplified and biologically-inspired version of the texture gradient method used in the Pb algorithm [47] was proposed. The Pb algorithm is the first stage of a recent near state-of-the-art algorithm gPb-owt-ucm [18]. The main complexity advantage of the proposed texture gradient method is that this proposed method directly utilized the filter responses to represent the structure of texture without employing the non-biologically-inspired K-Means algorithm to cluster these filter responses. This results in the computation time of this proposed method can be significantly reduced compared to the Pb texture gradient method [47]. Furthermore, the proposed method can be interpreted as a biologically-inspired method. Despite the proposed simplifications, the proposed texture gradient method can still generate the same result as the Pb texture gradient method when evaluated by the BSDS300 benchmark [36].

Inspired by the neurophysiological evidence for sparse coding in the human/animal visual system [33], the sparse reconstruction-based discrimination (SRBD) method [34] has been applied in [48] to refine the contour detection results derived from the Canny edge detector [8]. In Chapter 6, a modified version of the SRBD method was applied to refine the contour detection results derived from the colour PC/BC model of V1 and the two recent near state-of-the-art contour detection algorithms: gPb-owt-ucm [18] and SCG [49]. The performance of the three refined contour detection results is at least slightly improved compared to the results derived from the corresponding original method when evaluated by both the BSDS300 [36] and the BSDS500 [18] benchmark. This is because the refinement procedure can suppress the *pb* value of some false contour pixels detected by the method being refined and almost have no influence on all true contour pixels detected by both the method being refined and human observers. Specifically, when the result derived from a refined method is quantitatively tested with both the BSDS300 [36] and the BSDS500 [18] benchmark, the precision value can

increase without decreasing its recall value at most of threshold values and therefore this can increase its F-score/ODS, OIS and AP, despite some of these increments are negligible.

7.2 Implications of These Findings

The colour PC/BC model of V1 can simulate the response properties of orientation-selective double-opponent single neurons and complex cells in macaque V1 and have a slightly better performance of contour detection compared to some other recently proposed contour detection methods. This finding shows that the PC/BC model of V1 can be extended to simulate another neurophysiological activity in V1 related to contour detection well and successfully improve the performance of contour detection under the framework of the PC/BC model. Furthermore, this finding increases the possibility of that a neurophysiological-inspired model has advantages of modelling the features which people may use to recognize contours compared to other kinds of computer vision methods. This finding also indicates that there might be orientation-selective double-opponent single neurons in the human visual system being related to the contour detection behaviour of locating boundaries defined by local colour discontinuities.

The two versions of the PC/BC model of V2 for texture boundary detection and corner detection are both integrated with the PC/BC model of V1 to generate the worse contour detection results compared to the original PC/BC model of V1. Although the integrated contour detection results are worse, the PC/BC model of V2 for texture boundary detection can locate some texture boundaries which cannot be located by the original PC/BC model of V1 and the PC/BC model of V2 for corner detection is also able to enhance some weak contours obtained from the PC/BC model of V1 and defined by insignificant intensity discontinuities. This finding shows that the PC/BC model of V1 is able to be extended to simulate the neurophysiological activity in a higher cortical area and the PC/BC model simulating different cortical areas can be integrated to operate sequentially and recurrently. This finding also indicates that there might be some mechanisms in the human visual system that can represent texture information within an image accurately and sufficiently before locating texture boundaries and can efficiently suppress texture edges after both texture edges and boundary edges are enhanced by the feedback from neuron responses representing corners in V2.

A simplified and biologically-inspired version of the texture gradient method used in the

Pb algorithms (the first stage of the recent near state-of-the-art algorithm) is obtained. The proposed simplified and biologically-inspired texture gradient method directly utilizes filter responses to represent the structure of texture and is able to reduce the computation time without generating a worse contour detection result compared to the Pb texture gradient method, which can also simulate the illusory contour interpreted as texture boundary detection in V2. This finding shows that the structure of texture within a natural image can also be represented well to some extent by using a set of linear filter responses instead of employing some complicated and time-consuming statistical algorithms. Furthermore, this finding also indicates that the texture boundary detection in V2 inspired the proposed simplified texture gradient method is more likely to related to the contour detection in the human/animal visual system and the proposed method simulates the texture boundary detection in V2 well.

A modified version of the SRBD method inspired by the neurophysiological evidence for sparse coding in the human/animal visual system is applied to refine the contour detection result derived from three contour detection algorithms. These three algorithms include the colour PC/BC model of V1 and two recent near state-of-the-art algorithms: gPb-owt-ucm and SCG. All these three contour detection algorithms can be at least slightly improved by the proposed refinement. This finding shows that the proposed version of SRBD method can, theoretically, be utilized to refine any category of contour detection method and improve their performance by suppressing the *pb* value of false contour pixels within their results. This finding also means the sparse coding activity in the human/animal visual system inspired the proposed modified version of the SRBD method is more likely to related to contour detection in the human/animal visual system, and the proposed algorithm can simulate the sparse coding activity in the human/animal visual system well.

7.3 Limitations of Research

The colour PC/BC model of V1 is inspired by neurophysiological data from single neurons in macaque V1. In this neurophysiological data, the orientation-selective double-opponent neurons are presumed to play an important role in locating boundaries defined by colour discontinuities. However, there is still controversy regarding how the colour information is processed in the human/animal visual system. Some researchers believe that neurons in the human/animal visual system which process colour information are poorly orientation-selective according to their neurophysiological data

[124][145]. Furthermore, the colour PC/BC model of V1 employed the same mechanisms to locate boundaries defined by local intensity and colour discontinuities. In contrast to the good understanding of different neural mechanisms used to process luminance information in V1, research of the neural mechanisms used to process colour information is still at an initial stage. It is possible that the same mechanism is utilized in the human visual system to process both luminance and colour information as simulated in the colour PC/BC model of V1. But it is also possible that this assumption is completely wrong.

The PC/BC models of V2 for texture boundary detection and corner detection are inspired by the corresponding neurophysiological activity found in the V2. However research into the response properties of neurons in V2 is just beginning and the neural mechanisms of texture boundary detection and corner detection are still unclear. It is also possible that texture boundary detection needs to be achieved by multiple visual cortical areas and corner detection is not directly related to contour detection behaviour in the human visual system. Furthermore, the bad performance of the two versions of the integrated PC/BC model also indicates that some new mechanisms related to texture boundary detection and corner detection need to be found in the human visual system and simulated in the integrated model to obtain a better contour detection result.

The proposed version of the texture gradient method directly utilizes linear filter responses to represent the structure of texture, can simulate the texture boundary detection in V2 [42][43] and generate a good-performing texture boundaries result. A better representation of the structure of texture in an algorithm can increase the possibility of locating texture boundaries more accurately. Although directly employing linear filter responses can reduce the computation time, it is obvious that the accuracy of the structure of texture represented by using only linear filter responses is limited. Therefore, it can be expected that using a sophisticated but time-consuming statistical method to represent the structure of texture within an image is still necessary if much more precise texture boundaries need to be located. Furthermore, the difference of the computation time between the texture gradient method used in the Pb algorithm and the proposed simplified method may become negligible if these two methods are operated on GPU instead of CPU.

The proposed version of the SRBD method inspired by the neurophysiological evidence for sparse coding in the human/animal visual system [33] can theoretically be applied to refine any category of contour detection methods and improve their performance by suppressing the pb value of false contour pixels and not affecting the pb value of true contour pixels within their results. However, it was found the effect of the refinement was significantly decreased for the two recent near state-of-the-art contour detection algorithms compared to the colour PC/BC model of V1. Hence it is probable that there is a limitation in the improvement that can be made by the proposed version of the SRBD method. In other words, if the performance of a contour detection algorithm is good enough such as the two current state-of-the-art deep CNNs methods [104][105], the results derived from these methods probably cannot be improved by the refinement made by SRBD. Moreover, the current proposed version of the SRBD method is devised to suppress the pb value of false contours generated by the contour detection method being refined. Therefore, if a contour detection method locates just a few false contours, the effect of the current version of SRBD will be limited. But a new version of SRBD probably can be devised to enhance the pb value of true contour pixels within the contour detection results generated by the contour detection algorithm to improve its performance.

7.4 Future Research Recommendations

As previously mentioned, the neural mechanism of colour processing are still being explored, therefore, the performance of the the colour PC/BC model of V1 may be improved if some new neural mechanisms related to colour boundary detection can be found in the human/animal visual system and then successfully simulated in this model.

The PC/BC model of V2 for texture boundary detection and corner detection integrated with the PC/BC model of V1 produce worse results. Therefore, if the texture information can be represented more accurately and sufficiently by a transformation of prediction neuron responses in the PC/BC model of V1, the performance of contour detection derived from the integrated PC/BC model for texture boundary detection may be improved. Likewise, if a new effective texture suppression mechanism can be defined in both the PC/BC model of V1 and V2, the performance of the integrated PC/BC model for corner detection may be improved.

The PC/BC model can also be applied to simulate other neurophysiological activity related to contour detection in the V1, V2 or any visual cortical area to try to improve the performance of contour detection under the framework of this neurophysiologically-inspired model.

The performance of the original PC/BC model of V1 [39] may be improved by using two dictionaries trained by the proposed version of the SRBD method. Because the PC/BC model of V1 [39] can be interpreted as a sparse coding model, which uses sparse representations (prediction neuron responses) to encode an image employing a set of redundant basis or a set of elements in a dictionary (first order derivative of Gaussian filters). Some additional elements may also be introduced in the PC/BC model of V1 to generate sparse representations (prediction neuron responses) of the corresponding visual information. Hence if the proposed version of the SRBD method is applied to refine the contour detection results derived from the PC/BC model of V1 [39], some elements in the two dictionaries belonging to the true and false contour classes generated by SRBD can be obtained. Afterwards, these elements can be introduced into the PC/BC model of V1 to generate prediction neuron responses that represent true contour and texture edges (false contour) which are not accurately distinguished in the original PC/BC model of V1 [39]. Then the performance of the PC/BC model of V1 may be improved because the new version of the PC/BC model of V1 can distinguish the prediction neuron responses representing true contours and texture edges more accurately.

Two recent near state-of-the-art algorithms: gPb-owt-ucm [18] and SCG [49] both use the Normalized Cuts algorithm [69] to combine global contour information with the contours derived from the local information. If the Normalized Cuts algorithm can be implemented by or replaced with a neurophysiologically-inspired model, the performance of the biologically-plausible contour detection algorithms may also be improved.

Top-down knowledge in the human visual system probably plays an important role in contour detection, because it is common to find some contours that can hardly (easily) be identified by a computer vision method based on local and/or global cues but can be (are not) recognized by human beings. However, few computer vision algorithms were devised to employ top-down knowledge to improve the contour detection performance.

Therefore, if a neurophysiologically-inspired model such as the PC/BC model or other kinds of computer vision algorithms can be utilized to successfully simulate how top-down knowledge is involved in the contour detection behaviour in the animal/human visual system, the performance of the computer vision algorithms related to contour detection can potentially be improved significantly.

Bibliography

- [1] Pal, N. R., Pal, S. K.: 'A review on image segmentation techniques', *Pat. Recog.*, 1993, vol. 26, pp. 1277-1294.
- [2] Walther, D. B., Chai, B., Caddigan, E., Beck, D. M., Fei-Fei, L.: 'Simple line drawings suffice for functional MRI decoding of natural scene categories', *Proc. National. Acad. Sci.*, April 2011, vol. 108, no. 23, pp. 9661-9666.
- [3] Destremes, F., Mignotte, M., Angers, J. F.: 'Localization of shapes using statistical models and stochastic optimization', *IEEE Tran. Pat. Anal. Mach. Intel.*, 2007, vol. 9, pp. 1603-1615.
- [4] Lowe, D. G.: 'Object recognition from local scale-invariant features', *Proc. Seventh. IEEE. Intl. Conf. Comput. Vis.*, 1999, vol. 2, pp. 1150-1157.
- [5] Comport, A., Marchand, É., Chaumette, F.: 'Robust model-based tracking for robot vision', *Proc. IEEE/RSJ Intl. Conf. Intel. Robot. Sys.*, 2004, vol.1, pp. 692-697.
- [6] Chalana, V., Linker, D. T., Haynor, D. R., Kim, Y.: 'A multiple active contour model for cardiac boundary detection on echocardiographic sequences', *IEEE. Tran. Med. Image.*, Jun 1996, vol. 15, pp. 290-298.
- [7] Benameur, S., Mignotte, M., Labelle, H., De Guise, J. A.: 'A hierarchical statistical modelling approach for the unsupervised 3-D biplanar reconstruction of the scoliotic spine', *IEEE Tran. Biomed. Eng.*, 2005, vol. 52, pp. 2041-2057.
- [8] Canny, J.: 'A Computational approach to edge detection', *IEEE Tran. Pat. Anal. Mach. Intel.*, 1986, vol. 8, pp. 679-698.
- [9] Marr, D., Hildreth, E.: 'Theory of edge detection', *Proc. R. Soc. Lond. B: Bio. Sci.*, 1980, vol. 207, pp. 187-217.
- [10] Tomasi, C., Manduchi, R.: 'Bilateral filtering for grey[50] and colour images', *Sixth. Intl. Conf. IEEE. Comput. Vis.*, 1998, pp. 839-846.
- [11] Lee, J. S., Haralick, R. M., and Shapiro, L. G.: 'Morphologic edge detection', *IEEE Jour. Robot. Auto.*, 1987, vol. 3, pp. 142-156.
- [12] Huang, J. S., Tseng, D. H.: 'Statistical theory of edge detection', *Comput. Vis. Graph. Image. Proc.*, 1988, vol.43, pp. 337-346.
- [13] Ojala, T., Pietikainen, M.: 'Unsupervised texture segmentation using feature distributions', *Pat. Recog. March.* 1999, vol.32, pp.477-486.
- [14] Adams, R., Bischof, L.: 'Seeded region growing', *IEEE Tran. Pat. Anal. Mach. Intel.*, Aug. 2002, vol. 16, no. 6, pp. 641-647.
- [15] Comaniciu, D., Meer, P.: 'Mean shift: A robust approach toward feature space analysis', *IEEE Tran. Pat. Anal. Mach. Intel.*, 2002, vol. 24, pp. 603-619.

- [16] Kass, M., Witkin, A., Terzopoulos, D.: 'Snakes: Active contour models', Intl. Jour. Comput. Vis., 1988, vol. 1, pp. 321-331.
- [17] Staib, L. H., Duncan, J. S.: 'Boundary finding with parametrically deformable models', IEEE Tran. Pattern. Anal. Mach. Intel., 1992, vol.11, pp. 1061-1075.
- [18] Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: 'Contour detection and hierarchical image segmentation', IEEE Tran. Pat. Anal. Mach. Intel., 2011, vol. 33, pp. 898-916.
- [19] Kovese, P.: 'Image features from phase congruency', Videre. Jour. Comput. Vis. Res., 1999, vol. 1, pp. 1-26.
- [20] Sappa, A. D.: 'Unsupervised contour closure algorithm for range image edge-based segmentation', IEEE Tran. Image. Proc., 2006, vol. 15, pp. 377-384.
- [21] Ren, X., Fowlkes, C. C., Malik, J.: 'Scale-invariant contour completion using conditional random fields', Computer Vision, Tenth. IEEE. Intl. Conf. Comput. Vis., 2005, vol. 2, pp. 1214-1221.
- [22] Stahl, J. S., Wang, S.: 'Globally optimal grouping for symmetric closed boundaries by combining boundary and region information', IEEE Tran. Pat. Anal. Mach. Intel., 2008, vol. 30, pp. 395-411.
- [23] Mohan, R., Nevatia, R.: 'Perceptual organization for scene segmentation and description', 1992, IEEE Tran. Pat. Anal. Mach. Intel., vol. 14, pp.616-635.
- [24] Field, D. J., Hayes, A., Hess, R. F.: 'Contour integration by the human visual system: evidence for a local "association field"', Vis. Res., 1993, vol. 33, pp. 173-193.
- [25] Petkov, N., Westenberg, M. A.: 'Suppression of contour perception by band-limited noise and its relation to nonclassical receptive field inhibition', Bio. Cyber., 2003, vol. 88, pp. 236-246.
- [26] Ursino, M., La Cara, G. E.: 'A model of contextual interactions and contour detection in primary visual cortex', 2004, Neur. Net., vol. 17, pp. 719-735.
- [27] Wei, H., Lang, B. and Zuo, Q.: 'Contour detection model with multi-scale integration based on non-classical receptive field, Neurocomput., 2013, vol. 103, pp.247-262.
- [28] Yang, K. F., Li, C. Y., Li, Y. J.: 'Multi feature-based surround inhibition improves contour detection in natural images', IEEE Tran. Image. Proc., 2014, vol. 23, pp. 5020-32.
- [29] Grigorescu, C., Petkov, N., Westenberg, M. A.: 'Contour detection based on nonclassical receptive field inhibition', IEEE Tran. Image. Proc., 2003, vol. 12, pp. 729-739.
- [30] Mignotte, M.: 'A biologically inspired framework for contour detection', Form. Pat. Anal. App., 2015, pp.1-17.

- [31] Róka, A., ÁdámCsapó, Reskó, B., Baranyi, P.: 'Edge detection model based on involuntary eye movements of the eye-retina system', *Acta. Polytech.Hunga.*, 2007, vol. 4, pp. 31-46.
- [32] Aharon, M., Elad, M., Bruckstein, A.M.: 'The K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representations', *IEEE Tran. Sig. Proc.*, 2006, vol. 54, pp. 4311-4312.
- [33] Vinje, W. E., Gallant, J. L.: 'Sparse coding and decorrelation in primary visual cortex during natural vision', *Science*, 2000, vol. 287, pp. 1273-1276.
- [34] Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A.: 'Discriminative learned dictionaries for local image analysis', *IEEE Confer. Comput. Vision. Pattern. Recog.*, 2008, pp. 1-8.
- [35] Yang, J., Price, B., Cohen, S., Lee, H., Yang, M. H.: 'Object contour detection with a fully convolutional encoder-decoder network', *IEEE Confer. Comput. Vis. Patt. Recog.*, 2016, pp. 193-202.
- [36] Martin, D., Fowlkes, C., Tal, D., Malik, J.: 'A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics', *Proc. Int. Conf. Comput. Vis.*, Jul 2001, vol. 2, pp. 416-423.
- [37] Haindl, M., Mikes, S.: 'Texture segmentation benchmark', *Intern. Confer. Patt. Recog.*, 2009, pp. 1-4.
- [38] Spratling, M. W.: 'Predictive coding as a model of response properties in cortical area V1', *Jour. Neurosci.*, March 2010, vol. 30, pp. 3531-3543.
- [39] Spratling, M. W.: 'Image segmentation using a sparse coding model of cortical area V1', *IEEE Tran. Image. Proc.*, Feb 2013, vol. 22, no. 4, pp. 1631-1643.
- [40] Johnson, E. N., Hawken, M. J., Shapley, R.: 'The spatial transformation of color in the primary visual cortex of the macaque monkey', *Nat. Neurosci.*, 2001, vol. 4, pp. 409-416.
- [41] Johnson, E. N., Hawken, M. J., Shapley, R.: 'The orientation selectivity of colour-responsive neurons in macaque V1', *Jour. Neurosci.*, 2008, vol. 28, pp. 8096-8106.
- [42] Bakin, J. S., Nakayama, K., Gilbert, C. D.: 'Visual responses in monkey areas V1 and V2 to three-dimensional surface configurations', *Jour. Neurosci.*, 2000, vol. 20, pp. 8188-8198.
- [43] Montaser-Kouhsari, L., Landy, M. S., Heeger, D. J., and Larsson, J.: 'Orientation-selective adaptation to illusory contours in human visual cortex', *Jour. Neurosci.*, 2007, vol. 27, pp. 2186-2195.
- [44] Ito, M., Komatsu, H.: 'Representation of angles embedded within contour stimuli in area V2 of macaque monkeys', *Jour. Neurosci.*, 2004, vol. 24, pp. 3313-3324.
- [45] Anzai, A., Peng, X., Van Essen, D. C.: 'Neurons in monkey visual area V2

encode combinations of orientations', *Nat. Neurosci.*, 2007, vol. 10, pp. 1313-1321.

[46] Hegdé, J., Van Essen, D. C.: 'Selectivity for complex shapes in primate visual area V2', *Jour. Neurosci.*, 2000, vol. 20, pp. 61-66.

[47] Martin, D. R., Fowlkes, C. C., Malik, J., 'Learning to detect natural image boundaries using local brightness, color and texture cues', *IEEE Tran. Patt. Anal. Mach. Intel.*, May. 2004, vol. 26, no. 5, pp. 530-549.

[48] Mairal, J., Leordeanu, M., Bach, F., Hebert, M., Ponce, J.: 'Discriminative sparse image models for class-specific edge detection and image interpretation', *ECCV. Comput. Vis.*, 2008, pp. 43-56.

[49] Ren, X., Bo, L.: 'Discriminative trained sparse code gradients for contour detection', *Adv. Neur. Inf. Proc. Sys.*, 2012, pp. 584-592.

[50] Everingham, M., Eslami, S. M. A., Gool, L. V., Williams, C. K. I., Winn, J., Zisserman, A.: 'The pascal visual object classes challenge: a retrospective', *Intern. Jour. Comput. Vis.*, vol. 111, pp. 98-136.

[51] Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: 'Indoor Segmentation and Support Inference from RGBD Images', *Euro. Confer. Comput. Vis.*, 2012, vol. 7576, pp.746-760

[52] Grigorescu, C., Petkov, N., Westenberg, M. A.: 'Contour detection based on nonclassical receptive field inhibition', *IEEE Tran. Image. Proc.*, 2003, vol. 12, pp. 729-739.

[53] Daugman, J. G.: 'Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters', *JOSA. A.*, 1985, vol. 2, pp.1160-1169.

[54] Jain, A. K.: 'Fundamentals of digital image processing', Prentice-Hall, Englewood Cliffs., 1989.

[55] Boukerroui, D., Noble, J. A., and Brady, M.: 'On the choice of band-pass quadrature filters', *Jour. Math. Image. Vis.*, 2004, vol. 21, pp. 53-80.

[56] Papari, G., Petkov, N.: 'Edge and line oriented contour detection: State of the art', *Image. Vis. Comput.*, 2011, vol. 29, pp. 79-103.

[57] Elad, M.: 'On the origin of the bilateral filter and ways to improve it', *IEEE Tran. Image. Proc.*, 2002, vol. 11, pp. 1141-1151.

[58] Zhang, B., Allebach, J. P.: 'Adaptive bilateral filter for sharpness enhancement and noise removal', *IEEE Tran. Image. Proc.*, 2008, vol. 17, pp. 664-678.

[59] Rivest, J. F., Soille, P., Beucher, S.: 'Morphological gradients', *Jour. Electro. Image.*, 1993, vol.2, pp. 326-336.

[60] Evans, A. N., Liu, X. U.: 'A morphological gradient approach to colour edge detection', *IEEE Tran. Image. Proc.*, 2006, vol. 15, pp. 1454-1463.

- [61] Astola, J., Haavisto, P., Neuvo, Y.: 'Vector median filters', Proc. IEEE., 1990, vol. 78, pp. 678-689.
- [62] Jain, K., Farrokhnia, F.: 'Unsupervised texture segmentation using Gabor filters', Proc. IEEE Int. Conf. Sys. Man. Cyber., 1990.
- [63] Will, S., Hermes, L., Buhmann, J. M., and Puzicha, J., 'On learning texture edge detectors', Proc. Int. Conf. Image. Proc., 2000
- [64] Scarpa, G., Gaetano, R., Haindl, M., and Zerubia, J.: 'Hierarchical multiple Markov chain model for unsupervised texture segmentations', IEEE Tran. Image. Proc., May. 2009, vol. 18, no. 8, pp. 1830-1843.
- [65] Ren, X.: 'Multi-scale improves boundary detection in natural images', Proc. Euro. Conf. Comput. Vis., 2008.
- [66] Maire, M., Arbelaez, P., Fowlkes, C., and Malik, J.: 'Using contours to detect and localize junctions in natural image', Comput. Vis. Pat. Recog., June 2008, pp. 1-8.
- [67] Najman, L., Schmitt, M.: 'Geodesic saliency of watershed contours and hierarchical segmentation', IEEE Tran. Pat. Anal. Mach. Intel., 1996, vol. 18, pp. 1163-1173.
- [68] Arbelaez, P.: 'Boundary extraction in natural images using ultrametric contour maps', IEEE Conf. Comput. Vis. Pat. Recog. Workshop, 2006, pp. 182-182.
- [69] Shi, J., Malik, J.: 'Normalized cuts and image segmentation', IEEE Tran. Pat. Anal. Mach. Intel., 2000, vol. 22, pp. 888-905.
- [70] Cohen, L. D., Cohen, I.: 'Finite-element methods for active contour models and balloons for 2-D and 3-D images', IEEE Tran. Pat. Anal. Mach. Intel., 1993, vol. 15, pp. 1131-1147.
- [71] Gunn, S. R., Nixon, M. S.: 'A robust snake implementation; a dual active contour', IEEE Tran. Pat. Anal. Mach. Intel., 1997, vol. 19, pp. 63-68.
- [72] Yan, H.: 'Fuzzy curve-tracing algorithm', IEEE Tran. Sys. Man. Cyber. Part B: Cyber., 2001, vol. 31, pp. 768-780.
- [73] Leventon, M. E., Grimson, W. E. L., and Faugeras, O.: 'Statistical shape influence in geodesic active contours', Proc. IEEE. Conf. Comput. Vis. Pat. Recog., 2000, vol. 1, pp. 316-323.
- [74] Tsai, A., Yezzi Jr, A., Wells, W., Tempany, C., Tucker, D., et al.: 'A shape-based approach to the segmentation of medical imagery using level sets', IEEE Tran. Med. Image., 2003, vol. 22, pp. 137-154.
- [75] Dumitraş, A., Venetsanopoulos, A. N.: 'Angular map-driven snakes with application to object shape description in color images', IEEE Tran. Image. Proc., 2001, vol. 10, pp. 1851-1859.

- [76] Song, Y. Z., Arbelaez, P., Hall, P., Li, C., Balikai, A.: 'Finding semantic structures in image hierarchies using laplacian graph energy', *Comput. Vis. ECCV* 2010. Springer Berlin Heidelberg, 2010, pp. 694-707.
- [77] Gupta, S., Arbeláez, P., Girshick, R., Malik, J.: 'Indoor scene understanding with RGB-D images: Bottom-up segmentation, object detection and semantic segmentation', *Intern. Jour. Comput. Vis.*, 2015, vol. 112, pp. 133-149.
- [78] Oppenheim, A. V., Lim, J. S.: 'The importance of phase in signals', *Proc. IEEE.*, 1981, vol. 69, pp. 529-541.
- [79] Morrone, M. C., Burr, D. C.: 'Feature detection in human vision: A phase-dependent energy model', *Proc. Roy. Soc. London B: Bio. Sci.*, 1988, vol.235, pp. 221-245.
- [80] Wang, S., Kubota, T., Siskind, J. M., Wang, J.: 'Salient closed boundary extraction with ratio contour', *IEEE Tran. Pat. Anal. Mach. Intel.*, 2005, vol. 27, pp. 546-561.
- [81] Papari, G., Petkov, N.: 'Adaptive pseudo dilation for gestalt edge grouping and contour detection', *IEEE Tran. Image Process.*, 2008, vol. 17, pp. 1950-1962.
- [82] Jacobs, D.W.: 'Robust and efficient detection of salient convex groups', *IEEE Trans. Pat. Anal. Mach. Intell.*, 1996, vol. 18, pp. 23-37.
- [83] Elder, J. H, Zucker, S. W.: 'Computing contour closure', *Euro. Confer. Comput. Vision*, Springer-Verlag, 1996, pp. 399-412.
- [84] Thornber, K. K, Williams, L. R.: 'Analytic solution of stochastic completion fields', *Intern. Sympos. Comput. Vision*, *Proc. IEEE*, 1995, pp. 617-622.
- [85] Mahamud, S., Williams, L. R., Thornber, K. K., Xu, K.: 'Segmentation of multiple salient closed contours from real images', *IEEE Trans. Pat. Anal. Mach. Intel.*, vol. 25, pp. 433-444.
- [86] Bai, X., Latecki, L. J., Liu, W. Y.: 'Skeleton pruning by contour partitioning with discrete curve evolution', *IEEE Tran. Pat. Anal. Mach. Intel.*, 2007, vol. 29, pp. 449-62.
- [87] Tuytelaars, T., Turina, A., VanGool, L.: 'Noncombinatorial detection of regular repetitions under perspective skew', *IEEE Tran. Pat. Anal. Mach. Intel.*, 2003, vol. 25, pp. 418-432.
- [88] Siddiqi, K., Bouix, S., Tannenbaum, A., Zucker, S. W.: 'Hamilton-jacobi skeletons', *Intern. Jour. Comput. Vis.*, 2002, vol. 48, pp. 215-231.
- [89] Cornelius, H., Loy, G.: 'Detecting bilateral symmetry in perspective', *IEEE Conf. Comput. Vis. Pat. Recog.*, 2006, pp. 191-191.
- [90] Derrington, A. M., Lennie, P.: 'Spatial and temporal contrast sensitivities of neurones in lateral geniculate nucleus of macaque', *Jour. Physio.*, 1984, vol. 357, pp. 219-240.

- [91] Price, D. J., Morgan, J. E.: 'Spatial properties of neurones in the lateral geniculate nucleus of the pigmented ferret', *Exp. Brain. Res.*, 1987, vol. 68, pp. 28-36.
- [92] Dacey, D., Packer, O. S., Diller, L., Brainard, D., Peterson, B., Lee, B.: 'Center surround receptive field structure of cone bipolar cells in primate retina', *Vis. Res.*, 2000, vol. 40, pp. 1801.
- [93] Young, R. A., Lesperance, R. M.: 'The Gaussian derivative model for spatial-temporal vision: II. Cortical data', *Spatial. Vis.*, vol. 14, pp. 321-389.
- [94] Ringach, D. L.: 'Spatial structure and symmetry of simple-cell receptive fields in macaque primary visual cortex', *Jour. Neurophys.*, 2002, vol. 88, pp. 455-463.
- [95] Jones, J. P., Palmer, L. A.: 'An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex', *Jour. Neurophys.*, 1987, vol. 58, pp. 1233-1258.
- [96] Zhao, R., Wu, M., Liu, X., Zou, B., Li, F.: 'Orientation histogram-based center-surround interaction: an Integration approach for contour detection', *Neuro. Comput.*, 2016, vol.4, pp. 1-23.
- [97] Xiao, J., Cai, C.: 'Contour detection based on horizontal interactions in primary visual cortex', *Electro. Let.*, 2014, vol. 50, pp. 359-361.
- [98] Ben-Shahar, O., Zucker, S.: 'Geometrical computations explain projection patterns of long-range horizontal connections in visual cortex', *Neur. Comput.*, 2004, vol. 16, pp. 445-476.
- [99] Papari, G., Petkov, N.: 'An improved model for surround suppression by steerable filters and multilevel inhibition with application to contour detection', *Pat. Recog.*, 2011, vol. 44, pp. 1999-2007.
- [100] Zeng, C., Li, Y., Yang, K., Li, C.: 'Contour detection based on a non-classical receptive field model with butterfly-shaped inhibition subregions', *Neurocomput.*, 2011, vol. 74, pp. 1527-1534.
- [101] Yarbus, A. L.: 'Eye movements during perception of complex objects', *Eye Movements and Vision*, 1967, pp. 171-211.
- [102] Rafael, G. V. G., Jakubowicz, J., Morel, J. M., Randall, G.: 'Lsd: a fast line segment detector with a false detection control', *IEEE Tran. Pat. Anal. Mach. Intel.*, 2010, vol. 32, pp. 722-732.
- [103] Maire, M., Yu, S. X., Perona, P.: 'Reconstructive sparse code transfer for contour detection and semantic labeling', *Lect. Not. Computer Sci., ACCV 2014*, 2014, vol. 9006, pp. 273-287.
- [104] Yang, J., Price, B., Cohen, S., Lee, H., Yang, M. H.: 'Object contour detection with a fully convolutional encoder-decoder network', *IEEE Confer. Comput. Vis. Patt. Recog.*, 2016, pp. 193-202.

- [105] Xie, S., Tu, Z.: ‘Holistically-nested edge detection’, IEEE. Intern. Confer. Comput. Vis., 2015, pp. 1395-1403.
- [106] Bertasius, G., Shi, J., Torresani, L.: ‘Deepedge: a multi-scale bifurcated deep network for top-down contour detection’, IEEE. Conf. Comput. Vis. Patt. Recog., 2015, vol. 52, pp. 4380-4389.
- [107] Ganin, Y., Lempitsky, V.: ‘N⁴-fields: neural network nearest neighbor fields for image transforms’, Lect. Not. Computer Sci. ACCV 2014, 2014, vol. 9004, pp. 536-551.
- [108] Shen, W., Wang, X., Wang, Y., Bai, X.: ‘DeepContour: A deep convolutional feature learned by positive-sharing loss for contour detection’, IEEE. Confer. Comput. Vis. Patt. Recog., 2015, pp. 3982-3991.
- [109] Chen, L. C., Barron, J. T., Papandreou, G., Murphy, K., Yuille, A. L.: ‘Semantic image segmentation with task-specific edge detection using cnns and a discriminatively trained domain transform’, Comput. Sci., 2016, vol. 9, pp. e84558-e84558.
- [110] Kivinen, J., Williams, C., Heess, N.: ‘visual boundary prediction: a deep neural prediction network and quality dissection’, Proceed. 17th Intern. Confer. Art. Intell. Stast., 2014, vol. 33., pp. 512-521.
- [111] Martin, D., Fowlkes, C., Malik, J.: ‘Learning to find brightness and texture boundaries in natural images’, Neur. Infor. Proc. Sys., 2002.
- [112] Rubinstein, R., Zibulevsky, M., Elad, M.: ‘Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit’, CS. Technion., 2008, vol. 40, pp. 1-15.
- [113] Perez, F., Koch, C.: ‘Toward color image segmentation in analog VLSI: algorithm and hardware’, Intl. Jour. Comput. Vis, 1994, vol. 12, pp. 17-42.
- [114] Lennie, P., Movshon, J. A.: ‘Coding of color and form in the geniculostriate visual pathway (invited review)’, JOSA. A., 2005, vol. 22, issue. 10, pp. 2013-2033.
- [115] Schluppeck, D., Engel, S. A.: ‘Color opponent neurons in v1: a review and model reconciling results from imaging and single-unit recording’, Jour. Vis., 2002, vol.2, pp. 480-492.
- [116] Lafer-Sousa, R., Liu, Y. O., Lafer-Sousa, L., Wiest, M. C., Conway, B. R.: ‘Color tuning in alert macaque v1 assessed with fmri and single-unit recording shows a bias toward daylight colors’, Jour. Opt. Soc. Amer. A Optics. Image Sci. Vis., 2012, vol.29, pp. 657-670.
- [117] Ts’O, D. Y., Gilbert, C. D.: ‘The organization of chromatic and spatial interactions in the primate striate cortex’, Jour. Neurosci. Offi. Jour. Soc. Neurosci., 1988, vol. 8, pp. 1712-1727.
- [118] Yang, K. F., Gao, S., Guo, C., Li, C., Li, Y.: ‘Boundary Detection Using Double-Opponency and Spatial Sparseness Constraint’, IEEE Tran. Image. Proc., Apr. 2015, vol. 24, pp. 2565-2578.

- [119] Amulya, S., Mohan, C. L., Gandhi, S. G.: 'Color-opponent mechanism for boundary detection', Intern. Jour. Innov. Engineer. Tech., 2016.
- [120] Akbarinia, A., Párraga, C. A.: 'Biologically-inspired edge detection through surround modulation', British Mach. Vis. Confer., York, 2016.
- [121] Ratnasingham, S., Robles-Kelly, A.: 'A biologically motivated double-opponency approach to illumination invariance', Comput. Vis. ACCV 2012. Springer Berlin Heidelberg, 2013, pp. 296-309.
- [122] Gao, S., Yang, K., Li, C., Li, Y.: 'A color constancy model with double-opponency mechanisms', IEEE Intern. Confer. Comput. Vis., 2013, pp. 929-936.
- [123] Lee, T. W., Wachtler, T., Sejnowski, T.: 'Colour opponency constitutes a sparse representation for the chromatic structure of natural scenes', Nips, 2001, vol. 13, pp. 866-872.
- [124] Lennie, P., Krauskopf, J., Sclar, G.: 'Chromatic mechanisms in striate cortex of macaque', Jour. Neurosci., Feb. 1990, vol. 10, pp. 649-669.
- [125] Shapley, R., Hawken, M. J.: 'Colour in the cortex: single-and double-opponent cells,' Vis. Res., Apr 2011, vol.51, pp. 701-717.
- [126] Beaudot, W. H., Mullen, K. T.: 'Orientation selectivity in luminance and color vision assessed using 2-d band-pass filtered spatial noise,' Vis. Res., Mar 2005, vol. 45, pp. 687-696.
- [127] Gegenfurtner, K. R.: 'Cortical mechanisms of color vision', Nat. Rev. Neurosci., Jul. 2003, vol. 4, pp. 563-572.
- [128] Leventhal, A. G., Thompson, K. G, Liu D., Zhou, Y., Ault, S. J.: 'Concomitant sensitivity to orientation, direction, and color of cells in layers 2, 3, and 4 of monkey striate cortex', Jour. Neurosci., Mar. 1995, vol. 15, pp. 1808-1818.
- [129] Friedman, H. S., Zhou, H., Heydt, R.: 'The coding of uniform color figures in monkey visual cortex', Jour. Physio., Apr. 2003, vol. 548, pp. 593-613.
- [130] Caywood, M. S., Willmore, B., Tolhurst, D. J.: 'Independent components of color natural scenes resemble V1 neurons in their spatial and colour tuning', Jour. Neurophysio., Jun. 2004, vol. 91, pp. 2859-2873.
- [131] Girard, P., Morrone, M. C.: 'Spatial structure of chromatically opponent receptive fields in the human visual system', Vis. Neurosci., Jan. 1995, vol. 12, pp. 103-116.
- [132] Johnson, E. N., Hawken, M. J., Shapley, R.: 'Cone inputs in macaque primary visual cortex', Jour. Neurophysio., Jun 2004, vol. 91, pp. 2501-2514.
- [133] Mullen, K. T., Dumoulin, S. O., McMahon, K. L., De Zubicaray, G. I., Hess, R. F.: 'Selectivity of human retinotopic visual cortex to S - cone - opponent, L/M - cone - opponent and achromatic stimulation,' Euro. Jour. Neurosci., Jan 2007, vol. 25, pp. 491-502.

- [134] Johnson, E. N., Van Hooser, S. D., Fitzpatrick, D.: 'The representation of S-cone signals in primary visual cortex', *Jour. Neurosci*, Aug. 2010, vol. 30, pp. 10337-10350.
- [135] Lindeberg T: 'A computational theory of visual receptive fields', *Bio. Cyber.*, 2013, vol. 107, pp. 589-635.
- [136] Mardia, K. V.: 'Statistics of directional data', London: Academic, 1972.
- [137] Hubel, D. H., Wiesel, T. N.: 'Receptive fields, binocular interaction and functional architecture in the cat's visual cortex', 1962, *Jour. Physio.*, vol. 160, pp. 106.
- [138] Dollar, P., Tu, Z., Belongie, S.: 'Supervised learning of edges and object boundaries', *IEEE Comput. Soc. Conf. Comput. Vis. Pat. Recog.*, 2006, vol. 2: pp. 1964-1971.
- [139] Harris, C., Stephens, M.: 'A combined corner and edge detector', *Proc. 4th Alvey. Vis. Confer.*, 1988, pp. 147-151.
- [140] Landy, M. S., Graham, N.: '73 Visual perception of texture', *Vis. Neurosci.*, 2004, pp. 1106–1118.
- [141] Elad, M., Aharon, M.: 'Image denoising via sparse and redundant representations over learned dictionaries', *IEEE Tran. Image. Proc.*, 2006, vol. 15, pp. 3736-3745.
- [142] Mairal, J., Elad, M., Sapiro, G.: 'Sparse representation for colour image restoration', *IEEE Tran. Image. Proc.*, 2008, vol. 17, pp. 53-69.
- [143] Skretting, K., Husøy, J. H.: 'Texture classification using sparse frame-based representations', *EURASIP. Jour. Applied. Sig. Proc.*, Jan. 2006, pp. 102-102.
- [144] Wright, J., Yang, A. Y., Ganesh, A., Sastry, S. S., Ma, Y.: 'Robust face recognition via sparse representation', *IEEE Tran. Pat. Anal. Mach. Intel.*, 2009, vol. 31, pp. 210-227.
- [145] Mullen, K. T.: 'The contrast sensitivity of human colour vision to red - green and blue - yellow chromatic gratings', *Jour. Physio.*, 1985, vol. 359, pp. 381-400.