Perceptual Texture Similarity Estimation

Xinghui Dong M. Eng. B. Eng.

Submitted for the degree of Doctor of Philosophy

Heriot-Watt University

School of Mathematical and Computer Sciences

September 2014

Abstract

This thesis evaluates the ability of computational features to estimate perceptual texture similarity.

In the first part of this thesis, we conducted two evaluation experiments on the ability of 51 computational feature sets to estimate perceptual texture similarity using two different evaluation methods, namely, pair-of-pairs based and retrieval based evaluations. These experiments compared the computational features to two sets of human derived ground-truth data, both of which are higher resolution than those commonly used. The first was obtained by free-grouping and the second by pair-of-pairs experiments. Using these higher resolution data, we found that the feature sets do not perform well when compared to human judgements.

Our analysis shows that these computational feature sets either (1) only exploit power spectrum information or (2) only compute higher order statistics (HoS) on, at most, small local neighbourhoods. In other words, they cannot capture aperiodic, long-range spatial relationships. As we hypothesise that these long-range interactions are important for the human perception of texture similarity we carried out two more pair-of-pairs experiments, the results of which indicate that long-range interactions do provide humans with important cues for the perception of texture similarity.

In the second part of this thesis we develop new texture features that can encode such data. We first examine the importance of three different types of visual information for human perception of texture. Our results show that contours are the most critical type of information for human discrimination of textures. Finally, we report the development of a new set of contour-based features which performed well on the free-grouping data and outperformed the 51 feature sets and another contour type feature set with the pair-of-pairs data.

To my parents!

Acknowledgements

First and foremost I would like to thank my supervisor, Prof. Mike J. Chantler, for his guidance, patience, encouragement, support, discussions and everything that I have learned from him. I would also like to thank Prof. Junyu Dong for leading me into the field of texture analysis. My thanks are also due to Dr. Fraser Halley and Dr. Alasdair D. F. Clarke for their earlier work and to Dr. Stefano Padilla, Dr. Thomas S. Methvenand, Mr. David Robb and Mr. Craig Mackie for their proofreading and help in the preparation of this thesis.

Many thanks to Stefano, Al, Fraser, Lin, Tom, Pawel, Dave, Craig and all people who have ever worked in the Texture Lab for creating a great and fun research environment that make my life in Edinburgh richer. I am also grateful to my Chinese fellow students for their friendships and helps.

I would like to thank the Heriot-Watt University for offering me the “Life Science Interface” scholarship to pursue this thesis.

Finally, I would like to express my gratitude to my parents, my sister, my brother and the rest of the family for their consistent support, encouragement and everything that they have been doing for me.

Table of Contents

List of Figures

List of Tables

List of Terms

1st-Order Similarity (Judgement) A similarity (judgement) between a pair of objects.

1st-Order Statistic A statistic computed from one pixel individually.

2D Texture Classification Texture classification using an image for one texture sample.

2nd-Order Similarity (Judgement) A similarity (judgement) between two object pairs.

2nd-Order Statistic A statistic computed from two pixels simultaneously.

3D Texture Classification Texture classification using multiple images acquired under different illuminations, view angles, or rotation angles, for each texture sample.

8D-ISO 8-dimensional Isomap of the original perceptual similarity matrix. It is also referred to as "8D-ISO similarity matrix".

Agreement Rate (%) A measure of the consistency of two different pair-of-pairs judgement sets.

Boolean-Valued Similarity A similarity with only two possible values: "1" (similar) or "0" (dissimilar) obtained from one pair of objects. It is a 1st-order similarity.

Computational Similarity (Judgement) The similarity (judgement) estimated using a certain computational feature set.

Estimated Perceptual Similarity (Judgement) In this thesis, this is the same as "computational similarity (judgement)".

Feature (Image Feature) An individual measurable property of an image.

Features A general term for various features or computational feature extraction algorighms.

Feature Set A set of features extracted from an image or a series of images using one computational algorithm or several algorithms. In this thesis, it is also used to represent the computational algorithm(s).

Higher Order Statistic (HoS) A statistic computed from at least three pixels simultaneously.

Higher Resolution Similarity The similarity data which is more precise than the Boolean-valued similarity.

Image-Based Similarity The similarity between two whole-sized images.

Inter-Class/Cluster Similarity The similarity between different classes/clusters of objects.

Intra-Class/Cluster Similarity The similarity within the same class/cluster of objects.

Isomap (Similarity Matrix) The result of applying Isomap analysis on the original perceptual similarity matrix.

Long-Range Interactions In this thesis, it is defined as the higher order spatial relationship of the pixels in a spatial extent (image region) that is greater than 25×25 pixels.

Multi-resolution The combination of five individual resolutions: 1024×1024, 512×512, 256×256, 128×128 and 64×64.

Original Perceptual Similarity Matrix In this thesis, it denotes the perceptual similarity matrix obtained using free-grouping without any post-processing.

Original Texture Image In this thesis, it is a texture image in the *Pertex* database.

Pair-of-Pairs Similarity (Judgement) A similarity judgement between one pair of objects and one other pair of objects.

Patch-Based Similarity The similarity between two image patches cropped from their original images.

Perceptual Similarity The similarity data obtained using the judgements of humans.

Pixel-Based Similarity The similarity between two pixels in the same image or different images.

Ranking-Valued Similarity The order of the ranked list of a set of objects according to the similarity between these and a (query) object.

Rational-Valued Similarity A similarity value which can be computed from the fraction p/q of two integers, where the denominator q should not be equal to zero. In a free-grouping experiment, p is the number of human observers who put two textures into the same group and q is the total number of human observers.

Real-Valued Similarity The similarity data which is in the real number range.

Short-Range Interactions In this thesis, it is defined as the higher order spatial relationship of the pixels in a spatial extent (image region) that is no more than 25×25 pixels.

Similarity (Judgement) A qualitative or quantitative similarity measure or description.

Similarity Value A quantitative similarity metric.

Six Resolutions Five individual resolutions (1024×1024, 512×512, 256×256, 128×128 and 64×64) and the multi-resolution scheme.

Texture (Image) An image taken of a certain texture sample under some conditions.

Texture (Image) Patch An image patch cropped from one texture image.

(Texture) Property Image The image obtained from an original texture image using certain image processing techniques.

Texture Sample A piece of physical material, e.g. cloth, wallpapers, etc, which contains a certain texture pattern.

Table of Acronyms

Acronym	Terminology
2AFC	Two-alternative forced choice
ANOVA	Analysis of variance
DFT	Discrete Fourier transform
FFT	Fast Fourier transform
F-R-F	Filter-rectify-filter
FT	Fourier transform
HoS	Higher order statistic(s)
L-N-L	Linear-nonlinear-linear
MDS	Multidimensional scaling
SDoCS	Spatial distributions of contour segments
SVM	Support vector machine(s)

Table of Symbols

Performance Measures	
ρ or *rho*	Spearman's rank correlation coefficient
τ or *tau*	Kendall's rank correlation coefficient
NP	Normalised precision
NR	Normalised recall
SF	Spearman's footrule
G	G measure
M	M measure (the improved version of G measure)

Computational Features	
$f(x, y)$	An M×N image or only one pixel at the position of (x, y)
$f(x + \Delta x, y + \Delta y)$	The displaced matrix or pixel of $f(x, y)$
$f'(x, y)$	The filtered image or only one filtered pixel of $f(x, y)$
$h(x, y)$	A spatial filter (function)
$H(u, v)$	The transform function of $h(x, y)$
$F(u, v)$	The transform function of $f(x, y)$
(u, v)	A coordinate in the frequency domain

Resolutions	
1024 (×1024)	The resolution of 1024×1024
512 (×512)	The resolution of 512×512
256 (×256)	The resolution of 256×256
128 (×128)	The resolution of 128×128
64 (×64)	The resolution of 64×64
Multi (-resolution)	The multi-resolution scheme

Similarity Matrix	
$S_p(i,j)$	Perceptual similarity value between texture i and j
$DS_p(i,j)$	Perceptual dissimilarity value between texture i and j

Pair-of-Pairs Experiments	
POP	Pair-of-pairs experiment
POP_O	Original pair-of-pairs experiment using original texture images
POP_N	Non-randomized blocked pair-of-pairs experiment using non-randomized blocked images
POP_R	Randomized blocked pair-of-pairs experiment using randomized blocked images

Pair-of-Pairs Judgements for Evaluation Experiments	
$POPJ$	A pair-of-pairs judgement set
$POPJ(i)$	The pair-of-pairs judgement made by the majority of participants in trial i ($i = 1, 2, \dots, 1000$)
$POPJ_{POP}$	The pair-of-pairs judgement set made by the majority of participants in all 1000 trials in POP_O
$POPJ_P$	The pair-of-pairs judgement set constructed from the original perceptual similarity matrix corresponding to the 1000 trials in POP_O
$POPJ_{ISO}$	The pair-of-pairs judgement set constructed from the 8D-ISO similarity matrix corresponding to the 1000 trials in POP_O
$POPJ_E$	The pair-of-pairs judgement set estimated using a computational similarity matrix corresponding to the 1000 trials in POP_O

Pair-of-Pairs Judgements for *t*-test	
$SPOPJ_N(m,i)$	The single pair-of-pairs judgement made by participant m ($m = 1, 2, \dots, 10$) in trial i ($i = 1, 2, \dots, 80$) in POP_N
$SPOPJ_R(m,i)$	The single pair-of-pairs judgement made by participant m ($m = 1, 2, \dots, 10$) in trial i ($i = 1, 2, \dots, 80$) in POP_R
$AR_N(m)$	The agreement rate between the pair-of-pairs judgement set $SPOPJ_N$ made by observer m in POP_N and the baseline $POPJ_{POP}$ (only the 80 most inconsistent pairs of pairs are considered)
$AR_R(m)$	The agreement rate between the pair-of-pairs judgement set $SPOPJ_R$ made by observer m in POP_R and the baseline $POPJ_{POP}$ (only the 80 most inconsistent pairs of pairs are considered)

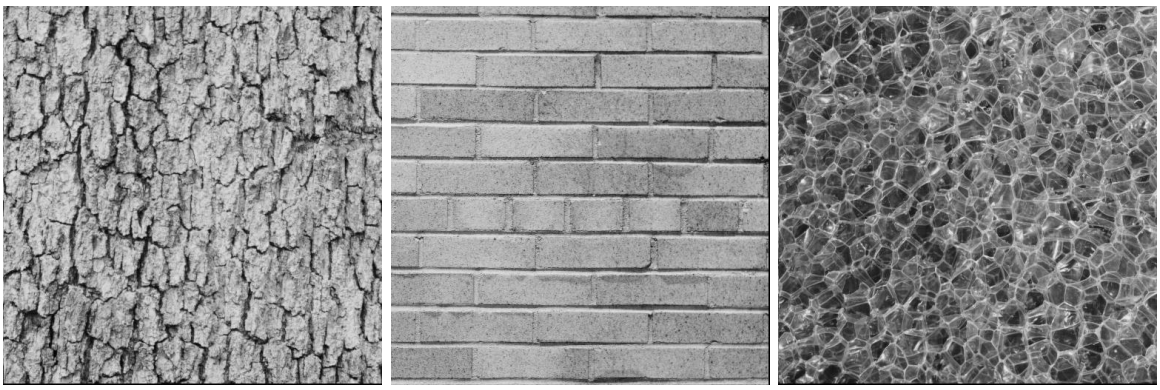
Chapter 1

Introduction

Although performances in the high nineties are typically obtained for tasks such as texture segmentation and classification, the same cannot be said of estimating texture similarity. Such perceptual similarity data are useful for a variety of tasks, from measuring the perceived difference between the appearance of textures (e.g. the visual difference between a worn carpet and a new sample) to simply perceptually ranking search results. One possible reason is that it is time-consuming to collect perceptual similarity data over a large texture database using human observers. This research thereby aims at bring together vision science knowledge and computer vision techniques to estimate perceptual texture similarity using computational features.

1.1 Background

Texture can be found at every corner of the real world: nearly any visible object has surface texture at some scale. A large number of textures have been observed on both natural and artificial objects (see Figure 1.1), such as soils, brick walls, leaves, and so on. Textures are normally divided into two categories: tactile and visual. The former represents the immediate tangible feel of a surface while the latter stands for the visual impression that textures bring to human observers, which are associated with pattern, colour, orientation and intensity within an image. However, the use of the term of “texture” is slightly confusing in the field of computer vision because it deviates from its original meaning. Although texture can be perceived by the human vision system, there is no general definition for it in the literature.



(a)

(b)

Figure 1.1: Examples of textures: (a) three textures in the Brodatz texture dababase [Brodatz, 1966] and (b) three textures in the CUReT texture database [Dana et al., 1999].

Texture is generally regarded as an important surface characteristic. Haralick [1979] described texture as "The image texture we consider is nonfigurative and cellular. We think of this kind of texture as an organized area phanomena" and "An image texture is described by the number and types of its primitives and the spatial organization or lay-out of its primitives". In the opinion of Bovik et al. [1990], "a perceptual surface texture may be informally defined to be a spatial pattern of local surface radiances giving rise to a perception of surface homogeneity. Within this context, an image texture may be defined as a local arrangement of image irradiances projected from a surface patch of perceptually homogeneous radiances". In addition, Jain and Karu [1996] presented this different description: "texture is characterised not only by the grey value at a given pixel, but also by the grey value 'pattern' in a neighbourhood surrounding the pixel".

As a popular topic in computer vision community, texture analysis involves texture synthesis, segmentation, classification, retrieval, and other topics. Texture segmentation, classification and retrieval tasks normally utilise the measurement of texture similarity. Texture segmentation [Bovik et al., 1990] [Jain and Farrokhnia, 1991] is one of the

most challenging issues involved in image segmentation. Montages are normally constructed from different texture images according to predefined masks (see Figure 1.2). The task of texture segmentation is thus to segment these spatially inhomogeneous montaged images into regions of different homogeneous textures.

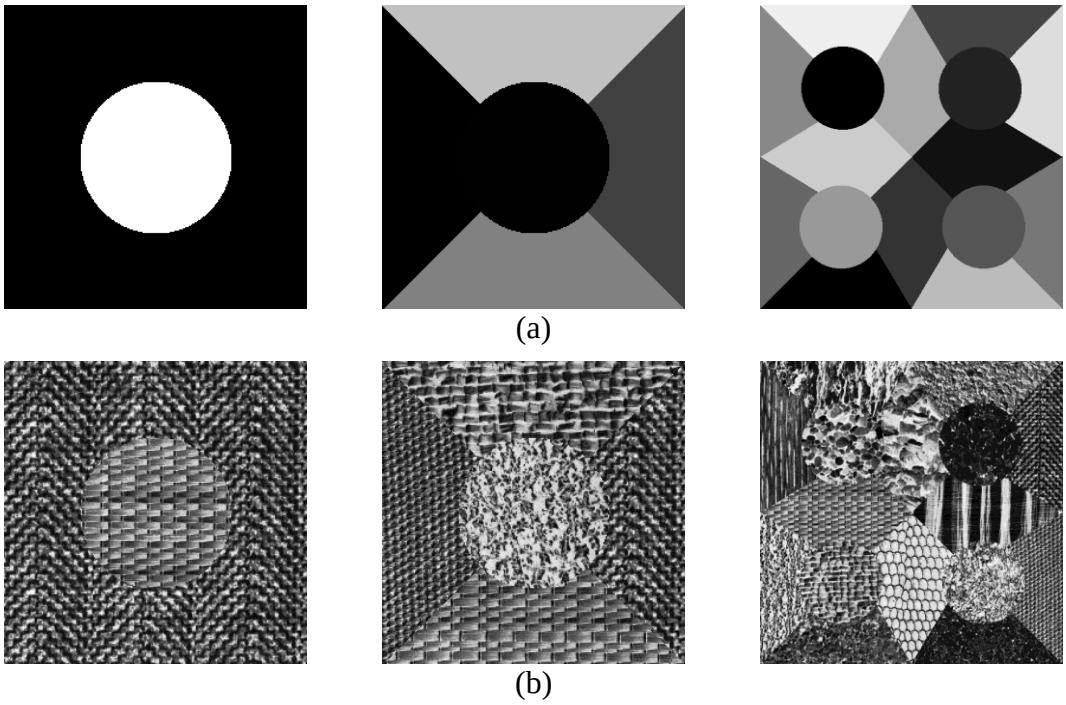


Figure 1.2: Predefined ground-truth masks (a) and corresponding montaged texture images (b) [Randen and Husøy, 1999].

Texture classification originally placed its emphasis on the preattentive discrimination of texture patterns in binary images. Then textures in gray level images with 2D variations became more attractive to texture classification research [Varma and Zisserman, 2009]. Two dimensional texture classification normally divided one texture image into *N* equal-sized patches. All *N* patches of one texture image are considered as an individual texture class. Figure 1.3 presents 32 texture patches obtained from Textures "001" to "032" in the *Pertex* texture database [Halley, 2011B] respectively. All texture patches are divided into training and test subsets and the *N* patches of each texture are partitioned into these subsets. Given an unknown texture patch (in the test dataset), 2D texture classification compares the patch with all known texture patches (in the training dataset) and assigns it the class label of the most similar known patch. In recent years, classifying textures with 3D variations due to changes in camera poses and illuminations has become a focus topic. In this case, texture images acquired from the same sample under different conditions are regarded as a separate texture class. The classification process is similar to the 2D classification.

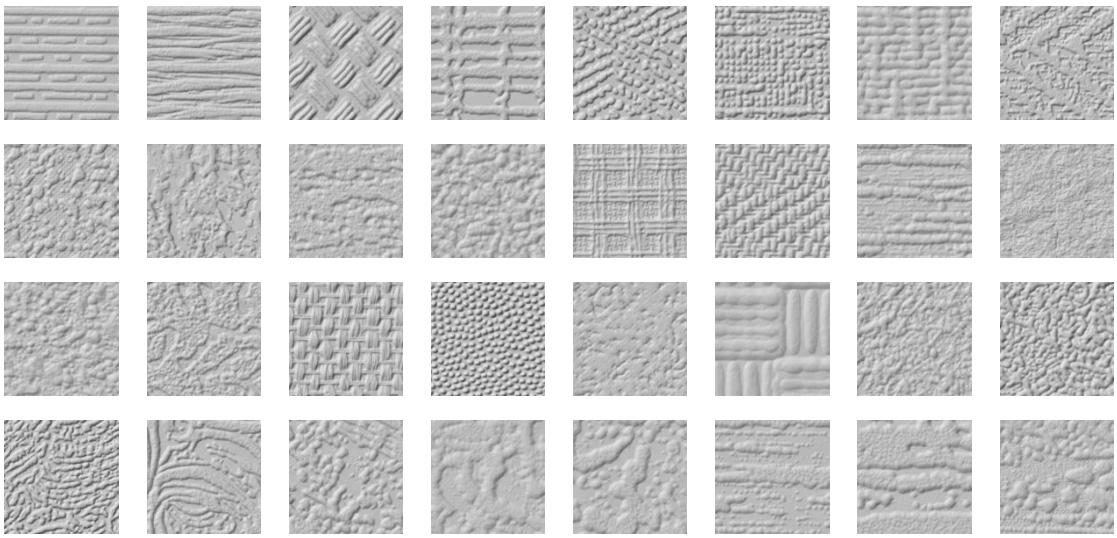
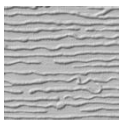
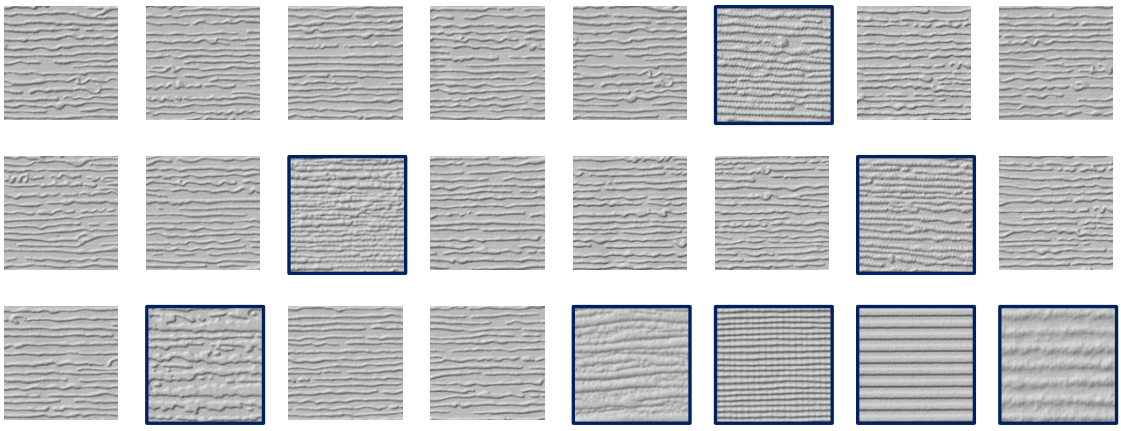


Figure 1.3: Texture patches used for 2D texture classification: Textures "001" to "032" (each texture is divided into 16 patches) in the Pertex texture dababase [Halley, 2011B].

Similar to 2D texture classification, texture retrieval also splits each texture image into *N* equal-sized patches. Only the patches obtained from the same texture are taken as relevant. However, for one retrieval operation, one patch is considered as query image and is compared with all the other image patches. Figure 1.4 (b) shows a retrieved texture patch list of the query texture patch displayed in Figure 1.4 (a).



(a)



(b)

Figure 1.4: Results of a texture retrieval operation (each texture is divided into 16 patches): (a) a query texture patch and (b) its top 24 retrieved texture patches. Here, the texture patches surrounded by a dark blue square box are irrelevant texture patches with the query texture patch.

However, the three tasks introduced above only consider elements (pixels, patches or images) in the same class as similar while regarding elements of different classes as dissimilar. Therefore, they only examine the Boolean-valued (binary) texture similarity. In the literature, very little research has been conducted on higher resolution texture similarity estimation. In this situation, it is an unanswered question as to whether or not computational features are able to estimate higher resolution texture similarity accurately even though some of these features have been shown to perform excellently in texture segmentation, classification and retrieval.

1.2 Motivation and Goals

Texture similarity normally concerns the magnitude of the likeness of two textures. Compared with other topics in the field of texture analysis, texture similarity estimation has received less attention. The few exceptions include the work conducted for measuring texture similarity using several perceptual texture properties [Tamura et al., 1978] [Amadasun and King, 1989] [Fujii et al., 2003] [Abbadeni, 2011]. In addition, perceptual texture dimensionality was investigated by Rao and Lohse [1993, 1996], and Cho et al. [2000]. However, Heaps and Handel [1999] pointed out that texture similarity is context dependent and that a dimensional model is not appropriate. In this context, the research on perceptual texture similarity based on several texture properties is limited.

Furthermore, Payne et al. [1999] benchmarked computational texture rankings against perceptual rankings. Their studies, however, only used 112 *Brodatz* [Brodatz, 1966] textures and a relatively small number of human observers. Santini and Jain [1999] asked human observers to rank only the union of two top *N* retrieved texture sets and thus did not obtain “real” perceptual rankings. Moreover, Zujovic [2011] only examined Boolean-valued similarity and never used higher resolution similarity (see Section 2.1). Generally speaking, higher resolution perceptual similarity can be acquired using psychophysical experiments [Lowe, 1985] [David, 1988] [Rao and Lohse, 1993] [Wenger, 1997] [Clarke et al., 2012]. However, these experiments are time-consuming when a large number of textures are involved. As an alternative, the estimation of texture similarity using computational features is more practical. If perceptual similarity obtained from a texture database can be estimated using computational features, the estimation can be propagated to other databases. In other words, these features can be used to help generate algorithms that mimic human perceptual judgements.

The goal of this research is, therefore, to combine vision science knowledge and computer vision techniques for estimating perceptual texture similarity using a large pool of computational features. First of all, one or more sets of perceptual texture similarity data are required as the ground-truth. In order to evaluate the ability of computational features to estimate perceptual texture similarity, an evaluation framework is also necessary. Existing computational features can be benchmarked against the perceptual similarity data under this framework in order to, determine good candidate measures, or to help understand the failure of less successful features (especially, in terms of human visual mechanisms). A set of perceptually-motivated new features can then be developed in order to exploit these human visual mechanisms.

1.3 Scope

1.3.1 Types of Similarity Data

This thesis is limited to investigating the estimation of image-based higher resolution perceptual texture similarity, rather than texture segmentation, classification or retrieval algorithms which often only concerned with pixel-based or pacth-based Boolean-valued texture similarity. Ideally, real-valued perceptual texture similarity should be utilised due to its high resolution (or precision). However, it is not possible to obtain these types of data when using a psychophysical experiment with a finite number of human observers. As a result, we mainly use the rational-valued perceptual similarity matrix obtained by Halley [2011B]. Specifically, only pair-of-pairs judgements and rankings (both own higher resolution than the Boolean-valued similarity data) generated from perceptual and computational similarity matrices are investigated in this thesis.

1.3.2 Range of Computational Features

In the last forty years, a huge number of computational image features have been proposed in the fields of computer vision and pattern recognition. It is not possible to investigate all of these features within this thesis. Therefore, we examine 51 sets of computational features (see Section 2.2.5), including classical and state-of-the-art measures.

In addition, one shape recognition type contour-based feature set is also compared with the new feature set proposed in Chapter 9.

1.3.3 Training

Machine learning techniques, such as artificial neural networks [Hopfield, 1988] and support vector machines (SVM) [Cortes and Vapnik, 1995], are generally utilised to train a learning model and are assumed to predict image similarity better. In addition, manifold ranking [Zhou et al., 2003] [He et al., 2004], local learning [Wu and Schölkopf, 2007], local regression and global alignment (LRGA) [Yang, 2012] and deep learning [Zhong et al., 2011] are also popular in the fields of image retrieval and classification. However, the performance of these tactics depends greatly on the selection of training samples and model; unsuitable sample or model selection can lead to overfitting [Cawley and Talbot, 2010]. Since our goal is obtaining a set of perceptually-motivated computational features rather than applying such techniques to improve the performance of existing feature sets, we therefore exclude machine learning techniques from this study.

1.4 Contributions

The main contribution of this research is that it brings together vision science knowledge and computer vision techniques for estimating perceptual texture similarity. To the best of our knowledge, perceptual texture similarity estimation using computational texture features has not been rigorously investigated so far. We believe that the contributions of this thesis can be identified as listed below.

- The use of higher resolution texture similarity (higher than the Boolean-valued similarity which is normally used by texture classification, segmentation and retrieval tasks) specifically, pair-of-pairs (texture similarity) judgements and (texture) rankings, to perform the investigation
- The development of a two-stage feature extraction model and review of 51 computational feature sets in terms of feature category, their statistical properties in both stages and the spatial extent that they exploit in the first stage
- The finding that none of the 51 feature sets exploit higher order statistics over a spatial extent of 25×25 pixels

- The introduction and use of two assessment methods and associated metrics, namely, pair-of-pairs based, and retrieval based, methods for evaluating computational features using higher resolution similarity data
- The provision of evidence that suggests that long-range interactions are important to human perception of texture similarity
- The confirmation of the importance of contours for the perception of texture and development of a set of contour-based texture features

1.5 Thesis Organisation

This thesis consists of ten chapters. The organisation of these chapters is displayed in Figure 1.5. In addition, Chapters 2 to 10 are briefly introduced below.

Chapter 2 investigates related publications to this study. We investigate existing sources of perceptual similarity data, computational features and texture databases for this research. In addition, a set of performance measures and four popular image properties for texture representation are investigated. Finally, we discuss related human vision mechanisms for the perception of contours.

Chapter 3 first proposes a two-stage feature model and then briefly reviews the 46 feature sets that we chose in Section 2.2 according to this model. The implementation of these feature sets and five new feature sets is also introduced in this chapter. This chapter is associated with computer vision techniques.

Chapter 4 introduces a pair-of-pairs based evaluation framework for comparing computational and perceptual pair-of-pairs judgements. Although the framework uses human perceptual data as the ground-truth, it is closer to computer vision techiques.

Chapter 5 reports the results of conducting two pair-of-pairs based evaluation experiments, using two different sources of human ground-truth data, in order to examine the ability of the computational feature sets to estimate perceptual pair-of-pairs judgements. The experiments conducted in this chapter are related to computer vision techniques.

Chapter 6 firstly introduces a retrieval-based evaluation method and then carries out a retrieval-based evaluation experiment on the same feature sets as those examined in Chapter 5. The evaluation method and the evaluation experiments conducted in this chapter are related to computer vision techniques.

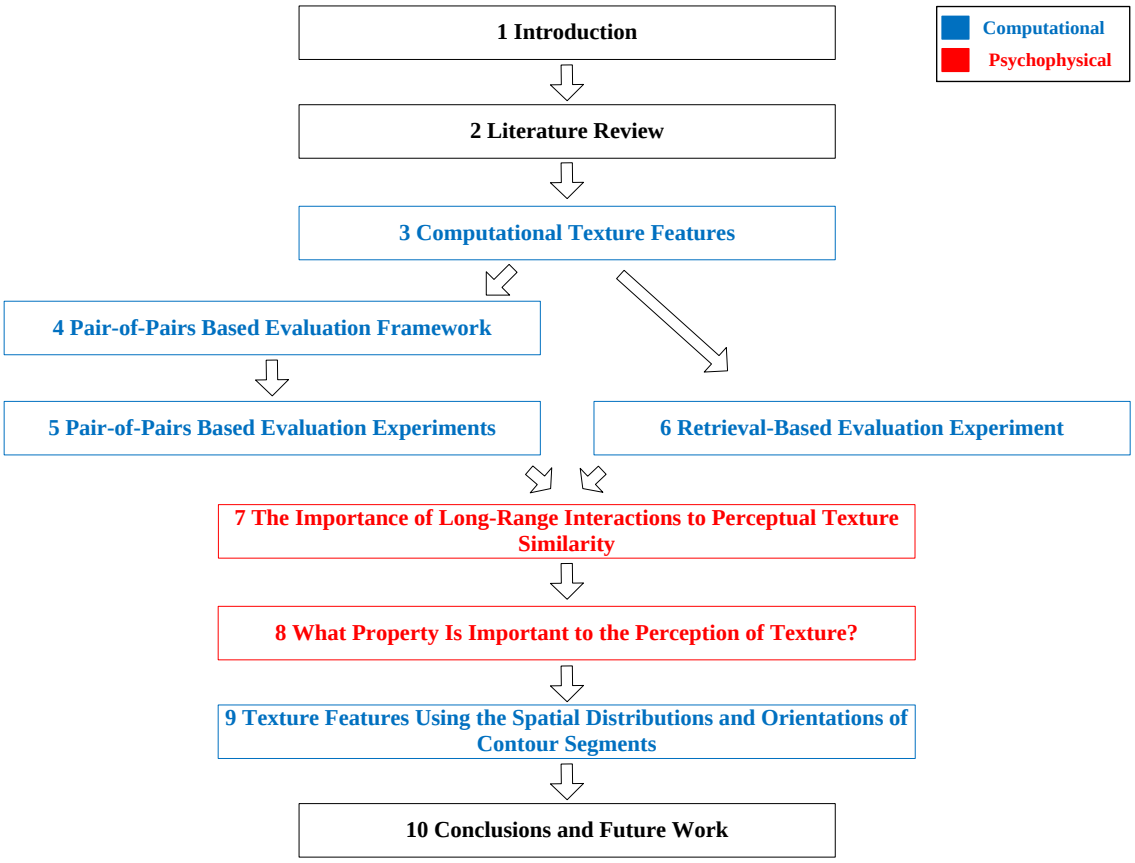


Figure 1.5: The organisation of the ten chapters in this thesis. It can be seen that this research is comprised of computer vision (computational) techniques and vision science (psychophysical) knowledge.

Chapter 7 investigates the importance of long-range interactions to perceptual texture similarity using two modified pair-of-pairs experiments. This chapter is associated with vision science (psychophysics).

Chapter 8 examines three image properties: power spectra, texture exemplars and contours on the human perception of texture, for the purpose of determining which property is most important to texture perception. This chapter is related to vision science (psychophysics).

Chapter 9 first reviews a number of existing contour representation techniques and then proposes a new contour-based texture feature set. The proposed feature set is compared with the 51 feature sets previously analysed and one shape recognition type feature set, using both the pair-of-pairs based and retrieval based evaluation methods. This chapter is associated with computer vision techniques.

Chapter 10 draws the conclusions of this thesis and discusses potential future work.

Chapter 2

Literature Review

This thesis investigates perceptual texture similarity. In this chapter, we review related publications in order to:

(1) survey different types of perceptual texture similarity and decide on which type of similarity should be the focus of our research;

(2) examine existing computational texture features and determine what features will be further investigated in this study;

(3) compare existing texture databases and select one for further study;

(4) investigate commonplace performance measures for similarity comparison and decide whether or not these are suitable for this research;

(5) survey popular image properties for texture representation and identify potential properties for developing a new feature set; and

(6) investigate and identify related human vision mechanisms for the perception of long-range visual interactions in texture.

To be more specific, Section 2.1 investigates related knowledge for collecting two types of perceptual texture similarity. In Section 2.2, publications are reviewed in order to identify potential computational features for estimating perceptual texture similarity. A series of published texture databases are examined and compared in Section 2.3. Section 2.4 examines two types of performance measures used for texture analysis tasks and ranking comparisons. In addition, a set of popular image properties are reviewed in Sec-

tion 2.5 and a number of related publications in vision science are surveyed in Section 2.6. Finally, conclusions are drawn in Section 2.7.

2.1 Perceptual Texture Similarity

Similarity measurement normally generates a positive quantitative value or a qualitative judgement (e.g. similar/dissimilar) concerning the likeness of two objects. It can be divided into perceptual similarity and computational similarity according to different acquisition sources (humans or computational). The former is normally used as the ground-truth data for evaluating the performance of the latter [Payne et al., 1999] [Bar-Ilan et al., 2007] [Metaxas et al., 2009] [Hariri, 2011].

Hence, "texture similarity" concerns the strength of the likeness of two texture images, texture patches, or pixels. In this thesis we restrict our research to image-based texture similarity, i.e. the similarity between whole images of different texture samples.

2.1.1 Boolean-Valued Perceptual Texture Similarity

In general, automated texture segmentation, classification and retrieval do not use perceptual texture similarity directly. A set of "class labels" is normally utilised by these tasks as their ground-truth data and Boolean-valued similarity matrices can be generated from these class labels. Since the class labels are obtained by humans, the Boolean-valued similarity matrix can also be considered as a set of, albeit low resolution, perceptual similarity data.

Texture Segmentation

Image segmentation normally partitions images into a series of non-overlapping regions and can be used for object or edge (contour) detection [Malik et al., 2001]. Given that pixels in one region share common visual characteristics, they are assigned the same label. Similarly, texture segmentation partitions a spatially inhomogeneous texture image into regions of a homogeneous texture [Bovik et al., 1990] [Jain and Farrokhnia, 1991] [Chang et al., 1999]. Traditionally, human manual segmentation maps are utilised as the ground-truth. Alternatively, montages are constructed from different texture images according to a region mask and, in this case, the pixel labels can be derived from

the mask. Thus, the similarity data that can be obtained from this type of ground-truth is binary: a pixel is "similar" to another pixel if it is in the same region, otherwise it is "dissimilar".

Texture Classification

The images used by image classification are normally grouped into "classes" according to their content. The task of image classification is one of classifying an image into certain class according to its visual content [Wu and Schölkopf, 2007] [Zhang et al., 2007]. Similarly, texture classification assigns one of a set of texture class labels to an unknown texture image [Varma and Zisserman, 2009], texture image patch [Ojala et al., 2002B], or pixel (pixel-based texture classification) [Randen and Husøy, 1999]. The ground-truth for each of these three types of classification is the set of class labels of the image, region, or pixel. Hence, the similarity data that can be obtained from such ground-truth is again binary: an image, region or pixel can be considered as similar to another image, region or pixel or not depending on whether or not it has the same label.

Texture Retrieval

Given a query image, content-based image retrieval (CBIR) generally compares the query with images in a database and returns top N most relevant images [He et al., 2004] [Yang, 2012] [Xu et al., 2013]. The images used by CBIR are usually grouped into classes according to their content. Only images in the same class are considered relevant. Similar to some types of texture classification assessment, texture retrieval [Manjunath and Ma, 1996] [Do and Vetterli, 2002] [Khelifi and Jiang, 2011] also splits each texture image into a number of patches. Different patches of one texture image can be regarded as "identical" if the texture is considered to be homogenous. Thus, only patches of the same texture are generally given the same label. Again the similarity data that can be obtained from such ground-truth are binary.

Summary of Boolean-Valued Perceptual Texture Similarity

Texture segmentation, retrieval and classification tasks generally consider elements (images, patches or pixels) in the same class as similar while regarding elements of different classes as dissimilar. In other words, they do not discriminate different intra-class (within the same class) or inter-class (between different classes) similarity and always

take all intra-/inter-class similarity as binary (similar or not). When one class is considered, the similarity between elements in this class and elements of other classes is set dissimilar ("0") while elements within a class are considered "similar" and are assigned a "1". Hence, although these tasks do not use a similarity matrix directly, the ground-truth can be used to generate a Boolean-valued similarity matrix SM (see Figure 2.1).

	Class 1	Class 2	…	…	Class m
Class 1	SSM_{11}	SSM_{12}	…	…	SSM_{1m}
Class 2	SSM_{21}	SSM_{22}	…	…	SSM_{2m}
…	…	…	…	…	…
…	…	…	…	…	…
Class m	SSM_{m1}	SSM_{m2}	…	…	SSM_{mm}

Figure 2.1: A perceptual similarity matrix (SM) obtained from the ground-truth data of texture segmentation, classification or retrieval. In this matrix, m sub-matrices $SSM_{ii}, i=1, 2, ..., m$ are employed to hold the similarity of elements within each class i. Meantime, the rest $m(m-1)$ sub-matrices $SSM_{ij}, i \neq j, i, j=1, 2, ..., m$ hold the similarity between elements of class i and class j.

1	1	…	…	1
1	1	…	…	1
…	…	…	…	…
…	…	…	…	…
1	1	…	…	1

0	0	…	…	0
0	0	…	…	0
…	…	…	…	…
…	…	…	…	…
0	0	…	…	0

Figure 2.2: Two sub-similarity matrices: (left) the (intra-class) similarity (all "1") between elements within class i, $i = 1, 2, ..., m$, and (right) the (inter-class) similarity (all "0") between elements of class i and elements of class $j, j \neq i$.

To be specific, there are m classes of elements in a database (or an image) and $n_i, i=1, 2, ..., m$ elements are included in class i. Generally, $n_i, i=1, 2, ..., m$ could be different for each class. In this case, the SM can be expressed as that shown in Figure 2.1. In this matrix, m sub-matrices $SSM_{ii}, i=1, 2, ..., m$ are employed to hold the similarity of elements within each class i. At the same time, the rest of the sub-matrices $SSM_{ij}, i \neq j, i, j=1, 2, ..., m$ hold the similarity between elements of class i and elements

of class j. For one class i, $i = 1, 2, ..., m$, similarity values in SSM_{ii} are set as "1" (see Figure 2.2 (left)) while similarity values of all $(m-1)$ sub-matrices $SSM_{pq}, p{=}i, q{\neq}i, q{=}1, 2, ..., m$ are assigned with "0" (see Figure 2.2 (right)). All m sub-matrices $SSM_{pq}, p{=}i, q{=}1, 2, ..., m$ (the i-th row in Figure 2.1) are used as the ground-truth data for class i, $i = 1, 2, ..., m$.

Since the labelling of classes is performed by humans, the Boolean-valued similarity matrix SM (see Figure 2.1) can be regarded as a set of perceptual similarity data. This type of Boolean-valued similarity matrices is coarsely quantised compared with the similarity obtained by using psychophysical experimental approaches which will be discussed next.

2.1.2 Higher Resolution Perceptual Texture Similarity

In the literature, the collection of higher resolution perceptual similarity data has been reported using several different methods. The most popular approaches include pair-wise comparison [David, 1988], perceptual ordering [Lowe, 1985] [Wenger, 1997] [Payne et al., 1999], relevance feedback [Rui et al., 1998], pair-of-pairs comparison [Clarke et al., 2012] and free-grouping [Rao and Lohse, 1993] [Halley, 2011A].

Pair-wise Comparison

Pair-wise comparison (or paired comparison) [David, 1988] has been applied in a variety of fields, such as marketing, psychology, etc, as it can build an overall ranking [Agresti, 2002]. Generally speaking, the method of pair-wise comparison presents two objects simultaneously and observers are asked to respond with an answer of yes/no, or use a metric value to describe their common/distinct properties. The results obtained from one observer are normally a set of Boolean-valued (similar or not), ordinal-valued or interval-valued similarity. Given M images, a complete pair-wise comparison involves $M(M-1)/2$ trials. Pair-wise comparison is thus time-consuming and of $O(n^2)$ time complexity.

Perceptual Ordering and Relevance Feedback

In a perceptual ordering experiment [Lowe, 1985] [Amadasun and King, 1989] [Wenger, 1997] [Payne et al., 1999], observers are required to rank images according to

their similarity compared with one given image, or rank all images according to one or more qualities. In comparison, relevance feedback [Rui et al., 1998] is an iterative process in which users provide feedback on the retrieval result repeatedly until an acceptable ranking is derived. The results obtained by using both perceptual ordering and relevance feedback are ranked lists. The difference is that the former usually ranks all (M-1) or M images while the latter only ranks the top N images retrieved by a search engine. Perceptual ordering and relevance feedback are time-consuming when a large number of images are involved. In addition, it is also difficult for human observers to rank a huge number of images accurately.

Pair-of-Pairs Comparison

Compared with pair-wise comparison, in pair-of-pairs experiments [Charrier et al., 2007] [Clarke et al., 2012] two pairs of images are simultaneously displayed on the monitor and participants are required to judge which pair's perceived difference (or similarity) is greater. In essence, pair-of-pairs comparisons generate a 2nd-order judgement in each trial. In other words, one pair-of-pairs judgement is based on the difference of two pair-wise judgements. During the comparison process, one pair is regarded as a reference for the other pair. Thus, pair-of-pairs methods can generate more precise results than pair-wise comparisons. However, given M images, a complete pair-of-pairs comparison consists of 4^M trials ($2^{O(n)}$ complexity) in total. Thus, time cost also restricts the application of this method when M is large.

Free-Grouping

Another well-known experimental method is free-grouping (free-sorting). Rao and Lohse [1993, 1996], Heaps and Handel [1999] and Halley [2011A] asked participants to group texture images into as many groups as they liked. A rational-valued similarity matrix is obtained by using free-grouping, the "similarity" being the number of times two textures have been put into the same group, divided by the number of opportunities for such grouping to occur.

2.1.3 Summary of Perceptual Texture Similarity

Although none of the methods described above can acquire real-valued similarity data, the majority can provide higher resolution similarity than the Boolean-valued data typically available in texture segmentation, classification and retrieval training and test methods.

2.2 Computational Texture Features

Researchers in computer vision and pattern recognition communities have proposed a large number of texture (or image) features over the last forty years [Haralick, 1973] [Van Gool et al., 1985] [Reed and Buf, 1993] [Tuceryan and Jain, 1993] [Mirmehdi et al., 2009]. As a popular and focused research topic in the field of texture analysis, feature extraction is normally conducted for computing texture properties for use in segmentation, classification and retrieval applications. "Similarity" data can then be simply obtained by applying a distance measure to the difference between the feature vectors of any two textures.

Tuceryan and Jain [1993] divided texture features into five major categories: statistical, geometrical, structural, model-based and signal processing based features. Similarly, texture features were also classified into: signal processing based, statistical, structural and model-based approaches by Mirmehdi et al. [2009]. In addition, according to the form of feature vectors, computational texture features can be divided into histogram-based (see Table 2.1) and non-histogram based features (see Table 2.2). In this section, we will briefly review a number of popular texture features corresponding to the four categories described by Mirmehdi et al. [2009].

2.2.1 Signal Processing Based (Filtering-Based) Features

Many signal processing based features have been obtained by using the energy (or variance) of filter responses produced by applying a filter or a filter bank to an image. Thus, in this thesis, these features are also referred to as "filtering-based features". Gradient filters specified in the spatial domain, such as Roberts Cross [Roberts, 1965], Prewitt operator [Prewitt, 1970], Marr operator [Marr and Hildreth, 1980], Canny detector [Canny, 1986], Sobel operator [Sobel, 1990], Shen-Castan operator [Shen and Castan,

1992], can be used to extract lines, edges, isolate dots, and so on. There also exist many other spatial filters or filter banks, for example, eigenfilters [Ade, 1983], discrete cosine transform based channel filters [Ng et al., 1992], Laws masks[Laws, 1980], Gabor filter [Fogel and Sagi, 1989] or Gabor filter banks [Bovik et al., 1990] [Jain and Farrokhnia, 1991] and Wavelet transforms [Chen and Kundu, 1994], etc.

Filtering-based texture features can also be obtained by applying filters in the frequency domain [Coggins and Jain, 1985] [Jain and Farrokhnia, 1991] [Manjunath and Ma, 1996], especially in the case that it is expensive to implement the kernels in the spatial domain. In this situation, an image is first transformed into the Fourier domain using the Fourier transform (FT) [Lizorkin, 2001]. The Fourier component is then multiplied by the corresponding filter value, followed by application of the inverse Fourier transform (IFT) in order to transform back into the spatial domain. Hence, the time-consuming convolution operation is avoided. It should be noted that, providing the filters are linear, then they can be designed and implemented in either domain.

However, different kinds of linear filters can also be utilised together in order to encode various textures [Varma and Zisserman, 2005], for instance, Ring and Wedge filters [Coggins and Jain, 1985], LM filter bank [Leung and Malik, 2001], S filter bank [Schmid, 2001], and RFS or MR8 filter bank [Varma and Zisserman, 2005].

In addition, quadrature filters based features, e.g. the JSCW (Joint Statistics of Complex Wavelet) [Portilla and Simoncelli, 2000], were designed to extract local phase information from an image.

2.2.2 Statistical Features

Statistical features are designed to describe the spatial distribution of grey level values. Local statistics are generally extracted from a texture image using statistical features, then global statistics are computed from the local statistics and it is these that are used as texture features.

Popular 1st-order statistical features include the mean of the grey levels and the grey level histogram (GLH) [Mirmehdi et al., 2009]. In contrast, the relationship between pairs of pixels throughout the image is characterised by 2nd-order statistical features such as those that can be obtained from the autocorrelation function. Grey level co-occurrence matrices (GLCM) [Haralick et al., 1973] provide one of the most classical

feature sets of this type. Another similar approach is the use of absolute grey level difference histograms (GLADH) [Weszka et al., 1976]. In addition, the histograms of the signed grey level difference (GLSDH), the grey level sum (GLSH) and the combination (GLSDSH) of these were further used by Unser [1986] as texture features. Kim et al. [1999] also designed a surrounding region dependence method (SRDM). Higher order statistical features concerning pixel relationships between three or more pixels, for instance, grey level run length matrix (GLRLM) [Galloway, 1975] and grey level gap length matrix (GLGLM) [Wang et al., 1994] features, encode more complex spatial patterns.

In recent years, local image features have received particular attention for texture analysis tasks. Harwood et al. [1995] introduced four local centre-symmetric covariance based feature sets, including two different local centre-symmetric auto-correlations with linear and rank-order versions (SAC and SRAC), a related covariance measure (SCOV) and a variance ratio (SVR). Following these, a local covariance matrix (CVM) based texture feature set was introduced by Liu and Madiraju [1996]. Zhang et al. [2007] also compared many local image features and kernels for image classification.

Transform-based statistical features have also proven popular. The Radon transform and its generalisation, namely, the Trace transform (TT) were used in the field of texture analysis [Jafari-Khouzani and Soltanian-Zadeh, 2005] [Kadyrov et al., 2002]. Furthermore, Rahtu et al. [2005] proposed an affine invariant image transform based on a probabilistic interpretation of one image, i.e. multi-scale autoconvolution (MSA).

In addition to these texture features, perceptual texture properties were also explicitly modelled using computational statistics [Fujii et al., 2003]. The combination of different computational statistics of perceptual texture properties was also used to obtain texture rankings [Tamura et al., 1978] [Amadasun and King, 1989].

2.2.3 Structural Features

Structural features generally suppose that textures are comprised of primitives which are placed according to certain spatial placement rules [Haralick, 1979] [Vilnrotter et al., 1986]. Either single pixels, small even regions, or line segments can be regarded as primitives. The placement rules are normally described by either modelling geometric relationships between primitives or describing their statistical properties.

Carlucci [1972] used line segments, open polygons and closed polygons as primitives to model textures. Placement rules were syntactically described in a graph-like language. The primal sketch was used by Marr [1982] and Guo et al. [2007] to represent spatial texture features, such as blobs, edges and bars. In addition, Julesz [1981] introduced the concept of textons for representing texture primitives. The concept of textons has also been popularised to filter responses [Leung and Malik, 2001] [Schmid, 2001] [Cula and Dana, 2004] [Zhu et al., 2005] [Varma and Zisserman, 2005] or image exemplars [Varma and Zisserman, 2009]. The occurrence frequency of textons in images (their texton histograms) are often exploited by these methods. This type of features was also known as "Bag-of-Words" (BoW) or "Bag-of-Visual-Words" [Sivic and Zisserman, 2003] [Csurka et al., 2004] [Willamowski et al., 2004]. Similarly, the histogram comparison of gradient magnitudes and/or gradient directions [Ojala et al., 1996], local binary patterns (LBP) [Ojala et al., 2002] and its variants [Ahonen and Pietikäinen, 2009] [Ahonen et al., 2009], local derivatives [Zhang et al., 2010] and local phase information [Ojansivu et al., 2008] makes these methods appear similar to the texton-based or structural approaches.

2.2.4 Model-Based Features

Several texture models were also developed to describe textures, such as fractal models [Mandelbrot, 1982] [Pentland, 1984] [Chaudhuri et al., 1993], (simultaneous) auto-regressive models [Mao and Jain, 1992] [Bennett and Khotanzad, 1998], Markov random field models [Chellappa and Chatterjee, 1985] [Gimel'farb and Zalesny, 1993] and the epitome models [Jojic et al., 2003]. Generative and stochastic models are mainly utilised by these methods and their estimated parameters are used as texture features.

2.2.5 Summary of Computational Texture Features

Although many hybrid features have been designed and can thus be categorised into more than one category, for simplicity, we utilise the four categories introduced by Mirmehdi et al. [2009] to categorise computational texture features. In addition, in this thesis, computational features are also divided into histogram-based (see Table 2.1) and non-histogram based features (see Table 2.2) according to the form of feature vectors.

The goal of this thesis is to discover or develop computational features for estimating perceptual texture similarity. In this case, the computational texture similarity obtained using any features can be regarded as a potential measure of the perceptual texture similarity. Therefore, it is necessary to perform a set of evaluation experiments in order to examine the ability of existing features to estimate perceptual texture similarity.

However, it is not practical to test all existing feature sets. We therefore examine a number of representative texture feature sets. A feature set will be chosen for investigation if it satisfies two criteria: (1) it is popular in the literature; and (2) its source code is published or it can be easily implemented according to the definition in the original publication. In this research, we identified 46 sets of computational features, including either classical or state-of-the-art features, for use in our evaluation experiments. In addition, we implemented five other feature sets: GMAGGDIRCANNY, GDIRCANNY, GDIRSOBEL, GMAGCANNY and GMAGSOBEL (see Table 2.1) by considering the derivation of the GMAGGDIRSOBEL feature set [Ojala et al., 1996]. The 51 feature sets can be categorised into four categories: filtering-based, structural, statistical and model-based features. They can also be categorised into histogram-based and non-histogram based features. Tables 2.1 and 2.2 list the feature sets of these two categories respectively. In Chapter 3, we will introduce the 51 feature sets in more detail.

Identifier	Full Name	Reference
GDIRCANNY	*Canny Gradient Direction*	[Ojala et al., 1996]
GDIRSOBEL	*Sobel Gradient Direction*	
GMAGCANNY	*Canny Gradient Magnitude*	
GMAGGDIRCANNY	*Joint Distribution of Canny GMAG and GDIR*	
GMAGGDIRSOBEL	Joint Distribution of Sobel GMAG and GDIR	
GMAGSOBEL	*Sobel Gradient Magnitude*	
LBPBASIC	Basic Local Binary Patterns	[Ahonen and Pietikäinen, 2009]
LBPDF	Local Derivative Filters Based LBP	
LBPHF	Local Binary Pattern Histogram Fourier Features	[Ahonen et al., 2009]
LBPRIU2	Rotation-Invariant Uniform Local Binary Patterns	[Ojala et al., 2002]
LBPRIU2&VAR	Joint Distribution of LBPRIU2 and VAR	
VAR	Rotation Invariant Local Variance	
LDP	Local Derivative Patterns	[Zhang et al., 2010]
LDPSE	Spatially Enhanced LDP	
RI-LPQ	Rotation-Invariant Local Phase Quantisation	[Ojansivu et al., 2008]
SAC	Centre-Symmetric Auto-correlation	[Harwood et al., 1995]
SRAC	Centre-Symmetric Rank-Order Auto-correlation	
SVR	Centre-Symmetric Variance Ratio	
VZ-MR8	Varma-Zisserman MR8 Textons	[Varma and Zisserman, 2005]
VZ-MRF	Varma-Zisserman Markov Random Field Textons	[Varma and Zisserman, 2009]
VZ-NEIGHBORHOOD	Varma-Zisserman Neighbourhood Textons	

Table 2.1: Histogram-based texture feature sets chosen in Section 2.2. Italic fonts mean the feature sets which are not included in the original publicaions.

Identifier	Full Name	Reference
ACF	Autocorrelation Function	[Fujii et al., 2003]
CVM	Covariance Matrix	[Liu and Madiraju, 1996]
DCT	Discrete Cosine Transform Based Channel Filters	[Ng et al., 1992]
EIGENFILTER	Eigen Filters	[Ade, 1983]
FRACTALDIMENSION	Fractal Dimension	[Chaudhuri et al., 1993]
GABORBOVIK	Bovik Localised Gabor Filters	[Bovik et al., 1990]
GABORENERGY	Gabor Energy Filters	[Fogel and Sagi, 1989]
GABORJFFD	Dyadic Gabor Filter Bank (Frequency Domain)	[Jain and Farrokhnia, 1991]
GABORJFSD	Dyadic Gabor Filter Bank (Spatial Domain)	
GABORMM	Manjunath-Ma Gabor Wavelet Filter Bank	[Manjunath and Ma, 1996]
GLADH	Absolute Grey Level Differences Histograms	[Weszka et al., 1976]
GLCM	Grey Level Co-occurrence Matrices	[Haralick et al., 1973]
GLSDH	Signed Grey Level Differences Histograms	[Unser, 1986]
GLSDSH	Signed Grey Level Differences and Sum Histograms	
GLSH	Grey Level Sum Histograms	
GLGLM	Grey Level Gap Length Matrix	[Wang et al., 1994]
GLH	Grey Level Histogram	[Mirmehdi et al., 2009]
GLRLM	Grey Level Run Length Matrix	[Galloway, 1975]
GMRF	Gaussian Markov Random Field	[Chellappa and Chatterjee, 1985]
JSCW	Joint Statistics of Complex Wavelet	[Portilla and Simoncelli, 2000]
LAWS	Laws Masks	[Laws, 1980]
LM	Leung-Malik Filter Set	[Leung and Malik, 2001]
MRSAR	Multi-resolution Simultaneous Autoregressive	[Mao and Jain, 1992]
MSA	Multi-scale Autoconvolution	[Rahtu et al., 2005]
MR8	Maximum Response Filter Set	[Varma and Zisserman, 2005]
RFS	Root Filter Set	
RING & WEDGE	Ring and Wedge Filters	[Coggins and Jain, 1985]
S	Schmid Filter Set	[Schmid, 2001]
SRDM	Surrounding Region Dependence Method	[Kim and Park, 1999]
TT	The Trace Transform	[Kadyrov and Petrou, 2001]

Table 2.2: Non-histogram based texture feature sets chosen in Section 2.2.

2.3 Texture Databases

In this section, we investigate whether or not any large texture databases exist that provide a set of readily available higher resolution perceptual similarity data.

2.3.1 Criteria for the Selection of Texture Databases

In order to derive reliable experimental results, an appropriate texture database is required. A large (e.g. 5000) dataset is normally used in image retrieval [He et al., 2004] [Xu et al., 2013] or image classification [Zhang et al., 2007]. However, it is not practical

to source such a large number of texture samples complete with higher resolution (non-binary) similarity data.

Since different illumination and viewpoint conditions affect both human perception and computation of texture features [Halley, 2011A], it is necessary to keep such conditions constant in order to reduce their influence.

Hence, the three criteria used for the evaluation of existing databases are listed below:

(1) images should have been captured under “constant illumination”;

(2) images should have been captured under “constant viewpoint”; and

(3) the database should be available with higher resolution “perceptual similarity data”.

2.3.2 Review of Existing Texture Databases

In this subsection, we review 14 published texture databases according to the three criteria presented above.

Brodatz

As one of the most popular texture databases among computer vision and pattern recognition communities, *Brodatz* consists of 112 textures taken from a photo album [Brodatz, 1966]. Textures in *Brodatz* have also been used to obtain higher resolution perceptual texture similarity [Rao and Lohse, 1993] [Payne et al., 1999] [Long et al., 2000]. However, the illumination conditions under which the images were acquired are unknown. Only the third criterion is satisfied by the *Brodatz* database.

VisTex

Although *VisTex* [MIT, 1995] includes four main subsets (reference textures, texture scenes, video textures and video orbits), only reference textures are related to this study. The reference texture subset contains 167 textures. However, the acquisition conditions for the *VisTex* database did not conform to strict frontal plane perspectives and constant lighting conditions. Thus, this database fails to satisfy any of the three criteria.

Meastex

In total 69 artificial and natural textures are included in *Meastex* [Smith and Burns, 1997]. All texture images were captured under undocumented illumination and viewpoint conditions. This database satisfies none of our three criteria.

CUReT

The *CUReT* database [Dana et al., 1999] was acquired in order to capture the visual appearance of real-world surfaces. In total 61 textures are comprised of this database with over 200 images acquired under different, documented, viewing and illumination directions. However, no higher resolution perceptual similarity data is available for this database.

Outex

In total 320 surface textures are included in *Outex* [Ojala et al., 2002a]. All texture images were acquired under different resolutions, illumination angles and rotation angles. However, a part of images were obtained from the same sample, which impairs the reliability of the inter-class variation. Similar to the *CUReT* database, one subset could be selected from *Outex*. All images in the subset would share the same resolution, illumination angle and rotation angle conditions. However, no any higher resolution perceptual similarity data is available for this database.

PhoTex

In order to provide photometric data for texture analysis, the *PhoTex* [Texture Lab, 2003] database was introduced with 64 surface textures. Texture images were acquired under controlled illumination conditions and constant viewpoint. At the same time, height maps of the textures were also provided. In this case, any illumination condition can be used to relight the height maps for acquiring illumination-constant images. However, this database does not provide any higher resolution perceptual similarity data.

Ponce

The *Ponce* [Lazebnik, 2003] texture database consists of 25 textures with 40 images per texture. The images of each texture were taken at different viewing angles and unknown

illuminations. Thus, this database at most satisfies the second criterion if and only if a subset of images is selected under a fixed viewing angle.

KTH-TIPS & KTH-TIPS2

KTH-TIPS [CVAP, 2004] was introduced in order to extend *CUReT* by imaging one subset of it under various scales, poses and illuminations. In total 10 textures of *CUReT* were used. Furthermore, 11 textures from *CUReT* were included in *KTH-TIPS2* [CVAP, 2005]. Even if one image is selected for each texture under the same illumination and pose conditions, there still remains the problem that no higher resolution perceptual similarity data is available.

UIUCTex

UIUCTex [Lazebnik et al., 2005] consists of 25 different textures. Forty images of each texture were imaged under different viewpoints and scales. But illumination conditions were uncontrolled. Thus, the first and third criteria are not satisfied.

Tex1 & MoMA

120 natural and synthetic textures are included in *Tex1* [Emrith, 2008]. In addition, *MoMA* [Emrith, 2008] consists of another set of 100 textures. Free-grouping experiments were conducted on the two databases with 8 and 19 subjects respectively. Correspondingly, two rational-valued similarity matrices were obtained. Similar to the *PhoTex* database, both *Tex1* and *MoMA* were published with height maps of the textures. Consequently, it is feasible to use arbitrary illumination condition to relight the height maps for obtaining illumination-constant images.

Pertex

Halley [2011A] acquired a texture database with 500 surface textures under constant viewpoint and named it as *Tex500*. Using this database, a free-grouping experiment was conducted and a rational-valued perceptual similarity matrix was obtained. Halley [2011B] further obtained an unconfidential subset (334) of this database as well as an associated 334×334 perceptual similarity matrix subset. The *Pertex* database also provides the height map of each texture. Hence, the height maps can be relit under any given illumination. In addition, in order to prevent observers from grouping similar tex-

tures according to their dominate directions, all directional textures were rotated to have a horizontal dominate direction (if exhibited). Consequently, *Pertex* satisfies all three criteria and contains a "significant" number of textures compared with its counterparts.

STex

The Salzburg Texture Image Database (*STex*) [Kwitt and Meerwald] consists of a collection of 476 textures. Only one image was taken for each texture. The number of the textures is attractive. However, illuminations and viewpoints are not specified and there is no perceptual similarity data with it. Thus, none of the three criteria are statisfied.

2.3.3 Summary of Texture Databases

Table 2.3 summarises the 14 published texture databases according to one property and the three criteria introduced in Section 2.3.1. Only the *Tex1*, *MoMA* and *Pertex* texture databases satisfy all three criteria. However, the numbers of textures in *Tex1* and *MoMA* are only 120 and 100 respectively. Thus, *Pertex* with its higher resolution rational-valued perceptual similarity data was selected for this study.

Texture Database	Number of Textures	Criteria		
		Constant Illumination	Constant Viewpoint	HR Perceptual Similarity Available
Brodatz [Brodatz, 1966]	112	✗	✗	✓
VisTex [MIT, 1995]	167	✗	✗	✗
Meastex [Smith and Burns, 1997]	69	✗	✗	✗
CUReT [Dana et al., 1999]	61	✓	✓	✗
Outex [Ojala et al., 2002a]	320	✓	✓	✗
PhoTex [Texture Lab, 2003]	64	✓	✓	✗
Ponce [Lazebnik, 2003]	25	✗	✓	✗
KTH-TIPS [CVAP, 2004]	10	✓	✓	✗
KTH-TIPS2 [CVAP, 2005]	11	✓	✓	✗
UIUCTex [Lazebnik et al., 2005]	25	✗	✓	✗
Tex1 [Emrith, 2008]	120	✓	✓	✓
MoMA [Emrith, 2008]	100	✓	✓	✓
Pertex [Halley, 2011B]	334	✓	✓	✓
STex [Kwitt and Meerwald]	476	✗	✗	✗

Table 2.3: Summary of 14 published texture databases reviewed in Section 2.3.2 according to one property and three criteria (HR: higher resolution).

Since computational similarity is obtained in a different way from perceptual similarity, it is likely that they are represented in different scale spaces. In this case, it is difficult to compare their numerical magnitude values. However, pair-of-pairs judgements or rankings only use the relative magnitude of similarity data. By using the pair-of-pairs judgements or rankings obtained from different sources, direct comparison of the numerical magnitude of different sources of similarity is avoided. Hence, we chose pair-of-pairs judgements and rankings (both are higher resolution data) as the specific forms of texture similarity assessment in this thesis.

2.4 Performance Measures

As discussed in Section 2.3.3, pair-of-pairs judgements and rankings are chosen as the specific forms of evaluation in this thesis. In order to compare computational pair-of-pairs judgements or rankings with their perceptual counterparts, certain performance measures are required.

Generally speaking, performance measures used by texture segmentation, classification and retrieval can be divided into accuracy-based and rank-based measures. Accuracy-based measures [Randen and Husøy, 1999] [Varma and Zisserman, 2009] [Khelifi and Jiang, 2011] only consider the percentage correctness and are normally employed for measuring Boolean-valued similarity. On the other hand, rank-based measures [Payne et al., 1999] [Long et al., 2000] take ranks of the similarity into consideration and are generally utilised for measuring the performance of texture retrieval or other information retrieval applications. In addition, measures for comparing two rankings [Diaconis and Graham, 1977] [Fagin et al., 2003] [Bar-Ilan et al., 2006] can also be regarded as rank-based measures.

2.4.1 Accuracy-Based Performance Measures

Accuracy-based performance measures include classification accuracy [Randen and Husøy, 1999] [Varma and Zisserman, 2009], precision and recall [Khelifi and Jiang, 2011].

Classification Accuracy

Classification accuracy (see Equation (2.1)) is normally used to measure the performance of texture (or image) classification or segmentation. Similarly, error classification rate was also applied in some publications. Ideally, if all elements (pixels, image patches, or images) are classified correctly, the classification accuracy is 100%.

$$CA(\%) = \frac{Number\ of\ Elements\ that\ are\ Classified\ Correctly}{Number\ of\ All\ Elements} \times 100 \qquad (2.1)$$

Precision and Recall

Precision (Equation (2.2)) and recall (Equation (2.3)) are two classical measures for information retrieval and lie in the range of [0, 1]. Generally speaking, precision measures the effectiveness of a retrieval algorithm while recall is a measure of the completeness of one algorithm.

$$Precision = \frac{Number\ of\ Relevant\ Samples\ Retrieved}{Number\ of\ Samples\ Retrieved} \qquad (2.2)$$

$$Recall = \frac{Number\ of\ Relevant\ Samples\ Retrieved}{Number\ of\ All\ Relevant\ Samples} \qquad (2.3)$$

Summary of Accuracy-Based Performance Measures

Generally speaking, accuracy-based measures are computed by counting the number of correctly classified elements (pixels, patches or images). None of these measures consider the relative similarity between elements of different classes as well as elements within the same class. Thus, those measures are unsuitable for comparing two ranked lists. However, if the comparison only concerns the number of correct items, then accuracy-based measures are suitable. Since the comparison of two pair-of-pairs judgements produces a “1” or “0” result, the performance measure of comparing two sets of pair-of-pairs judgements is similar to classification accuracy. Consequently, an accuracy-based measure is chosen for the comparison of two sets of pair-of-pairs judgements.

2.4.2 Rank-Based Performance Measures

Rank-based measures are computed using ranks (sequence indices). This type of measures is suitable for measuring retrieval performance or comparing two rankings.

Spearman's Rank Correlation Coefficient

Spearman's rank correlation coefficient (also Spearman's *rho* or ρ) [Field, 2009] is a nonparametric measure of statistical dependence between two variables. It measures the magnitude of the correlation of two variables and normally lies in the range of [-1, 1]. ρ is defined as the Pearson's correlation coefficient [Field, 2009] between two ranked variables. Given two sample variables: X_i and Y_i ($i = 1,2,\ldots,n$), two sets of ranks: r_i and r_i' ($i = 1,2,\ldots,n$) are obtained respectively. ρ is computed as:

$$\rho = \frac{\sum_{i=1}^{n}(r_i-\overline{r})(r_i'-\overline{r'})}{\sqrt{\sum_{i=1}^{n}(r_i-\overline{r})^2(r_i'-\overline{r'})^2}}, \tag{2.4}$$

where $\overline{r}$ and $\overline{r'}$ are the means of r_i and r_i' respectively.

Kendall's Rank Correlation Coefficient

Similarly, Kendall's rank correlation coefficient (also Kendall's *tau* or τ) [Field, 2009] is also a rank-based measure. Given two sample variables: X_i and Y_i ($i = 1,2,\ldots,n$), two sets of ranks: r_i and r_i' ($i = 1,2,\ldots,n$) are obtained respectively. For one pair of observations (X_i, Y_i) and (X_j, Y_j) ($i,j = 1,2,\ldots,n$), if their ranks (r_i, r_i') and (r_j, r_j') are agreed, i.e. both $r_i > r_j$ and $r_i' > r_j'$ or both $r_i < r_j$ and $r_i' < r_j'$ hold true, they are considered "concordant". Furthermore, (X_i, Y_i) and (X_j, Y_j) are regarded as "discordant" if both $r_i > r_j$ and $r_i' < r_j'$ or both $r_i < r_j$ and $r_i' > r_j'$ hold true. In addition, they are neither "concordant" nor "discordant" if $r_i = r_j$ or $r_i' = r_j'$. Kendall's *tau* is defined as:

$$\tau = \frac{Number\ of\ Concordant\ Pairs - Number\ of\ Discordant\ Pairs}{\frac{1}{2}n(n-1)}. \tag{2.5}$$

Normalised Precision and Normalised Recall

Normalised precision (*NP*) and normalised recall (*NR*) were proposed by Rocchio [1971] to measure the inconsistency between the actual rankings of relevant documents and their "ideal" rankings. Originally, the *NP* and *NR* were defined as:

$$NP = 1 - \frac{\sum_{i=1}^{R}(\log r_i - \log i)}{\log N! - \log((N-R)!R!)}, \tag{2.6}$$

$$NR = 1 - \frac{\sum_{i=1}^{R}(r_i - i)}{(N-R)R}, \tag{2.7}$$

where R is the number of all relevant documents in the database, N is the number of all documents in the database, r_i is the rank order of the i-th relevant document, and i is the "ideal" rank position for the i-th relevant document.

Spearman's Footrule

Similar to *NP* and *NR*, Spearman's footrule [Diaconis and Graham, 1977] only considers relevant subsets of two lists as well. After the relevant subsets are re-sorted, their permutations: r_i and r_i' (i = 1,2,…,n) are obtained. Spearman's footrule is then computed as:

$$SF(r, r') = \sum_{i=1}^{n} |r_i - r_i'|. \tag{2.8}$$

Spearman's footrule can also be normalised by its maximum as:

$$NSF = \begin{cases} \frac{2SF}{n^2}, & n\ is\ even \\ \frac{2SF}{(n-1)(n+1)}, & n\ is\ odd \end{cases}. \tag{2.9}$$

G Measure

G measure (see Equation (2.10)) introduced by Fagin et al. [2003] compares the top N rankings obtained by two search engines, no matter whether they are identical or not. It can be regarded as an extension of normalised recall (see Equation (2.7)). Similarly, the extended normalised precision NP' is defined in Equation (2.11).

$$G = 1 - \frac{\sum_{i=1}^{R}(|r_i - r_i'|) + \sum_{i=1}^{N-R}[(N+1) - r_i] + \sum_{i=1}^{N-R}[(N+1) - r_i']}{N(N+1)}, \tag{2.10}$$

$$NP' = 1 - \frac{\sum_{i=1}^{R}(|\log r_i - \log r_i'|) + \sum_{i=1}^{N-R}[\log(N+1) - \log r_i] + \sum_{i=1}^{N-R}[\log(N+1) - \log r_i']}{2\log\frac{(N+1)^{N+1}}{(N+1)!}}, \quad (2.11)$$

where R is the number of all relevant documents in N retrieved documents (i.e. texture images in our study), r_i is the rank order of i-th relevant/irrelevant document retrieved by a search engine (i.e. a computational feature set in our study), and r_i' is the "ideal" rank order (i.e. the rank order of i-th texture image ranked by human observers in our study) of the i-th relevant/irrelevant document retrieved.

The *G* measure considers not only relevant but also irrelevant documents of the two rankings and is able to capture human intuition better [Bar-Ilan et al., 2006]. Given that relevant documents lie in the same sequence in two rankings, if the indices of their positions are more similar, the value of the *G* measure should be larger. However, the magnitude of the change in *G* for a given relevant set is quite small and is mainly affected by the size of the relevant set.

M Measure

Bar-Ilan et al. [2006] further developed another measure, i.e. the *M* measure, in order to encode the intuition in which identical or nearly-identical rankings of the top *N* documents are more important to users than those among the lower placed documents. Given the same notation as that in Equation (2.10), it is defined as:

$$M = 1 - \frac{\sum_{i=1}^{R}(\left|\frac{1}{r_i} - \frac{1}{r_i'}\right|) + \sum_{i=1}^{N-R}(\frac{1}{r_i} - \frac{1}{N+1}) + \sum_{i=1}^{N-R}(\frac{1}{r_i'} - \frac{1}{N+1})}{2\sum_{i=1}^{N}(\frac{1}{i} - \frac{1}{N+1})}. \quad (2.12)$$

Obviously, *M* gives a higher weight to the higher ranking documents. It should be pointed out that the same formula as Equation (2.11) is obtained if each reciprocal in the numerator of Equation (2.12) is replaced by its log value. The *M* measure has been used for examining the relevance between rankings obtained using search engines and rankings sorted by humans [Bar-Ilan et al., 2007] [Metaxas et al., 2009].

Summary of Rank-Based Measures

Rank-based measures generally take ranks of the similarity into consideration. The "ideal" orders of relevant documents used by *NP* and *NR* are 1, 2,…, *R*, i.e. the retrieval sequences of the relevant documents. In essence, these two measures compare actual ranks and retrieval ranks of the relevant documents.

If only the top *N* documents in the two rankings are compared, irrelevant documents should also be considered. Spearman's footrule ignores irrelevant documents of two rankings. Thus, *NP*, *NR* and Spearman's footrule do not satisfy our requirement.

In addition, Spearman's and Kendall's rank correlation coefficients first sort two complete input lists and then compute the measure from the new ranks. In contrast, *G* and *M* measures use the two ranked lists directly. However, Spearman's *rho* and Kendall's *tau* can only compare two identical lists. In contrast, *G* and *M* measures are able to compare two either identical or non-identical ranked lists. Both of these can compare the top *N* elements (normally non-identical) in the ranked lists. This is important for image retrieval because images at top positions are most attractive to users [Xu et al., 2011]. Consequently, the *G* and *M* measures were chosen for comparing two texture rankings.

2.5 Image Properties

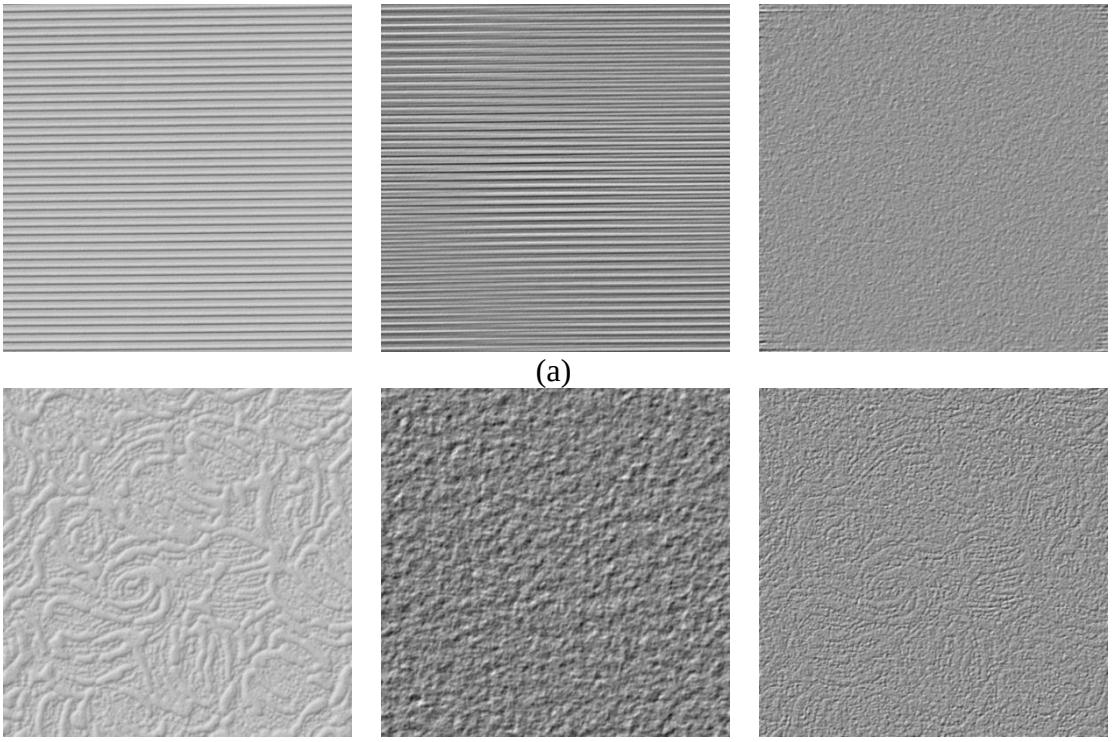
Spatial and frequency (Fourier) domains are normally used for applications of texture analysis. In the frequency domain, power (magnitude) and phase spectra are obtained after the Fourier transform (FT) [Lizorkin, 2001] is carried out. In the literature, the importance of phase spectra to image/texture structure has been discussed [Oppenheim and Lim, 1991] [Kovesi, 2000] [Hansen and Hess, 2007] [Emrith et al., 2010]. However, the power spectrum has also been shown to represent important content for certain natural images [Tadmor and Tolhurst, 1993].

Structural texture analysis [Haralick, 1979] [Vilnrotter et al., 1986], on the other hand, mainly considers texture in the spatial domain. Texture structure is normally related to placement rules of basic elements (or primitives). Textons [Julesz, 1981] and contours [Marr, 1982] [Guo et al., 2007] are two types of popular texture elements. Textons are normally extracted based on image exemplars. Besides, image exemplars are also utilised by other neighbourhood-based features.

2.5.1 Power and Phase Spectra

If the phase spectrum of one image is replaced by a randomised matrix (e.g. a white noise matrix), and the inverse Fourier transform is then performed on it and the actual magnitude matrix, one phase-randomised (power-only) image (see Figure 2.3 (middle))

is generated [Oppenheim and Lim, 1991] [Emrith et al., 2010]. In other words, the phase-randomised image still contains its original power spectrum but only carries randomised phase information and thus encodes only information concerning periodicity (see Figure 2.3 (a) (middle)). These images, as Oppenheim and Lim [1991] observed, differ from a phase-randomised image obtained from an aperiodic image (see Figure 2.3 (b) (middle)) because the "structure" information has been removed.



(a)

(b)

Figure 2.3: Each row presents an original texture image and its phase-randomised (power-only) and power-uniformised (phase-only) property images, from left to right. It can be seen that the periodicity is retained in the phase-randomised image while the aperiodic structure is preserved in the power-uniformised image.

Correspondingly, if the power spectrum of an image is replaced by the uniform distribution, and the inverse Fourier transform is then applied, a power-uniformised (phase-only) image (see Figure 2.3 (right)) is obtained. It can be seen from Figure 2.3 (b) (right) that the aperiodic structure in one image is preserved in its power-uniformised (phase-only) image. However, the periodic pattern in an image is lost (Figure 2.3 (a) (right)) in its power-uniformised image.

2.5.2 Image Exemplars

Image exemplars are normally cropped from one image and contain local image characteristics. In the field of texture analysis, exemplar-based texture synthesis has received much attention [Efros and Freeman, 2001] [Liang et al., 2001]. On the other hand, image exemplars are also used by neighbourhood-based features [Varma and Zisserman, 2005&2009] [Weszka et al., 1976] [Harwood et al., 1995] [Mao and Jain, 1992] [Chaudhuri et al., 1993]. Figure 2.4 presents two texture images with square exemplars.

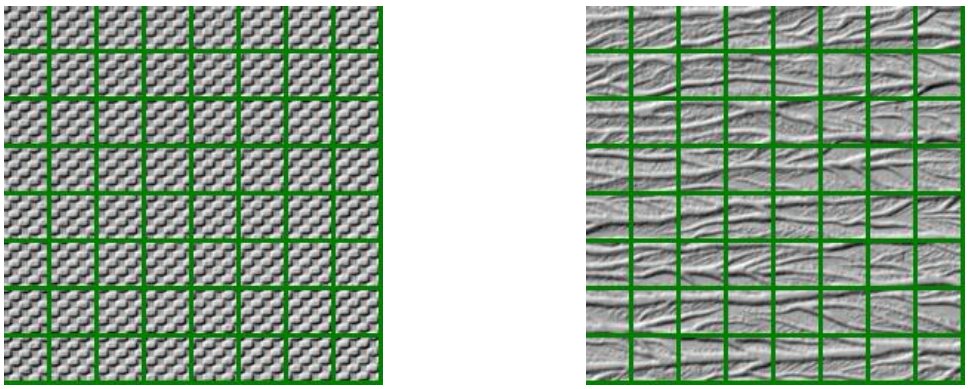


Figure 2.4: Two texture images with exemplars.

2.5.3 Contours

Irregular structure (see Figure 2.5 (a) (left)) is normally considered to be encoded by phase spectra rather than power spectra (see Section 2.5.1). However, global phase is difficult to unwrap [Ying, 2006]. As an alternative, contours/edges (see Figure 2.5 (a) (right)) are intuitive for the representation of this type of structure. Contours can be extracted using edge detectors, such as Roberts Cross [Roberts, 1965], Prewitt operator [Prewitt, 1970], Marr operator [Marr and Hildreth, 1980], Canny detector [Canny, 1986], Sobel operator [Sobel, 1990], Shen-Castan operator [Shen and Castan, 1992] and Kovesi detector [Kovesi, 2003], and post-processing. In addition, contours can also be obtained using perceptual grouping [Li et al., 2010] or clustering [Arbelaez et al., 2011] techniques.

2.5.4 Summary of Image Properties

Global power information cannot encode aperiodic image structure [Oppenheim and Lim, 1991] but can be used to represent image periodicity [Liu, and Picard, 1998]. On

the other hand, the phase spectrum is believed to encode aperiodic image structure [Oppenheim and Lim, 1991]. However, phase unwrapping is required. Since it is still an open problem [Ying, 2006], we ignore the exploitation of the phase spectrum in this study.

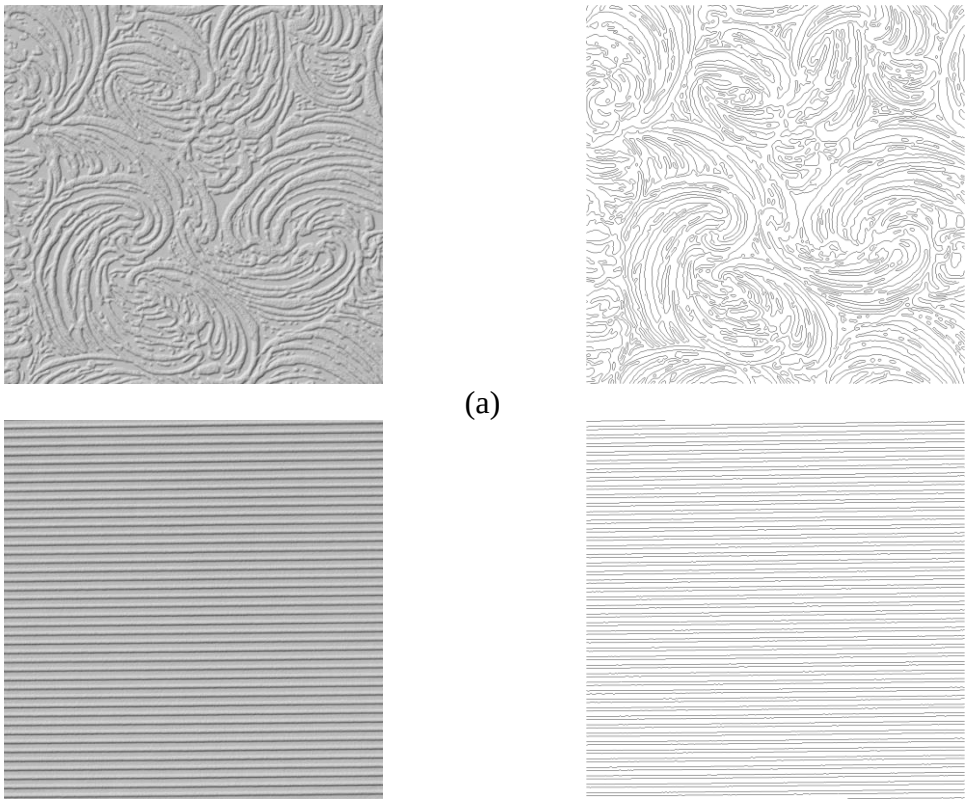


Figure 2.5: Original texture images (left) and their contour maps (right). The contour maps were extracted using the Canny edge detector [Canny, 1986].

Image exemplars are utilised by neighbourhood-based features and are able to encode local texture information. However, it is difficult to determine the optimal size of image exemplars over different features. In addition, computational cost becomes rapidly heavier with the increase of the size of exemplars. Furthermore, it will also produce an "averaging effect" and decrease the discriminatory power of features [Mao and Jain, 1992] when large exemplars are used. Thus, the size of image exemplars is limited and this limits the spatial extent exploited by those features. Although global statistics can also be computed from image exemplars, only 1st- or 2nd-order statistics are generally used (also see Table 3.2) and these statistics do not encode aperiodic spatial relationships between image exemplars. As a result, image exemplar based features normally cannot capture aperiodic long-range texture information.

Since contours can be derived from both periodic and aperiodic textures (see Figure 2.5), it is considered that contour maps are more suitable for representing both types of long-range texture information compared with phase spectra, power spectra or image exemplars. However, it is difficult to extract contours accurately when texture structures are small. This limits the representation ability of the contour.

2.6 Human Perception of Contours

As discussed in Section 2.5.4, the contour can represent long-range periodic or aperiodic texture structure. The studies on human perception of contours mainly involve two topics, namely, object outline identification [De Winter and Wagemans, 2004, 2008A, 2008B] [Panis et al., 2008] [Sassi et al., 2010] and contour integration (see Figure 2.6) [Field et al., 1993] [Pettet et al., 1998] [Braun, 1999] [Hansen and Hess, 2006]. Outline identification generally concerns the influence of the outline [De Winter and Wagemans, 2008B] or points/segments of the outline [De Winter and Wagemans, 2004&2008A] [Panis et al., 2008] [Sassi et al., 2010] on the identification of one object. On the other hand, contour integration mainly investigates how humans integrate discontinuous contour segments from a scattered background into a complete contour. Furthermore, contour intergration is also associated with long-range interactions [Polot, 1999].

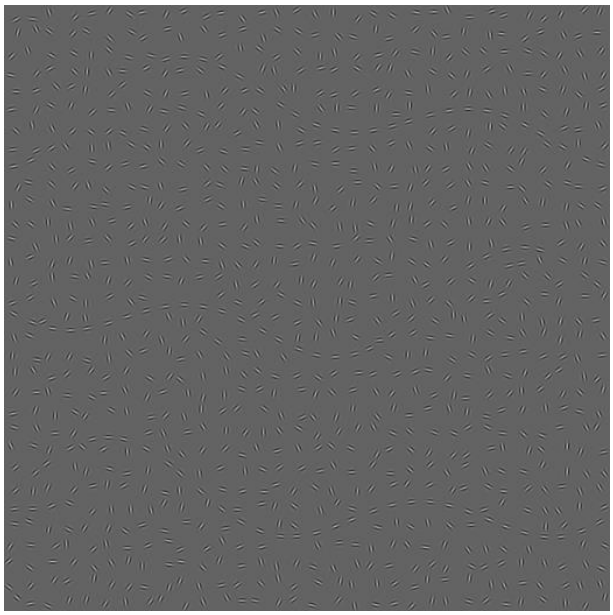


Figure 2.6: An example of contour integration. We can still recognise three contours made up of three sets of collinear Gabor elements from a scattered background.

2.6.1 Outline Identification

De Winter and Wagemans [2004] summarised their five studies on contour-based object identification. Further studies confirmed the importance of curvature extrema using full outline versions because locations at or near curvature extrema were mainly marked as salient points by human observers [De Winter and Wagemans, 2008A&B]. Panis et al. [2008] also used this set of outlines in order to investigate whether or not curved contour segments are most important in shape perception. It was found that fragments located at salient points did not necessarily yield better identification performance.

Furthermore, Sassi et al. [2010] investigated contour integration and texture segmentation using outlines of everyday objects. Each stimulus was comprised of the Gabor elements located and oriented along the outline of an object collinearly. The Gabor contour was surrounded by an evenly distributed Gabor elements field. Experimental results provided norms for the identifiability and name agreement. All explanations were based on a theory of the identification process is divided into two stages: (1) an early stage in which the fragmented contour is grouped or integrated in a primarily bottom-up manner; and (2) a later top-down stage during which the inferred contour shape is matched to representations in memory. In essence, the first stage is a contour integration process.

2.6.2 Contour Integration

Gestalt law [Todorovic, D., 2008] has been used to interpret a number of phenomena for the purpose of illustrating the importance of continuity in human perception. Field et al. [1993] investigated how continuity may be represented by a visual system that filters spatial data using arrays of cells that are selective in terms of orientation and spatial frequency. Kovács et al. [1993] also studied the influence of closure on contour integration. Experimental results implied that the extent of interaction between locally connected detectors is enhanced in relation to the global stimulus structure. However, it was found that this kind of enhancement cannot be predicted by local rules of grouping. In contrast, it is suggested that the connection of collinear segments was greatly affected by the global arrangement.

Lately, Pettet et al. [1998] examined the constraints on long-range interactions for mediating contour integration. Furthermore, Pennefather et al. [1999] conducted a second-order contour integration experiment in which the visibility of the contour was con-

trolled by changing the background element density. It was noted that when the average spacing of Gabor elements in the background decreases below the spacing in the contour, the contour can only be detected based on second-order cues. Braun [1999] also evaluated human performance for detecting the Gestalt-type grouping. What he found was that the detectability of salient contours reaches a peak when they comprise no less than 10 elements and are presented for over 200 ms. The importance of local absolute spatial phase to contour integration was also examined by Hansen and Hess [2006]. A significant main effect of phase was found when the element-to-path angle was set at 90º.

2.6.3 Long-Range Interactions

Long-Range Interactions

Around two centuries ago, Mach and Hering suggested that each region of the retina probably interacts with a number of other distant regions [Spillmann and Werner, 1996]. Field et al. [1993] utilised the concept of the "association field" which integrates information across neighbouring filters tuned to similar orientations to explain the influence of continuity. Polat and Sagi [1994] also studied lateral interactions between spatial filters. Grouping collinear line segments into smooth curves was found to account for the interactions. Generally speaking, classical receptive field (CRF) models are utilised to explain local perceptual effects, including border contrast and Mach bands [Spillmann and Werner, 1996]. However, these models are not suitable for explaining some global perceptual phenomenon, such as the perception of illusory contours, area contrast, colour constancy, depth planes, coherent motion and texture contrast. In this case, long-range interactions account for these effects [Spillmann and Werner, 1996].

Furthermore, Polat [1999] found that perceptual learning involves a larger range of interactions in early vision. It was believed that long-range interactions are produced by chains of local interactions. Thus, long-range interaction cannot take effect until short- and medium-range interactions occur. This study was followed by the work of Brincat and Westheimer [2000]. They found that facilitating interactions are dependent on the contrast polarity of the stimuli with smaller gaps between stimuli (short-range effects). On the other hand, it was found that only co-linearity of the stimuli is necessary for the production of the facilitation with larger gaps (long-range effects). A similar conclusion

was drawn by Tzvetanov and Dresp [2002], i.e. that short-range and long-range effects are produced by facilitating interactions between targets and inducing orientations. Recently, short-range and long-range surround modulation was also investigated by Nurminen et al. [2010] based on the perceived contrast of a centre and its surround. Long-range facilitation of perceived contrast was found to be involved in the surround modulation rather than the well-known suppression.

The Relationship between Long-Range Interactions and the Spatial Extent Exploited by Computational Features

In the literature, lateral interactions between stimuli took place when different angular separations were involved [Tzvetanov and Simon, 2006]. Normally, two types of lateral interactions are observed when the early human visual system and its spatial computational architecture are investigated using small collinear stimuli. The two interactions, namely, "short-range interactions" and "long-range interactions", are utilised for different sizes of spatial separation between stimuli. Considering the periodicity of stimuli, we divide long-range interactions into: "periodic long-range interactions" and "aperiodic long-range interactions".

However, to the best of our knowledge, there is no specific definition of the "spatial extent" for discriminating long-range interactions from short-range interactions in the literature related to vision science or computer vision. In Chapter 3, we surveyed 46 computational texture feature sets and found that none of these feature sets compute higher order statistics from an image region over 25×25 pixels except filtering-based feature sets. Nevertheless, the majority of the filtering-based features examined in this study only use power spectra which canot be used to encode aperiodic image structure. Therefore, we ignore the spatial extent exploited by these features and only consider its effect on other types of features.

In this thesis, the higher order spatial relationship between pixels in a spatial extent which is no more than 25×25 pixels (see Tables 3.1 and 3.2) is referred to as "short-range interactions". On the other hand, the higher order spatial relationship of the pixels in a spatial extent which is greater than 25×25 pixels is referred to as "long-range interactions". Given the experimental setup introduced in Section 7.2.2, a square area of 25×25 subtends approximately 0.73º of visual angle.

Given an image and a certain spatial extent, if we downsample this image to a smaller resolution, the spatial extent will contain larger image structure (see Figure 2.7 for example). However, the texture scale issue is outside the scope of this thesis. In other words, given a particular spatial extent, we use the same set of image sizes for our multi-resolution approach (see Figure 2.7).

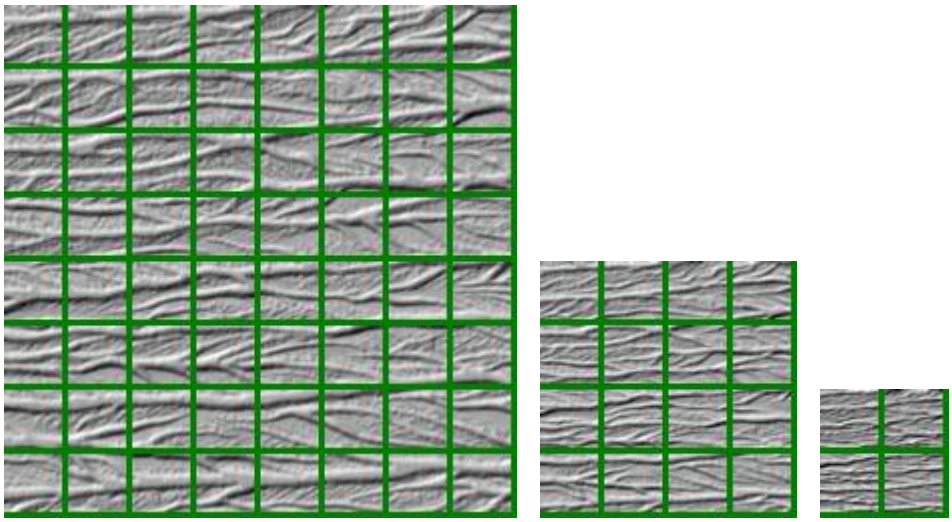


Figure 2.7: An example of the scale issue. The same spatial extent (32×32 pixels) spans different sizes of image structure when three different resolutions of an image are considered.

First, Second and Higher Order Statistics

Contours represent aperiodic and periodic interactions over longer range spatial extent. These interactions can be represented by different orders of spatial statistics. Since 1st-order statistics computed from an image do not encode the relationship between different pixels, they cannot encode both long-range interactions.

As a 2nd-order statistic, the dipole histogram is able to uniquely determine one finite image [Chubb and Yellott, 2000]. However, the computational complexity for obtaining a complete dipole histogram restricts its practical use. In this study, we leave out this "ideal" 2nd-order statistic. On the other hand, commonplace 2nd-order statistics include co-occurrence matrices and the autocorrelation function (ACF). The former are normally calculated based on incomplete pixel pairs. Although the latter computes information from all pixel pairs, the computation is "lossy" as each value of the autocorrelation function is effectively the sum of the products of all pixel pairs at a particular displacement vector. As a result, co-occurrence matrices and the autocorrelation function cannot

capture all structure information in an image. In this thesis, if we do not give any specific comment otherwise, the 2nd-order statistic only denotes these commonplace 2nd-order statistics. However, the power spectrum can be used to compute the ACF via the inverse Fourier transform. Since the power spectrum cannot retain aperiodic image structure [Oppenheim and Lim, 1991] and is generally associated with periodic image structure [Liu, and Picard, 1998] naturally, the ACF is only able to capture periodic image structure as well. In this situation, 2nd-order statistics cannot capture aperiodic long-range interactions but can encode periodic long-range interactions.

Consequently, higher order statistics (HoS) computed at longer spatial extent are required to capture aperiodic long-range image structure.

2.6.4 Summary for Human Perception of Contours

To summarise, contours (outlines) were found to play important roles in the identification of objects [De Winter and Wagemans, 2004, 2008A, 2008B] [Panis et al., 2008] [Sassi et al., 2010] and it is also well-known that humans are extremely adept at exploiting the long-range visual interactions evident in contour information [Field et al., 1993] [Pettet et al., 1998] [Hansen and Hess, 2006]. In order to capture aperiodic long-range interactions, higher order statistics (HoS) are required to be extracted from long-range spatial extent because 2nd-order statistics can only encode periodic long-range interactions.

2.7 Conclusions

In this chapter, we first reviewed two kinds of perceptual texture similarity and chose higher resolution perceptual texture similarity for use in our research. We then examined a large number of existing computational texture feature sets and chose 46 of these for further investigation. In addition, we investigated 14 published texture databases. *Pertex* [Halley, 2011B] was chosen for this study because it contains diverse textures and has higher resolution rational-valued perceptual similarity data. Both pair-of-pairs judgement and ranking were chosen as the tasks which will be used to assess the rational-valued similarity data.

We also reviewed two types of performance measures which were designed for different applications. It was found that: (1) an accuracy-based measure can be used for measuring the performance of the comparison of two sets of pair-of-pairs judgements; and (2) *G* and *M* measures, due to [Fagin et al., 2003] [Bar-Ilan et al., 2006], are suitable for measuring the consistency of two texture rankings.

In addition, four popular image properties were investigated in Section 2.5. Contours are found to be able to represent long-range periodic or aperiodic texture information. Finally, we investigated two popular research topics in vision science. It was found that contours (outlines) play an important role in the identification of objects [De Winter and Wagemans, 2004, 2008A, 2008B] [Panis et al., 2008] [Sassi et al., 2010], and that they are also a good candidate for encoding aperiodic long-range interactions.

Although long-range interactions have been extensively researched in the vision science literature, we found no definition of the spatial extent of "long-range". We have found only one computer vision publication [Wang et al., 2014] that uses these terms, but again no specific definition of "long-range" is given. We also discussed the order of statistics and concluded that higher order statistics are needed to capture aperiodic image structure. In this thesis, we consider long-range to be greater than 25×25 pixels.

In the next chapter, the 46 computational feature sets that we have identified in Section 2.2 will be briefly reviewed. In particular, the spatial extent exploited by these feature sets in the first stage of feature extraction, and the statistical properties of each feature set in the first and second steps of feature extraction are also discussed.

Chapter 3

Computational Texture Features

3.1 Introduction

In Section 2.6.3, we concluded that periodic long-range interactions can be encoded using 2nd-order statistics calculated over the appropriate spatial extent. However, aperiodic interactions can only be captured using higher order statistics. Therefore, the spatial extent utilised by higher order computational features is important to the ability of features to encode aperiodic long-range interactions.

We identified 46 sets of computational texture features (see Tables 2.1 and 2.2) in Section 2.2. In general, texture features can be divided into four categories: filtering-based (signal processing based), statistical, structural and model-based features [Mirmehdi et al., 2009]. However, many feature sets can also be classified into more than one category. Thus, for simplicity, we only choose one category for each feature set in this study.

For image-based analysis problems, such as texture classification and retrieval, feature extraction is normally conducted in two stages. In the first stage, local 2nd-order or higher order statistics are normally computed based on local neighbourhoods (or lines) in order to capture local structure. For the purpose of encoding global image characteristics and also reducing computational complexity, 1st- or 2nd-order features are generally calculated from the outputs of the local features. In this situation, the spatial extent exploited in the first stage is important for a feature to capture aperiodic long-range interactions.

On the other hand, for pixel-based tasks, for example, texture segmentation (including pixel-based texture classification), local features are also extracted but feature maps are

utilised as the input data of the segmentation operation. However, the use of feature maps for computing (image-based) texture similarity is time-consuming and is not applicable due to their high dimensionalities. Since this thesis is limited to investigating image-based texture similarity estimation, one more step should be appended to the local feature extraction stage if these features are used in our study.

Inspired by this, a two-stage feature extraction model that we term the "local-global model" is proposed in this chapter. We then briefly review the 46 feature sets in terms of feature category, their statistical properties in both stages and the spatial extent that they exploit in the first stage (the spatial extent of the second stage being image-wide).

The remainder of this chapter is organised as follows. Section 3.2 proposes a two-stage feature extraction model. Filtering-based, statistical, structural and model-based features are surveyed in Sections 3.3, 3.4, 3.5, and 3.6 respectively. The surveyed features are summarised in Section 3.7 and the implementation of these features is provided in Section 3.8. Finally, conclusions are drawn in Section 3.9.

3.2 A Two-Stage Feature Extraction Model

Texture classification and retrieval algorithms normally calculate a set of feature matrices from an image and then compute global statistics from each feature matrix. For instance, filtering-based features first conduct a filtering operation and then compute global statistics from each response matrix. The global statistics are finally combined into a feature vector. Similarly, the majority of the other types of features first extract 2nd-order or higher order features from local neighbourhoods, and then calculate global statistics from these local features. The extraction process of these features can thus be generalised into a two-stage model, i.e. "local-global model" (see Figure 3.1). To be exact, the computational feature sets that we identified in Section 2.2 fit this model. The model can also help us to understand these feature sets within a unified framework.

In the first stage, features are extracted from small local neighbourhoods or lines, in order to encode local patterns or other texture characteristics. Although using a larger neighbourhood can encode information concerning a larger spatial extent, computational complexity is also often increased. In addition, it can also produce an "averaging effect" which decreases the discriminatory power of features [Mao and Jain, 1992]. Hence, most features only utilise small neighbourhoods in the first stage. In the second stage,

global features are computed from the local feature outputs using 1st- or 2nd-order statistics. Some feature sets compute intermediate features from the local features before the global features are extracted. In this case, the intermediate process is also merged into the second stage.

Figure 3.1: A two-stage feature extraction model. Here, rhombuses donate the input or output data and rectangles mean the data processing, i.e. feature extraction. It is noteworthy that the local feature extraction operation could be "null". In this case, the output local features are input grey level values. In addition, some intermediate processes can also be merged into the global feature extraction stage.

In the following four sections, we will review the 46 feature sets identified in Section 2.2 in terms of feature category, statistical properties in both stages and the spatial extent that is exploited by the feature extraction operation of the first stage. Since the spatial extent exploited in the second stage is considered as the whole image, we do not report this information for each feature set separately.

3.3 Filtering-Based Features

Linear filtering operations can be conducted and defined in both spatial and frequency domains. In the spatial domain, filtering-based features are obtained by convolving an image with a mask (filter). Alternatively, the image can be transformed into the frequency domain using the Fourier transform (FT) [Lizorkin, 2001], and this is followed by the multiplication of the frequency domain version of the linear filter.

The L-N-L model (linear-nonlinear-linear model, also termed as "F-R-F" (filter-rectify-filter) model or a "back pocket" model [Malik and Perona, 1990]) consists of a linear filtering process, a nonlinear rectification and another linear process. In general, linear filters are divided into "spatial domain defined filters" and "frequency domain defined filters" depending on which domain they are designed and implemented in. According to Parseval's theorem [Weisstein], the sum of the squares of a response image obtained in the spatial domain is equal to the mean of the squares of the magnitude in the frequency domain (see Figure 3.2). The theorem can be expressed as:

$$\sum_{x,y}(f'(x,y))^2 = \frac{1}{MN}\sum_{u,v}|H(u,v)\cdot F(u,v)|^2, \quad (3.1)$$

where $f'(x,y)$ is the filtered image obtained by applying a linear filter $h(x,y)$ on an $M\times N$ image $f(x,y)$, $H(u,v)$ and $F(u,v)$ are transform functions of the filter and the image in the frequency domain respectively, (x,y) is the coordinate of one pixel in the spatial domain and (u,v) is the corresponding coordinate in the frequency domain. The right side of Equation (3.1) is the mean of the squares of the magnitude of the complex filtered image. We therefore conclude that linear filtering-based features, except quadrature filters-based features which are designed to use local phase, only utilise the power spectrum and ignore the phase information, no matter whether filtering is performed in the spatial or frequency domain, due to Equation (3.1).

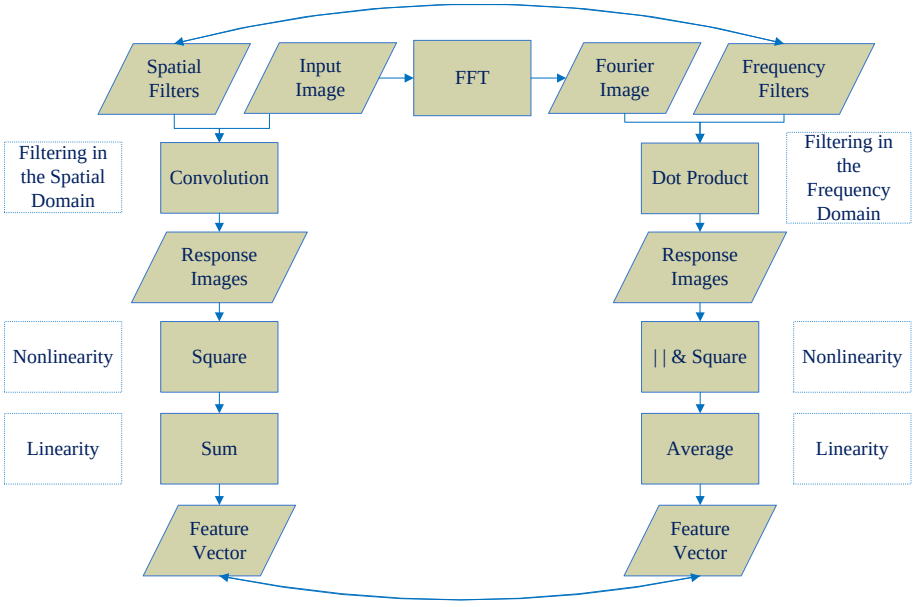


Figure 3.2: The Relationship between filtering operations in the spatial domain and frequency domain.

In this situation, filtering-based features can be divided into power spectra-based and phase spectra-based filtering features. For phase spectra-based filtering features, since the phase information extracted normally lies in the principal value range [Ying, 2006], phase unwrapping is used to recover original phase values. However, the use of such data is still an open problem [Ying, 2006]. Hence, in this research, we examine power spectra-based filtering features. However, we also tested one phase spectra-based filtering feature set: Joint Statistics of Complex Wavelet (JSCW) [Portilla and Simoncelli, 2000] for comparison.

3.3.1 Spatial Domain Defined Filtering Features

With respect to the two-stage model, the filtering operation is regarded as the first stage. Response matrices are considered as higher order statistics as the positional information in the matrix (image) is implicitly retained. The spatial extent exploited by spatial domain defined filtering featues in this stage is the size of the masks or filters involved. The post-processing of the response matrices is taken as the second stage. However, for those features designed for texture segmentation, the high dimensional output of this stage is not applicable for image-based similarity measurement (or estimation).

Discrete Cosine Transform Based Channel Filters (DCT)

Ng et al. [1992] interpreted the local linear transform, e.g. the discrete cosine transform (DCT), as a multichannel spatial filtering approach. Although this feature set is inspired by the DCT which has many similarities with the FFT, it is implemented in the spatial domain. Nine 3×3 filter masks (see Figure 3.3 (a)) are obtained from three 1D DCT basis vectors:

$$\mu_1 = \{1, 1, 1\}^T, \mu_2 = \{1, 0, -1\}^T, \text{and } \mu_3 = \{1, -2, 1\}^T. \quad (3.2)$$

After the filtering operation is performed by convolving the filter masks with the image, local variance matrices are computed from response matrices based on 15×15 local windows and are utilised as feature maps.

(a) (b)

Figure 3.3: (a): Nine 3×3 DCT masks; and (b) nine 3×3 eigen masks obtained from a texture image.

Stage 1 The filtering operation is conducted in the first stage. Response matrices are higher order statistics and the spatial extent used for each operation is 3×3 pixels.

Stage 2 The computation of local variance feature maps is performed in the second stage. However, these feature maps are not directly applicable for image-based similarity measurement.

Eigen Filters

Given that $f(x,y)$ is an image and $E[\cdot]$ is the expectation function, a 9×9 matrix

$$R_{xx} = \begin{bmatrix} E[f(x,y)f(x,y)] & \dots & E[f(x,y)f(x,y+2)] & \dots & E[f(x,y)f(x+2,y+2)] \\ \vdots & & \vdots & & \vdots \\ E[f(x,y+2)f(x,y)] & \dots & E[f(x,y+2)f(x,y+2)] & \dots & E[f(x,y+2)f(x+2,y+2)] \\ \vdots & & \vdots & & \vdots \\ E[f(x+2,y+2)f(x,y)] & \cdots & E[f(x+2,y+2)f(x,y+2)] & \dots & E[f(x+2,y+2)f(x+2,y+2)] \\ \vdots & & \vdots & & \vdots \end{bmatrix} \quad (3.3)$$

is estimated, and eigenvectors and eigenvalues are computed for each texture image [Ade, 1983]. Each 9×1 eigenvector is considered as a 3×3 eigen mask (see Figure 3.3 (b), termed as "EIGENFILTER"). The mean of the absolute values of the differences between pixel values and the local mean within a 15×15 window are also calculated for each position in a response image. The means computed from all response matrices are employed as the features of the pixel at the corresponding position.

Stage 1 The filtering operation is performed in the first stage. Response matrices are higher order statistics and the spatial extent utilised for each operation is 3×3 pixels.

Stage 2 The computation of feature maps is conducted in the second stage. However, these feature maps are not applicable for image-based similarity measurement.

Gabor Energy Filters

Fogel and Sagi [1989] represented a texture image by computing the Gabor power spectrum of micro-patterns. The Gabor function (termed as "GABORENERGY") is defined as:

$$h(x,y|W,\theta,\varphi,X,Y) = \exp^{\frac{-[(x-X)^2+(y-Y)^2]}{2\sigma^2}} \times \sin(W(x\cos\theta - y\sin\theta) + \varphi), \quad (3.4)$$

where σ is the Gaussian width, θ is the filter orientation, W is its frequency, φ is its phase angle, and (X,Y) is the centre of the filter. Given that $f(x,y)$ represents an input image and $h(x,y)$ stands for a Gabor filter, then $h * f$ ($*$ means the convolution operation) can encode spectra for different orientations and shifts. The sum of the squares of two response matrices is computed for each pixel. Thus, only power information is

used. The feature map is utilised for the representation of one micropattern image or montaged image.

Stage 1 The filtering operation is performed in the first stage. Response matrices are higher order statistics and the spatial extent utilised for each operation is 17×17 pixels.

Stage 2 The computation of the feature map is conducted in the second stage. However, the feature map is not applicable for image-based similarity measurement.

Laws Masks

Laws [1980] convolved an image with a set of 2D masks (referred to as "LAWS") to extract texture features. In total, 25 2D masks can be obtained by convolving a vertical 1D mask with a horizontal 1D mask, given five 1D masks: L5 = [1 4 6 4 1], E5 = [-1 -2 0 2 1], S5 = [-1 0 2 0 -1], W5 = [-1 2 0 -2 1] and R5 = [1 -4 6 -4 1]. The mean of the absolute values of responses or the square root of the sums of the squares of responses within a 15×15 windows, i.e. "texture energy measure", is computed for each position in one response image. These "energy measure" maps are finally used as features instead of the response images.

Stage 1 The filtering operation is conducted in the first stage. Response matrices are higher order statistics and the spatial extent utilised for each operation is 5×5 pixels.

Stage 2 The computation of "energy measures" feature maps is performed in the second stage. However, these feature maps are not applicable for image-based similarity measurement.

Localised Gabor Filters

Considering different textures possess distinct dominant characterising frequencies, Bovik et al. [1990] introduced a type of complex 2D Gabor functions (referred to as "GABORBOVIK") which is expressed as

$$h(x,y) = g(x',y') \cdot \exp[2\pi j(Xx + Yy)], \tag{3.5}$$

where $(x',y') = (x\cos\varphi + y\sin\varphi, -x\sin\varphi + y\cos\varphi)$, $j = \sqrt{-1}$, (X,Y) is the central frequency which is chosen from the frequency at which one spectral peak of a texture occurs, and

$$g(x,y) = (\frac{1}{2\pi\lambda\sigma^2}) \cdot \exp[-\frac{(x/\lambda)^2+y^2}{2\sigma^2}]. \quad (3.6)$$

Consequently, $h(x,y)$ can be taken as a complex sinusoidal grating modulated by a 2D Gaussian envelope with an aspect ratio λ, scale parameter σ, and the major axis orients at an angle φ from the *x*-axis. A post-processing, including a nonlinear process and a linear process, is also applied on each response matrix in sequence.

Stage 1 The filtering operation is conducted in the first stage. Response matrices are higher order statistics and the spatial extent utilised for each filtering operation is 85×3, 43×3, 21×3, 11×3 and 11×3 pixels at five different resolutions respectively.

Stage 2 The post-processing is performed in the second stage. However, the feature maps are not applicable for image-based similarity measurement.

Dyadic Gabor Filter Bank

Jain et al. [1991] developed a multi-channel filtering scheme using real-valued and even-symmetric Gabor filters (see Figure 3.4). The impulse response of an even-symmetric Gabor filter (termed as "GABORJFSD") is expressed as

$$h(x,y) = e^{\left\{-\frac{1}{2}\left[\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right]\right\}\cos(2\pi\mu_0 x)}, \quad (3.7)$$

where μ_0 stands for the frequency of a sinusoidal grating along the *x*-axis, and σ_x and σ_y are constants of a Gaussian envelope along *x* and *y* axes respectively. A nonlinear process is also performed on each response matrix.

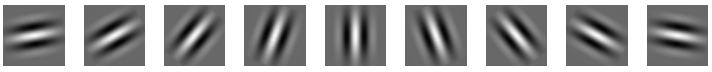


Figure 3.4: Even-symmetric Gabor spatial filters at nine different orientations. For display purposes, each filter is padded into a square matrix.

Stage 1 The filtering operation is performed in the first stage. Response matrices are higher order statistics and the spatial extent utilised for each operation is 409×329, 205×165, 103×83, 51×41 and 27×21 pixels at five different resolutions respectively.

Stage 2 The nonlinear process is conducted in the second stage. However, the feature maps are not applicable for image-based similarity measurement.

Leung-Malik Filter Bank

A hybrid filter bank (termed as "LM", see Figure 3.5), including 36 Gaussian derivative filters (1st- and 2nd-order derivatives at six orientations and three scales), eight Laplacian of Gaussian filters and four Gaussian low pass filters, was utilised by Leung and Malik [2001]. Given that $h(x, y)$ is a Gaussian function, 1st-order Gaussian derivative filters are defined as

$$h'_x(x, y) = -\frac{x}{\sigma^2} h(x, y) \text{ and } h'_y(x, y) = -\frac{y}{\sigma^2} h(x, y). \tag{3.8}$$

where σ is the scale (standard deviation).

Stage 1 The filtering operation is conducted in the first stage. Response matrices are higher order statistics and the spatial extent utilised for each operation is 49×49 pixels.

Stage 2 In the original publication, response matrices were used to extract textons and accumulate texton histograms. Since we only use the response matrices, there is no operation in the second stage. In addition, the response matrices are not applicable for image-based similarity measurement.

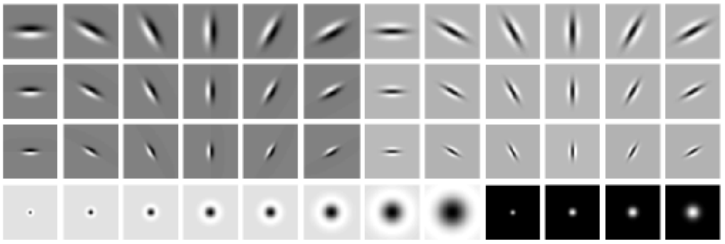


Figure 3.5: LM (spatial) filter bank [Varma and Zisserman, 2005].

Schmid Filter Bank

Schmid [2001] utilised a bank of 13 rotation-invariant isotropic "Gabor-like" filters (see Figure 3.6, termed as "S") to obtain grey level descriptors. These filters are defined as

$$h(x, y, \tau, \sigma) = h_0(\tau, \sigma) + \cos\left(\frac{\sqrt{x^2+y^2}\pi\tau}{\sigma}\right) e^{-\frac{x^2+y^2}{2\sigma^2}}, \tag{3.9}$$

where τ is the number of cycles of the harmonic function enclosed by the Gaussian envelope of a filter, σ is the scale (standard deviation) and $h_0(\tau, \sigma)$ is added to the function to obtain a zero DC (Direct Current) component.

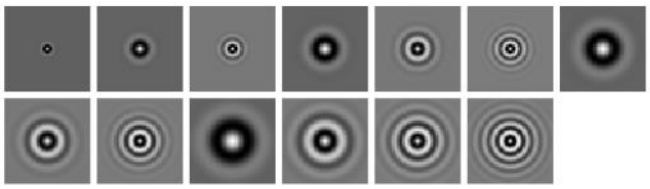


Figure 3.6: 13 isotropic (spatial) filters used by the Schmid filter bank [Varma and Zisserman, 2005].

Stage 1 The filtering operation is performed in the first stage. Response matrices are higher order statistics and the spatial extent utilised for each operation is 49×49 pixels.

Stage 2 In the original publication, the response matrices were used to cluster centroids. Since we only use the response matrices, the second stage is null. Besides, the response matrices are not applicable for image-based similarity measurement.

Root Filter Set and Maximum Response Set

Varma and Zisserman [2005] constructed a hybrid filter bank (Root Filter Set, i.e. RFS, see Figure 3.7) which involves 36 Gaussian derivative filters (see Equation (3.8)), one Gaussian low pass filter and one Laplacian of Gaussian filter. Furthermore, filter responses obtained at different orientations but the same scale are "collapsed" and only the maximum filter response over all orientations at each scale is kept, in order to achieve approximate rotation invariance. Finally, only six maximum filter responses and two isotropic filter responses, namely, maximum response set (MR8), are used for each pixel. Motivated by Weber's law, the filter response at each pixel (x, y) is normalised as

$$f(x,y) = \frac{f(x,y)\log\left(1+\frac{L(x,y)}{0.003}\right)}{L(x,y)}, \tag{3.10}$$

where $L(x,y) = \|f(x,y)\|_2$ is the magnitude of the filter response vector at that pixel.

Stage 1 The filtering operation is performed in the first stage. Response matrices are higher order statistics and the spatial extent utilised for each operation is 49×49 pixels.

Stage 2 In the original publication, normalised response matrices were used to extract textons and texton histograms. Since we only use the normalised response matrices, the normalisation operation is regarded as the second stage. However, the normalised response matrices are not applicable for image-based similarity measurement.

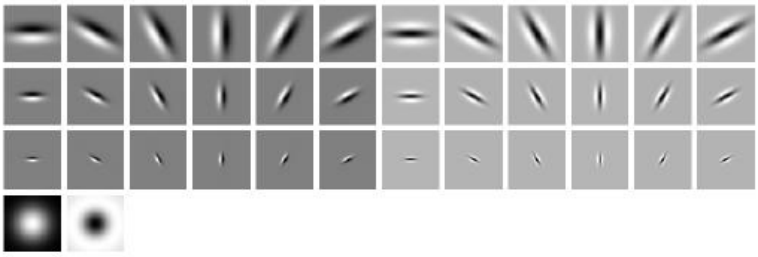


Figure 3.7: Root Filter Set (spatial filters) [Varma and Zisserman, 2005].

3.3.2 Frequency Domain Defined Filtering Features

According to the two-stage model, filtering operation is considered as the first stage. Response matrices are regarded as higher order statistics. Given a finite image, the actual spatial extent utilised by the Fourier transform in the first stage is the whole image. The global statistic extraction from the response matrices is taken as the second stage. Although the filters described in this subsection can also be implemented in the spatial domain, we only referred to their original definitions in the frequency domain for simplicity.

Gabor Wavelet Filter Bank

Manjunathi and Ma [1996] defined a 2D Gabor function $H(u,v)$ (referred to as "GABORMM") in the frequency domain as:

$$H(u,v) = \exp\left\{-\frac{1}{2}\left[\frac{(u-W)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2}\right]\right\}, \tag{3.11}$$

where (u,v) is the corresponding coordinate in the frequency domain. σ_u and σ_v are computed as

$$\sigma_u = \frac{(a-1)U_h}{(a+1)\sqrt{2\ln 2}}, \tag{3.12}$$

$$\sigma_v = \tan\left(\frac{\pi}{2K}\right)\left[U_h - 2\ln\left(\frac{2\sigma_u^2}{U_h}\right)\right]\left[2\ln 2 - \frac{(2\ln 2)^2\sigma_u^2}{U_h^2}\right]^{-\frac{1}{2}}, \tag{3.13}$$

where $a = (U_h/U_l)^{\frac{1}{S-1}}$, U_l and U_h stand for the lower and upper central frequencies of interest, K is the number of orientations, S is the number of scales and $W = U_h$. Only the magnitudes of responses are used. The mean and standard deviation are computed for each magnitude matrix and are concatenated into a feature vector.

Stage 1 The filtering operation is conducted in the first stage. Magnitude matrices are higher order statistics and the spatial extent used for the operation is the whole image.

Stage 2 The computation of means and standard deviations is performed in the second stage. Both mean and standard deviation are 1st-order statistics.

Ring and Wedge Filters

Given that textures can be discriminated by spatial frequency and orientation, Coggins and Jain [1985] proposed seven dyadically spaced ring filters and four wedge-shaped orientation filters (referred to as "RING & WEDGE"). In polar coordinate system, the Ring filters (see Figure 3.8 (a)-(g)) are defined as

$$P(r) = 2\sum_{\theta=0}^{\pi} P(r,\theta), \tag{3.14}$$

where $r = \sqrt{u^2 + v^2}$ stands for the radius, $\theta = \text{atan2}(v/u)$ denotes the angle, and (u, v) is the coordinate in the frequency domain. Meantime, the wedge filters (see Figure 3.8 (h)-(k)) are expressed as

$$P(\theta) = \sum_{r=0}^{\infty} P(r,\theta). \tag{3.15}$$

The average local energy features are computed from 11 response matrices based on their grey level histograms and are used as the representation of the input texture.

(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k)

Figure 3.8: Power responses of ring ((a)-(g)) and wedge ((h)-(k)) filters in the frequency domain.

Stage 1 The filtering operation is performed in the first stage. Response matrices are higher order statistics and the spatial extent used for the operation is the whole image.

Stage 2 The computation of average local energy features for texture classification is conducted in the second stage. These features are 1st-order statistics.

Joint Statistics of Complex Wavelet (JSCW)

Portilla and Simoncelli [2000] first built a steerable pyramid from one texture image based on complex "analytic" filters, whose real and imaginary parts correspond to a pair

of even- and odd-symmetric filters. In this way, local phase and magnitude are exploited. The steerable pyramid was obtained by recursively splitting an image into a series of oriented subbands and a lowpass residual band. The filters utilised are polar-separable in the frequency domain and are defined as:

$$L(r,\theta)=\begin{cases}2\cos\left(\frac{\pi}{2}\log_2\left(\frac{4r}{\pi}\right)\right), & \frac{\pi}{4}<r<\frac{\pi}{2}\\ 2, & r\le\frac{\pi}{4}\\ 0, & r\ge\frac{\pi}{2}\end{cases}, \tag{3.15}$$

$$B_k(r,\theta)=H(r)G_k(\theta). \tag{3.16}$$

The radial and angular parts are expressed as: $H(r)=\begin{cases}\cos\left(\frac{\pi}{2}\log_2\left(\frac{2r}{\pi}\right)\right), & \frac{\pi}{4}<r<\frac{\pi}{2}\\ 1, & r\ge\frac{\pi}{2}\\ 0, & r\le\frac{\pi}{4}\end{cases}$

and $G_k(\theta)=\begin{cases}\alpha_K\left[\cos\left(\theta-\frac{\pi k}{K}\right)\right]^{K-1}, & \left|\theta-\frac{\pi k}{K}\right|<\frac{\pi}{2}\\ 0, & otherwise\end{cases}$, where r and θ are polar coordinates, $\alpha_K=2^{k-1}\frac{(K-1)!}{\sqrt{K[2(K-1)]!}}$, $k\in[0,K-1]$, and K is the number of orientation bands.

A set of statistics, including marginal statistics, raw coefficient correlation, coefficient magnitude statistics and cross-scale phase statistics, are extracted from the original texture image or the phase and magnitude spectra of its pyramid images. All statistics are combined into a feature vector.

Stage 1 The construction of the steerable pyramid is conducted in the first stage. Pyramid images are higher order statistics and the spatial extent utilised is the whole image.

Stage 2 The computation of the statistics introduced above is performed in the second stage. These features are 1st- or 2nd-order statistics.

3.3.3 Summary of Filtering-Based Features

Regarding the two-stage model, the filtering operation is taken as the first stage. For filters defined in the spatial domain the spatial extent exploited by the first stage is simply that of the associated convolution masks. For frequency domain defined filters, the whole image is regarded as the maximal spatial extent used in the first stage because

the Fourier transform is applied to the whole image. The filtering response matrices are considered as higher order statistics.

The post-processing of the response matrices is considered as the second stage. The spatial domain defined filtering features examined in this section were originally utilised for texture segmentation. However, their feature matrices cannot be directly used for texture classification or image-based similarity measurement due to their high dimensionalities. In contrast, the frequency domain defined filtering features compute global 1st- or 2nd-order statistics directly. Hence, they cannot encode aperiodic long-range interactions.

However, the phase spectra-based filtering feature set, i.e. joint statistics of complex wavelet (JSCW) [Portilla and Simoncelli, 2000], captures local phase information in the first stage. Global statistics are then computed from the local phase as well as the original image in the second stage. Although local phase and grey level image data are considered as higher order statistics, the global statistics are only 1st- or 2nd-order statistics. Hence, the higher order aperiodic spatial relationship between different pixels is lost. As a result, JSCW cannot encode aperiodic long-range interactions.

3.4 Statistical Features

Statistical features normally compute the statistical distribution of image grey levels at specified relative pixel positions. Statistics are first computed based on individual pixels, pixel pairs or pixel groups in the spatial domain. Thus, statistics are divided into 1st-order, 2nd-order or higher order statistics respectively. Global statistics are then extracted from these local statistics in order to obtain global measures of one texture.

3.4.1 Review of Statistical Features

Autocorrelation Function Based Features (ACF)

Autocorrelation function analysis [Fujii et al., 2003] models three perceptual texture properties: contrast, coarseness and regularity. The autocorrelation function is computed from an $M \times N$ texture image $f(x, y)$ with a mean of zero as:

$$\Phi(\Delta x, \Delta y) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) f(x+\Delta x, y+\Delta y)/MN, \quad (3.17)$$

where $f(x+\Delta x, y+\Delta y)$ is the corresponding shifted image and $(\Delta x, \Delta y)$ is a displacement. Generally, the normalisation of the autocorrelation function is computed as:

$$\phi(\Delta x, \Delta y) = \Phi(\Delta x, \Delta y) / \left(\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)^2 / MN \right). \quad (3.18)$$

The denominator in Equation (3.18) is the maximum value of the autocorrelation function. The autocorrelation at the displacement $\Delta x = \Delta y = 0$ is utilised to represent the perceived contrast. Coarseness and regularity are represented by the displacement of the maximum peak in the autocorrelation function excluding the origin and the amplitude of the maximum peak respectively. When random textures are concerned, the estimated autocorrelation function are not periodic and the decay rate of the autocorrelation function is used to measure the coarseness and regularity.

Stage 1 The computation of the autocorrelation function is regarded as the first stage. The autocorrelation function is a 2nd-order statistic. The maximal spatial extent can be taken as the whole image because the maximal displacement used in this stage is equal to $(M-1)$ or $(N-1)$.

Stage 2 The estimation of the three properties is conducted in the second stage. These properties are 2nd-order statistics.

Covariance Matrix Based Features (CVM)

Covariance matrix based features introduced by Liu and Madiraju [1996] extract eigen features from a "variant" local covariance matrix which is defined as

$$C = \frac{1}{N} \sum_{i=1}^{N} (w_i - w_m)(w_i - w_m)^T, \quad (3.19)$$

$$w_m = \frac{1}{N} \sum_{i=1}^{N} w_i, \text{ and } w_i = [x_i, y_i, z_i]^T, \quad (3.20)$$

where $N = n^2$ is the size of the $n \times n$ $(n = 3)$ local neighbourhood, x_i, y_i are coordinates in row and column directions respectively and z_i is the grey level value of the *i*-th pixel. Three eigenvalue matrices of the covariance matrix are then computed. Six regional descriptor matrices are extracted from the first two eigenvalue matrices for pixel-based texture classification according to three moment-based statistics, namely, mean, variance and symmetry, based on a larger region up to 81×81 pixels.

Stage 1 The computation of the covariance matrix is performed in the first stage. Different from the ordinary covariance matrix, the covariance matrix computed here takes the coordinates of pixels into consideration and is considered as a higher order statistic. The maximal spatial extent utilised in this stage is 3×3 pixels.

Stage 2 The rest operations are comprised of the second stage. However, the feature matrices are not applicable for image-based similarity measurement.

Grey Level Histogram Features (GLH)

A histogram is a very effective statistical tool for grey level images. Global statistics, such as maximum, minimum, mean, variance, skewness, kurtosis and other statistics, can be directly extracted from a grey level histogram as features [Mirmehdi et al., 2009].

Stage 1 The first stage does not conduct any operation and the output is taken as the grey level image itself. The maximal spatial extent utilised in this stage is thus regarded as the whole image. The grey level image can be considered as a higher order statistic.

Stage 2 The accumulation of the histogram from one grey level image and the computation of global statistics are comprised of the second stage. Since the histogram is a 1st-order statistic, the statistics extracted from it are 1st-order statistics as well.

Grey Level Sum and Difference Histograms

The histogram of absolute differences (GLADH) [Weszka et al., 1976] first computes absolute grey level difference as

$$GLAD(x,y) = |f(x,y) - f(x+\Delta x, y+\Delta y)|, \tag{3.21}$$

where $f(x,y)$ and $f(x+\Delta x, y+\Delta y)$ are grey levels of the current and the displaced pixels respectively. One histogram is directly extracted from the grey level differences computed at each combination of the direction and distance. A set of statistics are then computed from each histogram. The mean and standard deviation of each statistic over different directions at each distance are finally combined as a feature vector. Unser [1986] also utilised histograms of signed grey level differences (GLSDH), grey level sums (GLSH) and their combination (GLSDSH).

Stage 1 The computation of sums and/or differences is taken as the first stage. Pairwise grey level sums/differences are 2nd-order statistics. Considering they are computed based on a displacement (≤ 8), the maximal spatial extent exploited is regarded as 1×9 (or 9×1) pixels.

Stage 2 The accumulation of histograms and the following computation from these are comprised of the second stage. The histograms, means, and standard deviations are 1st-order statistics.

Grey Level Co-occurrence Matrices

The grey level co-occurrence matrix (GLCM) [Haralick et al., 1973] was designed to encode the spatial grey level dependence relationship between a pixel and its neighbouring pixels. The original image is first quantised to *G* grey levels equiprobably. Then, the co-occurrence frequency of grey level pairs at certain relative displacement is accumulated into a co-occurrence matrix. In total, 14 statistics or their subset are computed from each matrix. The mean and standard deviation of each statistic over different directions are computed at each distance. All means and standard deviations are finally combined into a feature vector.

In addition, it is noteworthy that a co-occurrence matrix contains dipoles. Dipole histogram was believed to uniquely determine one finite image [Chubb and Yellott, 2000]. However, one or even all co-occurrence matrices obtained here cannot represent all dipoles because only a limited number of displacements are used.

Stage 1 The computation of the co-occurrence of two pixels and the accumulation of co-occurrence matrices are considered as the first stage. The co-occurrence matrices are regarded as 2nd-order statistics. Since the pairwise co-occurrence is extracted based on a displacement (≤ 8), the maximal spatial extent exploited is regarded as 1×9 (or 9×1) pixels.

Stage 2 The following computation from the occurrence matrices are comprised of the second stage. The mean and standard deviation computed from the statistics that are calculated from each co-occurrence matrix are 1st-order statistics.

Grey Level Run Length Matrices

The grey level run length matrix (GLRLM) records the length of some collinearly adjacent pixels with the same grey level [Galloway, 1975]. One GLRLM is obtained as

$$GLRLM(g,l|\theta) = \text{numel}\{(x,y)|f(x,y)=g, f(x+p,y+q)\neq g, f(x+\mu,y+v)=g, \mu<p \,\&\, v<q\}, \tag{3.22}$$

where g means the grey level, l denotes the run length, θ is one direction of 0°, 45°, 90° and 135°, $f(x,y)$ is the grey level at (x,y), "numel" is a function to count the number of the elements of its input, $p=(l+1)\cos\theta$, $q=(l+1)\sin\theta$, $0\leq g\leq G-1$ (G stands for the number of grey levels), $0\leq l\leq L$ (L is the longest run length), and $0^{\circ}\leq\theta\leq 180^{\circ}$. A set of statistics are then computed from the matrix at each direction. The mean and standard deviation of each statistic over four directions are combined into a feature vector.

Stage 1 The first stage locates runs. The runs are higher order statistics. Since the run length is varying, the maximal spatial extent utilised is regarded as the whole image (the possible longest length).

Stage 2 The second stage accumulates run length matrices and extracts features from these. The run length matrices are 1st-order statistics. Thus, the statistics calculated from these and the means and standard deviations computed in this stage are 1st-order statistics as well.

Grey Level Gap Length Matrices

The grey level gap length matrix (GLGLM) [Wang et al., 1994] is obtained from the distribution of grey level gap lengths for each grey level in an image. A GLGLM is computed as:

$$GLGLM(g,l|\theta) = \text{numel}\{(x,y)|f(x,y)=f(x+p,y+q)=g, f(x+\mu,y+v)\neq g, \mu<p \,\&\, v<q\}, \tag{3.23}$$

where all variables are the same as those in Equation (3.22) except that l denotes the gap length. Similarly, a series of statistics are computed from the gap length matrix at each direction. The mean and standard deviation of each statistic over different directions are finally utilised as features.

Stage 1 The first stage locates gaps. The gaps are higher order statistics. Since the gap length is varying, the maximal spatial extent utilised is regarded as the whole image (the possible longest length).

Stage 2 The second stage accumulates gap length matrices and extracts features from these. Since gap length matrices are 1st-order statistics, the statistics computed from these and the means and standard deviations calculated are 1st-order statistics.

Local Centre-Symmetric Covariance Based Features

Harwood et al. [1995] introduced four sets of local centre-symmetric covariance based texture features, including two different local centre-symmetric auto-correlations with linear and rank-order versions (SAC and SRAC), a related covariance measure (SCOV) and a variance ratio (SVR). The four local statistics are defined as:

$$SCOV = \frac{1}{4}\sum_i^4 (g_i - \mu)(g_i{}' - \mu), \tag{3.24}$$

$$SAC = \frac{SCOV}{\frac{1}{8}\sum_i^4 \left(g_i{}^2 + g_i{}'^2\right) - \mu^2}, \tag{3.25}$$

$$SRAC = 1 - \frac{12\{\sum_i^4 (r_i - r_i')^2 + T_x\}}{m^3 - m},\ T_x = \frac{1}{12}\sum_i^l (t_i^3 - t_i), \tag{3.26}$$

$$SVR = \frac{\sum_i^4 (g_i - g_i{}')^2}{\sum_i^4 (g_i + g_i{}')^2 - \mu^2}, \tag{3.27}$$

where g_i and $g_i{}'$ are centre-symmetric pairs of pixels in a 3×3 neighbourhood (see Figure 3.9), μ is the mean in the neighbourhood, m is equal to 3^2, r_i means the rank of the grey level of pixel i in the ranked 3×3 neighbourhood, t_i is the number of ties at rank r_i, l denotes the number of all ranks. A histogram is obtained from each set of statistics.

g_2	g_3	g_4
g_1		g_1'
g_4'	g_3'	g_2'

Figure 3.9: A 3×3 neighbourhood with four centre-symmetric pairs of pixels.

Stage 1 The computation of SAC, SRAC, SCOV and SVR is considered as the first stage. The statistics calculated in this stage are 2nd-order statistics. The spatial extent utilised in this stage is only 3×3 pixels.

Stage 2 The generation of the histogram is regarded as the second stage. The histogram obtained in this stage is a 1st-order statistic.

Multi-scale Autoconvolution

Multi-scale autoconvolution (MSA) is an affine invariant image transform based on the probabilistic interpretation of one image [Rahtu et al., 2005]. In order to enhance the computational speed, the discrete form of MSA can be computed as

$$F(\partial,\beta) = \frac{1}{MN}\frac{1}{\hat{f}(0)^3}\sum_{i=0}^{MN-1}\hat{f}(-w_i)\cdot\hat{f}(\partial w_i)\cdot\hat{f}(\beta w_i)\cdot\hat{f}(\gamma w_i), \tag{3.31}$$

where $\hat{f}$ is the discrete Fourier transform of an image $f(x,y)$, w_i are points in the Fourier domain, and $\gamma = 1-\alpha-\beta$. The averages of the real parts of a set of $F(\partial,\beta)$ are combined into a feature vector.

Stage 1 The computation of three dot products (·) can be regarded as the first stage. The real part of the $F(\partial,\beta)$ utilised in this stage is considered as a higher order statistic. Similar to frequency domain defined filtering features, the maximal spatial extent exploited here is regarded as the whole image.

Stage 2 The average operation of the $F(\partial,\beta)$ is considered as the second stage. The mean calculated in this stage is a 1st-order statistic.

Surrounding Region Dependence Method (SRDM)

Kim et al. [1999] obtained a surrounding region dependence matrix as

$$M(q) = [\partial(i,j)], 0 \le i \le m, 0 \le j \le n, \tag{3.28}$$

where q is a given threshold, and m and n denote the total numbers of pixels in the surrounding region R_1 and R_2 (see Figure 3.10) respectively. $\partial(i,j)$ is expressed as

$$\partial(i,j) = \text{numel}\{(x,y)|C_{R_1}(x,y) = i \;\&\&\; C_{R_2}(x,y) = j, (x,y) \in M\times N\}, \tag{3.29}$$

$$C_{R_t}(x,y) = \text{numel}\{(k,l)|[f(x,y)-f(k,l)] > q, (k,l) \in R_t, t = 1,2\}, \tag{3.30}$$

where $\text{numel}(S)$ counts the number of elements in the set S, $M\times N$ is an image and $f(x,y)$ is the grey level value at the position (x,y). Finally, four weighted-sum statistics are computed from $M(q)$ to represent an image.

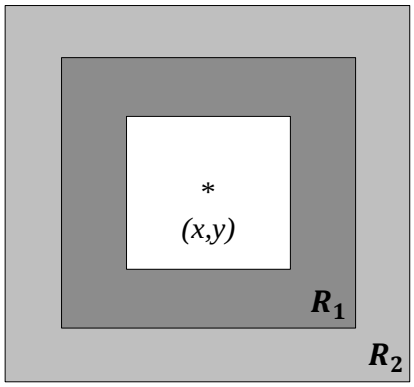


Figure 3.10: Two surrounding regions R_1 and R_2 at the current position (x, y).

Stage 1 The extraction of $\partial(i,j)$ and the accumulation of $M(q)$ are performed in the first stage. In essence, $\partial(i,j)$ is the co-occurrence of i and j counted in two surrounding regions and is a 2nd-order statistic. Hence, $M(q)$ is also a 2nd-order statistic. The maximal spatial extent exploited in this stage is 7×7 pixels, i.e. the size of the outer surrounding region.

Stage 2 The computation of four statistics is conducted in the second stage. The four statistics are 1st-order statistics.

The Trace Transform

The Trace transform (TT) introduced by Kadyrov et al. [2001, 2002] is a generalisation of the Radon transform [Toft, 1996]. Given that a tracing line t is drawn at changing values of ϕ and p (see Figure 3.11), where ϕ ranges from 0 to 2π and p lies in the range of $[-p_{max}, p_{max}]$ with p_{max} is no more than an half of the diagonal length of one input image. Trace functional T is first applied along the tracing line t. Functional P is then applied to the 2D Trace transform function T and a 1D function of ϕ is obtained. Finally, a third functional Φ along this 1D function generates a scalar value which is used as an image feature. Given that different functionals can be chosen for T, P and Φ, the features generated are expressed as

$$f^{ijk} = \Phi^i\left(P^j\left(T^k\left(S(C;\,\phi,p,t)\right)\right)\right). \tag{3.32}$$

Stage 1 The computation of T is regarded as the first stage. Since T is computed based on (straight) trace lines, given an $M \times N$ image, the maximal length of the trace lines, i.e. $\sqrt{M^2+N^2} \times 1$ pixels, is the maximal spatial extent used in the first stage. The T functionals include 1st-order and 2nd-order statistics.

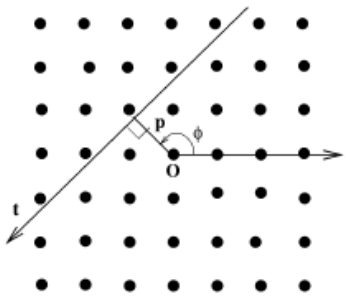


Figure 3.11: Introduction to the parameters of an image tracing line [Kadyrov 02].

Stage 2 The computation of P and Φ is considered as the second stage. Although the P and Φ functionals consist of 1st-order, 2nd-order and higher order statistics, the statistics computed from the 1st-order and 2nd-order T function values are also 1st-order or 2nd-order statistics.

3.4.2 Summary of Statistical Features

All of the statistical features described above, except for covariance matrix based features (CVM), fit into the two-stage model. However, even CVM features can fit into the model if we append a global feature extraction operation.

MSA [Rahtu et al., 2005] calculates higher order statistics using the FFT in the first stage while it computes 1st-order statistics in the second stage. Hence, the spatial relationship between pixels is lost. Other feature sets extract local 2nd-order or higher order features in the first stage based on small ($\leq 7\times7$, see Table 3.1) neighbourhoods except GLH [Mirmehdi et al., 2009], GLGLM [Wang et al., 1994], GLRLM [Galloway, 1975] and TT [Kadyrov and Petrou, 2001]. Then, 1st-order statistics are computed from these local features or the intermediate features obtained from these. In this case, the spatial relationship between neighbourhoods is again lost.

GLH accumulates a histogram from a grey level image. Since the histogram is a 1st-order statistic, the statistics computed from it cannot encode the spatial relationship between pixels. GLGLM, GLRLM and TT extract features based on runs, gaps and trace lines in one grey level image in the first stage. The runs or gaps obtained using GLGLM or GLRLM in this stage are considered as higher order statistics. However, the mean and standard deviation extracted using GLGLM or GLRLM in the second stage are only 1st-order statistics. Thus, the spatial relationship between the runs or gaps is also not encoded. Regarding TT, 1st- and 2nd-order statistics are extracted in the first stage and

1st-order or 2nd-order statistics are obtained in the second stage. Thus, the aperiodic spatial relationships between different trace lines are not encoded.

As discussed in Section 2.6.3, higher order statistics need to be extracted from a long-range spatial extent in order to encode aperiodic (and periodic) spatial relationship between local regions (pixels, lines, or neighourhoods), i.e. aperiodic (and periodic) long-range interactions. However, the output of the second stage is normally "orderless" because those feature sets generally use only 1st-order global features.

3.5 Structural Features

Structural texture analysis generally assumes that textures are comprised of primitives or elements [Haralick, 1979] [Vilnrotter et al., 1986]. Originally, Julesz [1981] employed "textons" to describe basic texture elements. Furthermore, the concept of textons was also applied to filters [Leung and Malik, 2001] [Zhu et al., 2005], image patches [Varma and Zisserman, 2009], gradient information [Ojala et al., 1996], local binary patterns (LBP) [Ojala et al., 2002], local derivatives [Zhang et al., 2010] and local phase information [Ojansivu et al., 2008]. Generally speaking, popular texton-based features first extract local features and then utilise vector quantisation techniques to map these local features into a texton space. Each pixel in one texture is assigned the label of the texton which lies closest in the local feature space. Finally, one histogram is accumulated in the texton space to describe the distribution of textons.

3.5.1 Review of Structural Features

Gradient-Based Feature Distributions

Ojala et al. [1996] compared the joint distribution histogram of the gradient magnitudes and directions computed using the Sobel operators [Sobel, 1990] from a texture image, namely, GMAG/GDIR (termed as "GMAGGDIRSOBEL" in this thesis), with other feature sets. Local derivatives are computed firstly. Gradient magnitudes and directions are then calculated from these data. Finally, a joint distribution histogram is extracted from the gradient magnitudes and directions.

Stage 1 The computation of gradient magnitudes and directions is regarded as the first stage. Both gradient magnitude and direction matrices are higher order statistics. The maximal spatial extents exploited by the Sobel operators are 3×3 pixels.

Stage 2 The histogram accumulation is considered as the second stage. The joint distribution histogram extracted in this stage is a 2nd-order statistic.

Local Binary Patterns (LBP)

Wang and He [1990] originally introduced the concept of "texture unit" and used the co-occurrence of the distribution of texture units computed in neighbourhoods as the texture spectrum. Ojala et al. [1996] further proposed its two-stage version (referred to as "LBPBASIC"), i.e. local binary patterns (LBP). In nature, it uses a mask-based filtering scheme firstly and then generates a histogram by thresholding response matrices.

LBP with a circular neighbourhood is defined as:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p, \tag{3.33}$$

where g_c corresponds to the grey value of the central pixel in the neighbourhood and $g_p (p = 0, 1, \ldots P-1)$ stands for the grey values of P equally spaced pixels on a circle of the radius R ($R > 0$) . In addition, $s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$ is applied. Furthermore, the idea of "uniform" was suggested [Ojala et al., 2002b] and the grey-scale and rotation invariant description $LBP_{P,R}^{riu2}$ ("LBPRIU2") was proposed as

$$LBP_{P,R}^{riu2} = \begin{cases} \sum_{p=0}^{P-1} s(g_p - g_c), & if\ U(LBP_{P,R}) \leq 2 \\ P+1, & otherwise \end{cases}, \tag{3.34}$$

where $U(LBP_{P,R}) = |s(g_{P-1} - g_c) - s(g_0 - g_c)| + \sum_{p=1}^{P-1} |s(g_p - g_c) - s(g_{p-1} - g_c)|$.

However, $LBP_{P,R}^{riu2}$ does not exploit other local texture characteristics, e.g. contrast. The performance of $LBP_{P,R}^{riu2}$ can be further enhanced via combining it with one rotation invariant variance measure $VAR_{P,R}$ ("VAR", see Equation 3.35). Hence, the joint distribution, i.e. $LBP_{P,R}^{riu2}/VAR_{P,R}$ ("LBPRIU2 & VAR"), was introduced. In addition, multiresolution $LBP_{P,R}^{riu2}$, $VAR_{P,R}$ and $LBP_{P,R}^{riu2}/VAR_{P,R}$ were also proposed.

$$VAR_{P,R} = \frac{1}{P}\sum_{p=0}^{P-1}(g_p - \mu)^2, \text{ where } \mu = \frac{1}{P}\sum_{p=0}^{P-1} g_p \tag{3.35}$$

Recently, Ahonen et al. [2009] extracted another set of LBP-based features, i.e. LBPHF, using discrete Fourier transform (DFT). The LBPHF features are invariant to the rotation of one image. Besides, Ahonen and Pietikäinen [2009] also developed a local derivative filters based LBP, i.e. LBPDF.

Stage 1 The computation of various local binary patterns is conducted in the first stage. The response matrices obtained using LBPDF, the LBP maps derived using LBPBASIC, LBPRIU2 and LBPHF, and the local variance matrix calculated using VAR are considered as higher order statistics. The maximal spatial extents employed by LBPBASIC, LBPDF, LBPRIU2, LBPHF and VAR in this stage are 3×3, 3×3, 5×5, 5×5 and 5×5 (circular neighbourhood with a radius of 2) pixels, in sequence.

Stage 2 The accumulation of histograms is regarded as the second stage. The 1D histogram and 2D joint distribution histogram are 1st- and 2nd-order statistics respectively.

Varma and Zisserman Textons (VZ-Textons)

Varma and Zisserman [2005] improved the 3D texton-based features proposed by Leung and Malik [2001]. The similar filter bank (see Figure 3.7) was used but only the maximal of the responses obtained using Gaussian derivative filters at each direction and the responses of two isotropic filters, i.e. maximal response sets (MR8), were used. Then *K*-means was applied on a number of images of each texture. The centroids obtained from the images of each texture were concatenated into a global textons dictionary which is different from the local textons dictionaries constructed by Leung and Malik [2001]. However, histograms are obtained in the similar way for both approaches. The texton-based method proposed by Varma and Zisserman [2005] is well-known as "VZ-MR8". Furthermore, when image patches were used to extract textons instead of filter responses, three sets of features referred to as "VZ-NEIGHBOURHOOD", "VZ-JOINT" and "VZ-MRF" were introduced [Varma and Zisserman, 2009] respectively.

Stage 1 The extraction of filtering responses or local image exemplars is performed in the first stage. Both the filtering responses and image exemplars can be regarded as higher order statistics. The maximal spatial extents exploited by VZ-MR8, VZ-NEIGHBOURHOOD, VZ-JOINT and VZ-MRF are 49×49, 19×19, 19×19 and 19×19 pixels, respectively.

Stage 2 The accumulation of texton histograms is conducted in the second stage. The 1D histogram used by VZ-MR8, VZ-NEIGHBOURHOOD and VZ-JOINT is a 1st-

order statistic. Regarding the 2D joint distribution histogram utilised by VZ-MRF, it is the co-occurrence matrix of the central pixel and texton labels of neighbouring pixels in each local neighbourhood in essence. As a result, it is a 2nd-order statistic.

Local Phase Quantisation

In the original local phase quantisation (LPQ) [Ojansivu and Heikkila, 2008], the phase is considered in neighbourhoods centred at (x, y) of one image $f(x, y)$. By using a short-term Fourier transform, local phase spectra $F(u, x, y)$ are obtained (u denotes the frequency). Furthermore, local Fourier coefficients are calculated at four frequency values: u_1, u_2, u_3, and u_4, and a vector

$$F(x, y) = [F(u_1, x, y), F(u_2, x, y), F(u_3, x, y), F(u_4, x, y)] \tag{3.36}$$

is obtained at each pixel position. The phase information in the Fourier coefficients is derived via $q_i = \begin{cases} 1, & if\ g_i \geq 0 \\ 0, & otherwise \end{cases}$ where g_i is the i-th component of $G(x, y) = [Re\{F(x, y)\}, Im\{F(x, y)\}]$. Finally, eight binary coefficients q_i are converted into integer values in the range of [0, 255] by the quantisation $f_{LPQ}(x, y) = \sum_{i=1}^{8} q_i 2^{i-1}$ and a 256-bin histogram is accumulated from these values. In addition, a rotation invariant local phase quantisation (RI-LPQ) method [Ojansivu et al., 2008] is also developed in order to obtain rotation invariance.

Stage 1 The local phase is computed in the first stage. The phase information is a higher order statistic. The maximal spatial extent used in this stage is 9×9 pixels.

Stage 2 The generation of the histogram is regarded as the second stage. The histogram is a 1st-order statistic.

Local Derivative Patterns

The *n*th-order local derivative pattern (LDP) [Zhang et al., 2010] is used to encode gradient changes in local neighbourhoods of $I_\partial^{n-1}(Z)$ and is defined as

$$LDP_\partial^n(Z_0) = \left\{ f\left(I_\partial^{n-1}(Z_0), I_\partial^{n-1}(Z_1)\right), f\left(I_\partial^{n-1}(Z_0), I_\partial^{n-1}(Z_2)\right), \ldots, f\left(I_\partial^{n-1}(Z_0), I_\partial^{n-1}(Z_8)\right) \right\}, \tag{3.37}$$

where $I_\partial^{n-1}(Z_0)$ is the (n-1)th-order derivative images at the direction of ∂ (0°, 45°, 90° and 135°) with $Z = Z_0$. Meantime, $f\left(I_\partial^{n-1}(Z_0), I_\partial^{n-1}(Z_i)\right) = \begin{cases} 0, & if\ I_\partial^{n-1}(Z_i) \,.\, I_\partial^{n-1}(Z_0) > 0 \\ 1, & if\ I_\partial^{n-1}(Z_i) \,.\, I_\partial^{n-1}(Z_0) \leq 0 \end{cases}, i =$

$_{1,2,\ldots 8}$ represents the (n-1)th-order gradient transitions with binary patterns. The nth-order LDP is defined as $LDP^n(Z) = \{LDP^n_\partial(\mathrm{Z}) | \partial = 0°, 45°, 90° \text{ and } 135°\}$. At each direction ∂, $LDP^n_\partial(\mathrm{Z})$ is encoded into an 8-bit binary string at each pixel position and then a 256-bin histogram is generated. Finally, four histograms are concatenated into one feature histogram. Furthermore, in order to capture the spatial pattern in larger spatial extent, the input image is first divided into a series of sub-regions. An LDP histogram is first extracted from each sub-region at each direction independently. Four histograms are concatenated into one histogram for each sub-region. All concatenated histograms are then combined into a spatially enhanced histogram (referred to as "LDPSE").

Stage 1 The extract of LDP maps is conducted in the first stage. The LDP maps are higher order statistics. The maximal spatial extent exploited in this stage is 3×3 pixels.

Stage 2 The second stage accumulates a histogram at each direction and concatenated these into one histogram. The histogram obtained in this stage is a 1st-order statistic. In addition, LDPSE is the concatenation of multiple 1st-order histograms. As a result, it is still a 1st-order statistic.

3.5.2 Summary of Structural Features

To summarise, the structural features examined in this section generally obtain statistics or filtering responses based on small (≤ 49×49 pixels) local neighbourhoods and then conduct vector quantisation in order to obtain feature histograms. The local features obtained in the first stage are in general higher order statistics. However, these structural features do not use the local statistics directly. Hence, they only tackle locally orderless images [Koenderink and Van Doorn, 1999]. The histograms accumulated in the second stage are only 1st- or 2nd-order statistics and are used as image descriptors. As a result, the aperiodic spatial relationship between local neighbourhoods is lost. That is to say, the aperiodic global topology (or spatial distribution) of local texture patches is discarded. Therefore, such structural features cannot encode aperiodic long-range interactions.

3.6 Model-Based Features

Several texture models such as fractal models [Mandelbrot, 1982] [Pentland, 1984] [Chaudhuri et al., 1993], Markov random field models [Chellappa and Chatterjee, 1985]

and simultaneous autoregressive models [Mao and Jain, 1992] have been introduced in the past decades.

3.6.1 Review of Model-Based Features

Fractal Dimension Models

The box-counting based fractal dimension (FD) models (termed as "FRACTAL-DIMENSION") were proposed by Chaudhuri et al. [1993]. Given a bounded set A in n-D Euclidean space, it will be self-similar if A is the union of N_r non-overlapping fragments of itself and each fragment is similar with A scaled down by a ratio r. The fractal dimension FD of A can be computed as

$$1 = N_r r^{FD} \text{ or } FD = \frac{\log(N_r)}{\log(1/r)}. \tag{3.38}$$

If an $M \times M$ image has been down-sampled to an $s \times s$ image where $M/2 \geq s > 1$, then $r = s/M$ approximately. Furthermore, if the image is regarded as a 3D space in which (x, y) represents a 2D position and the third coordinate (z) stands for the grey level, the (x, y) space is then partitioned into a series of $s \times s$ grids. In addition, there is a column of $s \times s \times s$ boxes which are labeled as 1, 2,… in each grid. Given that k and l denote the numbers of the maximum and minimum grey levels which fall into the box of the image in the (i, j)-th grid respectively, $n_r(i, j) = l - k + 1$ means the contribution of N in the (i, j)-th grid. Accumulating contributions over all grids, then $N_r = \sum_{i,j} n_r(i, j)$ is computed for different values of r. Three sets of FD of the original image I_1, the high grey-valued image I_2, and low grey-valued image I_3, and a set of multi-fractal FD can be calculated based on overlapping windows. Finally, means and variances are computed in local 7×7 windows of the four FD matrices as texture features.

Stage 1 The extraction of four sets of FD is considered as the first step. The FD is taken as a 2nd-order statistic. The maximal spatial extent utilised in this stage is the size of the overlapping windows, i.e. 17×17 pixels.

Stage 2 The computation of local means and variances is regarded as the second step. Means and variances are 1st-order statistics. However, the feature matrices are not applicable for image-based similarity measurement.

Gaussian Markov Random Field Models

Chellappa and Chatterjee [1985] proposed two sets of texture features on the basis of the assumption that textures are Gaussian and fit Gaussian Markov random field (GMRF) models. The first set of features was obtained from the least squares (LS) estimates of the parameters of the models. On the other hand, the sample correlations over a specific window were believed to be sufficient statistics for the parameters of the models, in the case that the texture examined is really generated by a Gaussian MRF model. Consequently, the sample correlation vector was utilised as a lossless feature set.

Stage 1 The estimation of GMRF models is performed in the first stage. The model coefficients are higher order statistics. The maximal spatial extent used in this stage is the size of the mask used for estimating GMRF models, i.e. 5×3 pixels.

Stage 2 The computation of the variances of model coefficients is performed in the second stage. The variances computed in this stage are 1st-order statistics.

Multi-resolution Simultaneous Autoregressive Models

The multi-resolution simultaneous autoregressive (MRSAR) model regards one texture as a non-causal Markov random field [Picard et al., 1993]. It can be estimated in different pyramid levels [Mao and Jain, 1992] or different levels of local neighbourhoods [Picard et al., 1993]. When the latter is applied, given that symmetric neighbourhood is used, i.e. c_i has the same value for $f(x+\Delta x_k, y+\Delta y_k)$ and $f(x-\Delta x_k, y-\Delta y_k)$, $f(x,y)$ is estimated by the combination of its neighbouring pixels as

$$f(x,y) = \mu + \sum_{i=1}^{4} c_i \times (f(x+\Delta x_i, y+\Delta y_i) + f(x-\Delta x_i, y-\Delta y_i)) + \varepsilon, \qquad (3.39)$$

where μ is the bias, c_i are four coefficients of the potential model, $(\Delta x_i, \Delta y_i)$ is one of {(-*l*, 0), (0, *l*), (-*l*, *l*), (*l*, *l*)}, l is the level of the neighbourhood, and ε is the estimation error.

However, the solution to Equation (3.39) might be underdetermined if the neighbouring pixels are considered alone. Thus, a larger $n \times n$ moving window is used on the image for the estimation. First of all, leaving boundary pixels out, one $(n-2l) \times 8$ matrix X which contains eight neighbours of each valid pixel and one $(n-2l) \times 1$ vector Y which consists of all valid pixels in the window are constructed, respectively. Secondly, the least-squares (LS) estimation is applied on X and Y. Four coefficients and the stand-

ard deviation of the estimation error are used to represent the central pixel of the window. For texture classification, the mean and covariance are computed from the feature matrix estimated at each neighbourhood level and are combined into a feature vector.

Stage 1 The estimation of SAR models is taken as the first stage. The SAR coefficients and estimation errors are higher order statistics. The size of the moving-window, i.e. 25×25 pixels, is the maximal spatial extent exploited in this stage.

Stage 2 The computation of the mean and covariance is conducted in the second stage. The mean and covariance are 1st- and 2nd-order statistics respectively.

3.6.2 Summary of Model-Based Features

The three model-based feature sets introduced above extract 2nd-order or higher order statistics based on small neighbourhoods in the first stage. However, only 1st- or 2nd-order statistics are computed using GMRF and MRSAR in the second stage. As a result, the aperiodic spatial relationship between local neighbourhoods is lost in this stage. These two feature sets can, hence, only encode 2nd-order or higher order statistics in a small spatial extent. In this situation, they cannot capture aperiodic long-range interactions. In addition, the fractal dimension (FD) model features were designed for the task of texture segmentation and cannot be directly used for image-based texture similarity measurement due to the high dimensionalities of their output feature matrices. Thus, a global feature extraction operation is required after the original algorithm is conducted.

3.7 Summary of Surveyed Feature Sets

Table 3.1 summarises the 46 feature sets in terms of the feature category [Mirmehdi et al., 2009], the tasks that these feature sets were used for, the feature's statistical properties (in two stages of feature extraction) and the maximal spatial extent exploited by one "primitive" operation (e.g. a computation in a neighbourhood) in the first stage at five different resolutions. It can be observed that the majority of these feature sets fit the two-stage model. However, the feature sets which were originally employed for texture segmentation are not suitable for use for image-based texture similarity estimation directly. In Table 3.1, the statistical properties of these feature sets in the second stage of feature extraction are marked as "N/A".

Identifier	Categories	Tasks	Feature Orders		Maximal Spatial Extent Used In Stage I				
			Stage I	Stage II	1024	512	256	128	64
ACF	♠	R	2nd	2nd	*				
CVM	♠	S	Higher	N/A	3×3				
DCT	♦	S	Higher	N/A	3×3				
EIGENFILTER	♦	S	Higher	N/A	3×3				
FRACTALDIMENSION	♣	S	2nd	N/A	17×17				
GABORBOVIK	♦	S	Higher	N/A	85×3	43×3	21×3	11×3	11×3
GABORENERGY	♦	S	Higher	N/A	17×17				
GABORJFFD	♦	S	Higher	N/A	*				
GABORJFSD	♦	S	Higher	N/A	409×329	205×165	103×83	51×41	27×21
GABORMM	♦	R	Higher	1st	*				
GLADH	♠	C	2nd	1st	1×9 or 9×1				
GLCM	♠	C	2nd	1st	1×9 or 9×1				
GLGLM	♠	PC	Higher	1st	*				
GLH	♠	PC	Higher	1st	*				
GLRLM	♠	C	Higher	1st	*				
GLSDH	♠	C	2nd	1st	1×9 or 9×1				
GLSDSH	♠	C	2nd	1st	1×9 or 9×1				
GLSH	♠	C	2nd	1st	1×9 or 9×1				
GMAGGDIRSOBEL	♥♦	C	Higher	2nd	3×3				
GMRF	♣	C	Higher	1st	5×3				
JSCW	♦♠	PC	Higher	1st&2nd	*				
LAWS	♦	S	Higher	N/A	5×5				
LBPBASIC	♥♠	C	Higher	1st	3×3				
LBPDF	♥♠	C	Higher	1st	3×3				
LBPHF	♥♠	C	Higher	1st	5×5 (Radius = 2)				
LBPRIU2	♥♠	C	Higher	1st	5×5 (Radius = 2)				
LBPRIU2 & VAR	♥♠	C	Higher	1st	5×5 (Radius = 2)				
LDP	♥♠	C	Higher	1st	3×3				
LDPSE	♥♠	C	Higher	1st	3×3				
LM	♦	PS	Higher	N/A	49×49				
MR8	♦	PS	Higher	N/A	49×49				
MRSAR	♣	S&C	Higher	1st&2nd	25×25				
MSA	♠	C	Higher	1st	*				
RFS	♦	PS	Higher	N/A	49×49				
RI-LPQ	♥♠	C	Higher	1st	9×9				
RING & WEDGE	♦	S&C	Higher	1st	*				
S	♦	PS	Higher	N/A	49×49				
SAC	♠	C	2nd	1st	3×3				
SRAC	♠	C	2nd	1st	3×3				
SRDM	♠	C	2nd	1st	7×7				
SVR	♠	C	2nd	1st	3×3				
TT	♠	C	1st&2nd	1st&2nd	$\sqrt{M^2+N^2}\times 1$ (For an $M\times N$ image)				
VAR	♠	C	Higher	1st	5×5 (Radius = 2)				
VZ-MR8	♥	C	Higher	1st	49×49				
VZ-MRF	♥	C	Higher	2nd	19×19				
VZ-NEIGHBOURHOOD	♥	C	Higher	1st	19×19				

(1) ♦, ♠, ♥ and ♣: filtering-based, statistical, structural and model-based features

(2) "S", "C" and "R": segmentation (including pixel-based classification), classification and retrieval (including ranking) tasks

(3) "PS" and "PC": can potentially be used for segmentation and classification tasks

(4) 1024, 512, 256, 128 and 64: five different resolutions, i.e. 1024×1024, 512×512, 256×256, 128×128 and 64×64

(5) *: the feature set works in the whole image

(6) N/A: the feature set was originally designed for segmentation (including pixel-based classification)

Table 3.1: Summary of 46 feature sets according to their original definations.

3.8 Implementation

In this section, we describe any modifications that were required to be made to the feature sets listed in Table 3.1 so that they could be used for texture similarity estimation as required for the research described in this thesis.

Generally speaking, experimental conditions should be kept as consistent as possible for different computational feature sets in order to obtain reliable and impartial evaluation results. In addition, the optimal working conditions of these feature sets, as described in their original publications, should be retained where possible. In our study, we used the published implementation of each feature set as far as was practicable. If the source code has been published along with a publication, it was utilised in our evaluation experiments. Otherwise, we used the implementation by others or implemented it by ourselves according to the original publication.

Regarding the parameters used for each feature set, we referred to the (optimal) conditions used in its original publication or the publications which use the feature set. Tuning the parameters of one feature set or feature selection is avoided as much as possible in order to produce unprejudiced evaluation results.

As we pointed out, feature sets originally designed for texture segmentation (see column "Tasks" in Table 3.1) are pixel-based and are unsuitable for use for image-based similarity measurement (or estimation) directly. In order to utilise these feature sets in our study, a global feature extraction stage was appended to the original implementation.

3.8.1 Revised Features

Filtering-Based Features

Since phase unwrapping is still an open problem [Ying, 2006], we only used the power spectra of response matrices obtained using Localised Gabor filters (GABORBOVIK) [Bovik et al., 1990] although phase spectra were also available. For all spatial domain defined filtering features, Gabor wavelet filter bank (GABORMM) [Manjunathi and Ma, 1996], and ring and wedge filters (RING & WEDGE) [Coggins and Jain, 1985], the original post-processing on response matrices is discarded. The square operation is first applied to each response matrix, and then the mean is computed from each squared re-

sponse matrix. All means are finally concatenated into a feature vector. These processes guarantee that the premise of Parseval's theorem [Weisstein] is satisfied.

Thus, the original first stages were kept intact. The response matrices obtained in this stage are considered as higher order statistics. In the first stage, local neighbourhoods are normally used by the spatial domain defined filters. Hence, the size of these filters is the maximal spatial extent (see Table 3.1 or 3.2) exploited in this stage. However, for frequency domain defined filters, the whole image is taken as the maximal spatial extent.

The appended stage, i.e. the computation of global statistics from response matrcies, is considered as the second stage. Since the premise of Parseval's theorem [Weisstein] is satisfied, all these features only utilise power spectra. Due to the fact that the power spectrum cannot retain aperiodic image structure [Oppenheim and Lim, 1991], as discussed in Section 2.6.3, these power spectra based filtering features cannot capture aperiodic long-range interactions even though some of these conduct filtering operation in a large spatial extent (see Table 3.1 or 3.2).

Covariance Matrix Based Features

Regarding covariance matrix based features (CVM) [Liu and Madiraju, 1996], the original first stage was kept intact. In the second stage, the mean and standard deviation are computed from each regional descriptor matrix and all means and standard deviations are combined into a feature vector. Since both of mean and standard deviation are 1st-order statistics, CVM cannot capture both periodic and aperiodic long-range interactions.

Gradient-Based Feature Distributions

For gradient-based feature distributions [Ojala et al., 1996], we extracted the histogram of gradient magnitudes and directions separately and also computed their joint distribution histogram. Gradient magnitudes and directions were computed using the Canny [Canny, 1986] or Sobel [Sobel, 1990] edge detectors. Thus, except for GMAGGDIR-SOBEL, five other sets of gradient-based feature distributions were obtained and were referred to as "GMAGCANNY", "GDIRCANNY", "GMAGGDIRCANNY", "GMAG-SOBEL" and "GDIRSOBEL" for Canny magnitude; Canny direction; Canny magnitude and direction, Sobel magnitude and Sobel direction respectively.

The computation of gradient magnitudes and/or directions is performed in the first stage. The gradient magnitude and/or direction matrices obtained in this stage are higher order statistics. The maximal spatial extents exploited by the Canny and Sobel operators in this stage are 9×9 pixels and 3×3 pixels respectively. The accumulation of histograms is considered as the second stage. The 1D histograms and 2D (joint) histograms are 1st- and 2nd-order statistics respectively.

Fractal Dimension Models

We used the version implemented by Smith and Burns [1997]. In the second stage, four variances are computed from the four sets of FD as texture features.

Multi-resolution Simultaneous Autoregressive (MRSAR) Models

We used the MRSAR algorithm implemented by Kwitt, R. [2009]. Considering the computational complexity, 3×3, 5×5 and 7×7 neighbourhoods ($l = 1, 2, 3$) and a 19×19 moving window were utilzied. In addition, a window shift of four pixels was used in order to enhance computational speed. In the original implementation, means and covariances were computed from model coefficient matrices. As Picard et al. [1993] mentioned, the *Mahalanobis* distance performs significantly better than the *Euclidean* distance on the covariances. In this study, however, we intend to employ the latter for all non-histogram based features (see Section 4.3.1) in order to compare these features using the same distance measure. Therefore, the mean and standard deviation were computed from the coefficient matrix estimated at one neighbourhood level. All means and standard deviations were combined into the feature vector.

The original first stage of MRSAR was kept intact. In the second stage, means and standard deviations were utilised as features. Both of these are 1st-order statistics. Thus, MRSAR cannot capture both periodic and aperiodic long-range interactions.

3.8.2 Summary of Implementation

In total, 51 feature sets, including GMAGCANNY, GDIRCANNY, GMAGGDIRCANNY, GMAGSOBEL and GDIRSOBEL, will be further examined. Table 3.2 summarises these feature sets as adapted where necessary for the research reported in this thesis in terms of the same aspects as those shown in Table 3.1 for the original versions.

Identifier	Categories	Feature Orders		Maximal Spatial Extent Used In Stage I				
		Stage I	Stage II	1024	512	256	128	64
ACF	♠	2nd	2nd	*				
CVM	♠	Higher	1st	3×3				
DCT	♦	Higher	2nd	3×3				
EIGENFILTER	♦	Higher	2nd	3×3				
FRACTALDIMENSION	♣	2nd	1st	17×17				
GABORBOVIK	♦	Higher	2nd	85×3	43×3	21×3	11×3	11×3
GABORENERGY	♦	Higher	2nd	17×17				
GABORJFFD	♦	Higher	2nd	*				
GABORJFSD	♦	Higher	2nd	409×329	205×165	103×83	51×41	27×21
GABORMM	♦	Higher	2nd	*				
GDIRCANNY	♥♦	Higher	1st	9×9				
GDIRSOBEL	♥♦	Higher	1st	3×3				
GLADH	♠	2nd	1st	1×9 or 9×1				
GLCM	♠	2nd	1st	1×9 or 9×1				
GLGLM	♠	Higher	1st	*				
GLH	♠	Higher	1st	*				
GLRLM	♠	Higher	1st	*				
GLSDH	♠	2nd	1st	1×9 or 9×1				
GLSDSH	♠	2nd	1st	1×9 or 9×1				
GLSH	♠	2nd	1st	1×9 or 9×1				
GMAGCANNY	♥♦	Higher	1st	9×9				
GMAGGDIRCANNY	♥♦	Higher	2nd	9×9				
GMAGGDIRSOBEL	♥♦	Higher	2nd	3×3				
GMAGSOBEL	♥♦	Higher	1st	3×3				
GMRF	♣	Higher	1st	5×3				
JSCW	♦♠	Higher	1st&2nd	*				
LAWS	♦	Higher	2nd	5×5				
LBPBASIC	♥♠	Higher	1st	3×3				
LBPDF	♥♠	Higher	1st	3×3				
LBPHF	♥♠	Higher	1st	5×5 (Radius = 2)				
LBPRIU2	♥♠	Higher	1st	5×5 (Radius = 2)				
LBPRIU2 & VAR	♥♠	Higher	1st	5×5 (Radius = 2)				
LDP	♥♠	Higher	1st	3×3				
LDPSE	♥♠	Higher	1st	3×3				
LM	♦	Higher	2nd	49×49				
MR8	♦	Higher	2nd	49×49				
MRSAR	♣	Higher	1st	19×19				
MSA	♠	Higher	1st	*				
RFS	♦	Higher	2nd	49×49				
RI-LPQ	♥♠	Higher	1st	9×9				
RING & WEDGE	♦	Higher	2nd	*				
S	♦	Higher	2nd	49×49				
SAC	♠	2nd	1st	3×3				
SRAC	♠	2nd	1st	3×3				
SRDM	♠	2nd	1st	7×7				
SVR	♠	2nd	1st	3×3				
TT	♠	1st&2nd	1st&2nd	$\sqrt{M^2+N^2}\times 1$ (For an $M\times N$ image)				
VAR	♠	Higher	1st	5×5 (Radius = 2)				
VZ-MR8	♥	Higher	1st	49×49				
VZ-MRF	♥	Higher	2nd	19×19				
VZ-NEIGHBOURHOOD	♥	Higher	1st	19×19				

(1) 1024, 512, 256, 128 and 64: five different resolutions, i.e. 1024×1024, 512×512, 256×256, 128×128 and 64×64

(2) ♦, ♠, ♥ and ♣: filtering-based, statistical, structural and model-based features

(3) *: the feature set works in the whole image

Table 3.2: Summary of the 51 feature sets that will be further examined in this thesis. Italic and bold fonts mean revised feature sets.

3.9 Conclusions

In this chapter, we first proposed a two-stage feature extraction model and then, briefly, reviewed the 46 feature sets that we identified in Section 2.2. As 13 of the feature sets were originally designed (or can potentially be used) for texture segmentation, their second stage was replaced by a standard global feature extraction process. In addition, five other sets of gradient-based feature distributions were implemented.

Among the 51 feature sets, it was found that:

(1) the filtering-based features, excepting JSCW [Portilla and Simoncelli, 2000], only utilise power spectra. In contrast, JSCW first obtains local phase and then computes 1st- or 2nd-order statistics from the original image and the local phase; and

(2) the statistical, structural and model-based features, except ACF [Fujii et al., 2003], MSA [Rahtu et al., 2005], GLH [Mirmehdi et al., 2009], GLGLM [Wang et al., 1994], GLRLM [Galloway, 1975] and TT [Kadyrov and Petrou, 2001], compute 2nd-order or higher order statistics only on small ($\leq 19\times19$) local neighbourhoods. Regarding ACF, MSA, GLH, GLGLM and GLRLM, although they employ higher order statistics in the first stage, they only produce 1st- or 2nd-order statistics in the second stage.

As discussed in Section 2.6.3, periodic long-range interactions can be modelled using long-range 2nd-order or higher order statistics while aperiodic long-range interactions can only be encoded using long-range higher order statistics. Since filtering-based features, excluding JSCW, only utilise discrete power spectrum, they cannot be used to capture aperiodic long-range interactions. On the other hand, JSCW, statistical, structural and model-based features normally compute 1st- or 2nd-order statistics in the second stage from the 2nd-order or higher order local statistics computed in the first stage. In this case, the aperiodic spatial relationship between different local neighbourhoods, i.e. aperiodic long-range interactions, is lost.

Since the features examined in this chapter have been extensively researched and reported, they will form the focus of our evaluations in the next three chapters. However, it should be noted that while they can exploit both short-range interactions and even periodic long-range interactions, they cannot exploit aperiodic long-range interactions.

Chapter 4

Pair-of-Pairs Based Evaluation Framework

4.1 Introduction

Chapter 2 reviewed different methods of acquiring, and different forms of, human similarity judgements. The pair-of-pairs format was identified as being both useful and relatively cheap to collect. This chapter introduces three different methods of deriving this type of data: (1) directly obtained from pair-of-pairs experiments; (2) derived from free-grouping experiments; and (3) obtained from the Isomap analysis of free-grouping data. Two of these methods are used to provide the ground-truth in an evaluation of the computational features introduced in the previous chapter. This chapter describes the acquisition of these datasets and a new pair-of-pairs based evaluation framework. In this framework the computationally derived pair-of-pairs judgements are compared with human derived pair-of-pairs judgements[1]. Figure 4.1 illustrates the evaluation pipeline of the framework.

The remainder of this chapter is organised as follows. Section 4.2 describes the acquisition of human derived pair-of-pairs judgements using three different approaches. A methodology for deriving pair-of-pairs judgements using computational features is then introduced in Section 4.3, and Section 4.4 proposes one approach for comparing computationally derived and human derived pair-of-pairs judgements. The conclusions are finally drawn in Section 4.5.

[1] In this thesis, pair-of-pairs judgements could be directly derived using a pair-of-pairs experiment [Clarke et al., 2012], or generated from a computational/perceptual similarity matrix. For simplicity, we term both of these as "pair-of-pairs judgements".

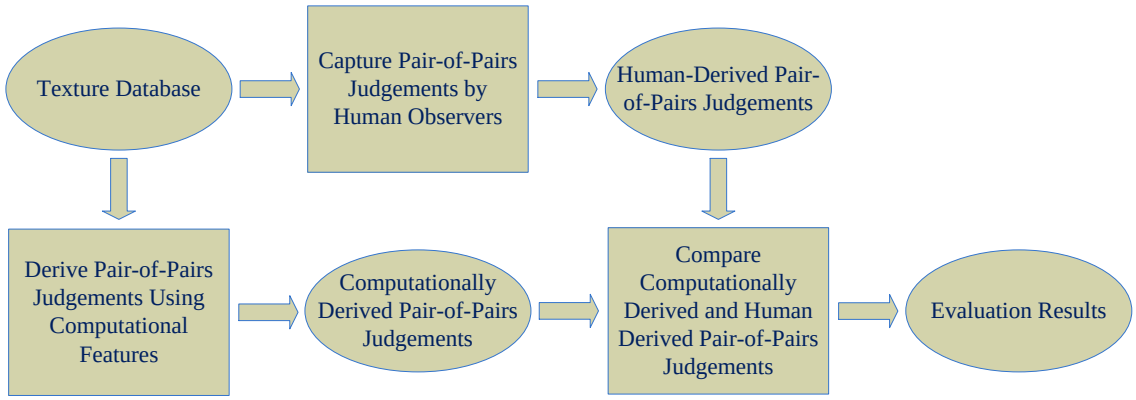


Figure 4.1: Flowchart of the pair-of-pairs evaluation framework.

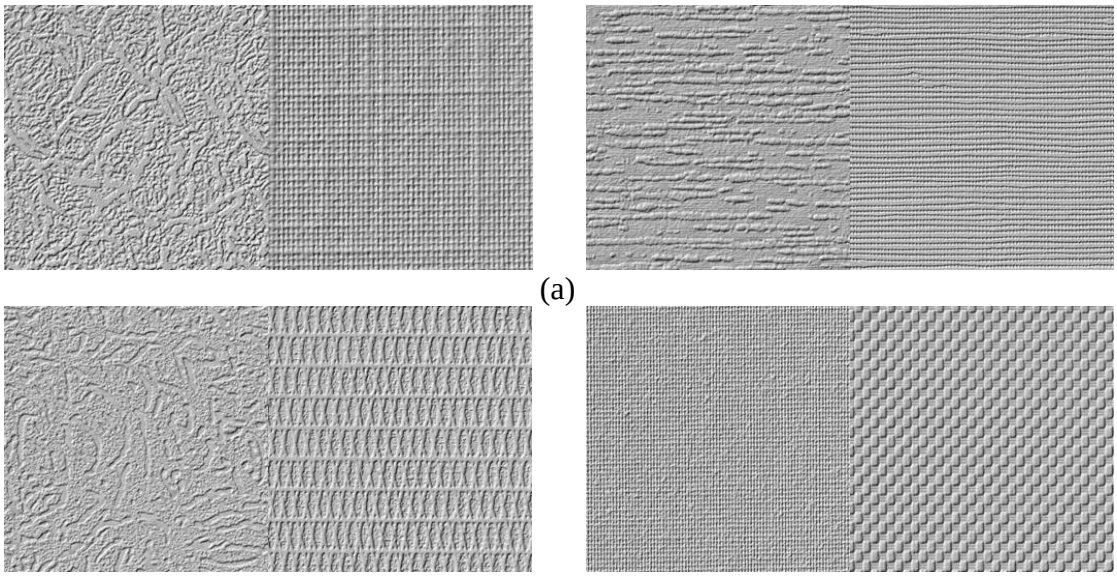
4.2 Human-Derived Pair-of-Pairs Judgements

As discussed in Section 2.1, texture segmentation, classification and retrieval normally use a set of class labels as their ground-truth data. In essence, the class labels provide only a Boolean-valued similarity matrix and do not generate higher resolution similarity data, such as rational-valued similarity and pair-of-pairs judgements. We obtained a set of perceptual pair-of-pairs judgments ($POPJ_{POP}$) directly using the pair-of-pairs method [Clarke et al., 2012]. Restricted by the time complexity of the experimental scheme, only 1000 pairs of pairs were used. However, pair-of-pairs judgements can also be generated from a similarity matrix. It is noteworthy that pair-of-pairs judgements generated from a Boolean-valued similarity matrix have a lower resolution than those constructed from a rational-valued similarity matrix. We therefore derived 1000 pair-of-pairs judgements ($POPJ_P$) from the rational-valued perceptual similarity matrix [Clarke et al., 2011] [Halley, 2011B] that was obtained from *Pertex* using free-grouping. In addition, due to the sparseness of this rational-valued similarity matrix, we used an 8D Isomap version (8D-ISO) obtained by applying Isomap analysis [Tenenbaum et al., 2000] to construct a third set of 1000 perceptual pair-of-pairs judgements ($POPJ_{ISO}$).

4.2.1 Direct Use of a Pair-of-Pairs Experiment ($POPJ_{POP}$)

We carried out a direct pair-of-pairs experiment [Clarke et al., 2012]. This differs from the standard pair-wise comparison tasks [David, 1988] where observers are required to judge whether two images are similar or not. The pair-of-pairs experiment involves two pairs of textures (see Figure 4.2) which are simultaneously displayed on the monitor in

each trial. During each trial, observers are required to judge which pair is more similar. Considering the heavy time cost of this experiment (around 2 hours for 1000 trials), only 1000 (out of all 4^{334} possible combinations) pairs of pairs $\{\{a, b\}, \{c, d\}\}$ were used. They were randomly selected from the 334 textures in the *Pertex* database. There were no other restrictions except that $a \neq b$ and $c \neq d$.



(a)

(b)

Figure 4.2: Two pairs of pairs of textures used in the pair-of-pairs experiment.

Once the pair-of-pairs experiment was completed, frequencies of the choices ("left" or "right") made by all 20 participants were accumulated and are labelled as FC_L and FC_R respectively. The judgement obtained in the i-th trial of the pair-of-pairs experiment, $J_{POP}(i)$, is computed from the normalised difference between the two figures:

$$J_{POP}(i) = \frac{FC_L(i) - FC_R(i)}{20}, i = 1, 2, \dots, 1000. \tag{4.1}$$

The pair-of-pairs judgement set directly obtained using the pair-of-pairs experiment, $POPJ_{POP}$, is then derived based on a set of $J_{POP}(i)$, as shown below:

$$POPJ_{POP}(i) = \begin{cases} 1, & J_{POP}(i) > 0 \\ 0, & J_{POP}(i) = 0 \\ -1, & J_{POP}(i) < 0 \end{cases}, i = 1, 2, \dots, 1000, \tag{4.2}$$

where "1" means that the left pair is more similar than the right one, "0" suggests that both pairs differ by the same level of similarity, and "-1" implies that the right pair is more similar than the left one.

4.2.2 Using a Free-Grouping Experiment ($POPJ_P$)

Pair-of-pairs judgements can also be generated from a similarity matrix obtained using a free-grouping experiment. Recently, Halley [2011A] derived a perceptual similarity matrix from a large (500) texture database using free-grouping. A subset (334), namely, *Pertex*, of this database and a subset of the perceptual similarity matrix was further obtained [Clarke et al., 2011] [Halley, 2011B]. As discussed in Section 2.1.2, free-grouping becomes time-consuming when the number of textures involved exceeds 200, and hence only 30 participants carried out this experiment. Figure 4.3 (a) plots the original similarity matrix in which the brightness at each point (i, j) denotes the magnitude ($\in$ [0, 1]) of the estimated similarity between textures i and j. In other words, the brighter a point is, the more similar the two textures are.

In order to provide an insight into the organisation of the similarities of the 334 textures, we clustered the original perceptual similarity matrix using hierarchical clustering analysis [Fraley and Raferty, 1998]. As shown in Figure 4.3 (b), it can be seen that most non-zero values are distributed close to the diagonal.

Only 1000 pair-of-pairs were examined in the original pair-of-pairs experiment. By using the *Pertex* perceptual similarity matrix, however, all pair-of-pairs can be examined and labelled with one of two perceptual similarity values: PS_L and PS_R. The judgement corresponding to the i-th trial of the pair-of-pairs experiment, i.e. $J_P(i)$, is computed based on the difference between these values:

$$J_P(i) = PS_L(i) - PS_R(i), i = 1, 2, \dots, 1000. \tag{4.3}$$

The pair-of-pairs judgement set: $POPJ_P$ obtained from the original perceptual similarity matrix is obtained based on a set of $J_P(i)$, $i = 1, 2, \dots, 1000$ as follows:

$$POPJ_P(i) = \begin{cases} 1, & J_P(i) > 0 \\ 0, & J_P(i) = 0 \\ -1, & J_P(i) < 0 \end{cases}, i = 1, 2, \dots, 1000, \tag{4.4}$$

where as before, "1" means that the left pair is more similar than the right one, "0" suggests that both pairs differ by the same level of similarity, and "-1" implies that the right pair is more similar than the left one.

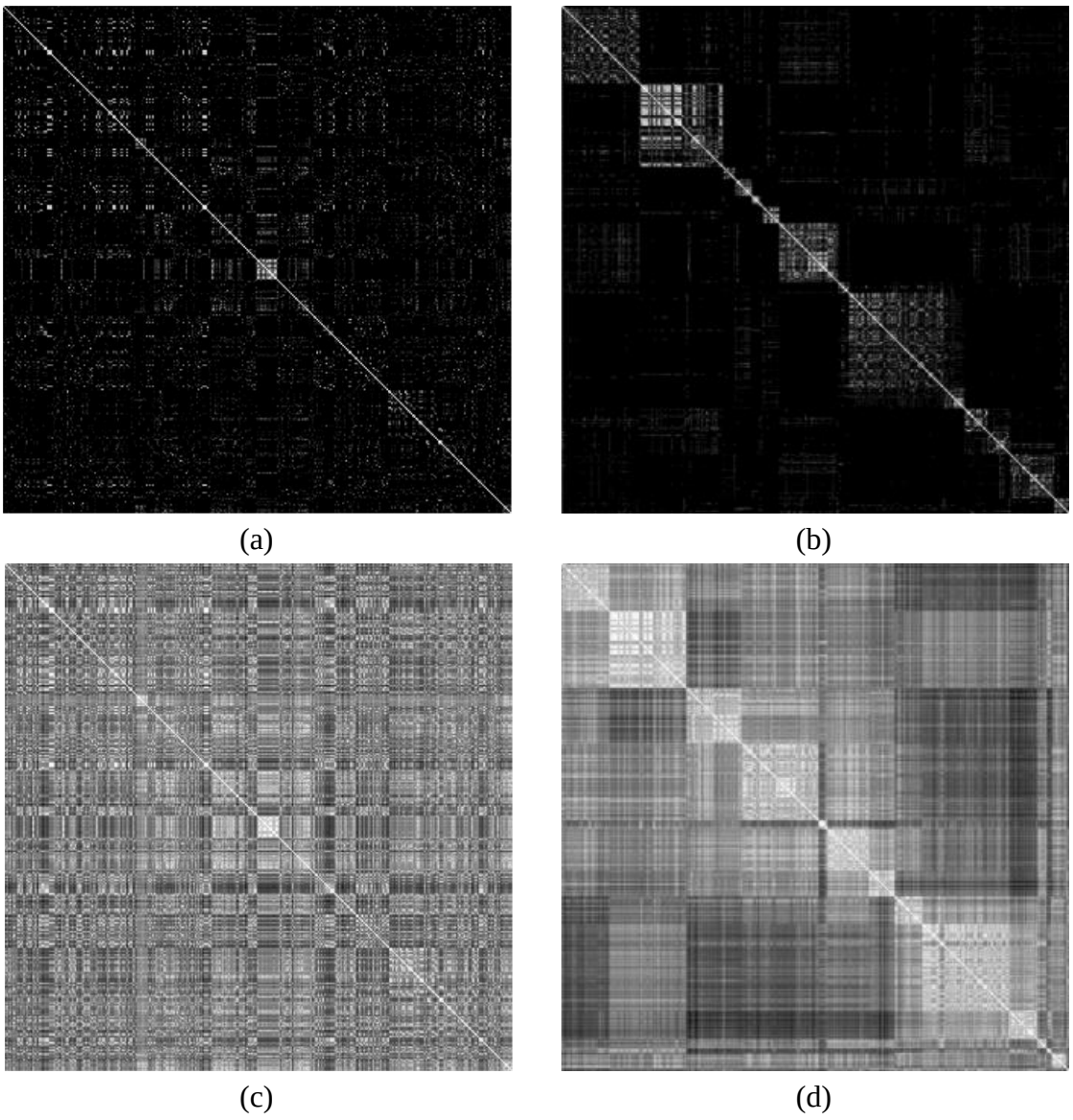


Figure 4.3: The plots of perceptual similarity matrices: (a) the original similarity matrix; (b) the original similarity matrix which are sorted according to 14 clusters obtained using hierarchical clustering analysis; (c) the 8D Isomap (8D-ISO) similarity matrix; and (d) the sorted 8D Isomap similarity matrix according to 14 clusters obtained using hierarchical clustering analysis.

4.2.3 Using Free-Grouping and Isomap Analysis ($POPJ_{ISO}$)

Considering the original perceptual similarity matrix obtained using free-grouping contains many zero entries, Isomap analysis [Tenenbaum et al., 2000] was used to obtain a higher resolution similarity matrix, i.e. an Isomap similarity matrix [Clarke et al., 2012]. The original perceptual similarity matrix $S_p(i,j)$ was first converted to a dissimilarity matrix $DS_p(i,j)$, where $DS_p(i,j) = 1 - S_p(i,j)$. Each texture was regarded as one

entry in this matrix and the value of $DS_p(i,j)$ represents the perceptual dissimilarity between the texture pair (i,j). The dissimilarity between two remote entries in the dissimilarity matrix was approximated by the length of the shortest path along those neighbouring entries connecting these. As a result, the majority of the dissimilarity matrix which had previously contained maximum dissimilarity values ($DS_p = 1$) now contained lower values of estimated dissimilarity based on connectivity information.

Figure 4.4 (left) shows residual variances for the computation of the Isomaps at different dimensionalities. The rate of decrease of the residual variance reduces after a dimensionality of five. At the same time, it is observed from Figure 4.4 (right) that Pearson's correlation coefficient [Field, 2009] between the original similarity matrix and the Isomap increases more slowly (>= 0.7180) after the dimensionality of the Isomap reaches eight. Figure 4.3 (c) presents the plot of the 8D Isomap (8D-ISO) similarity matrix.

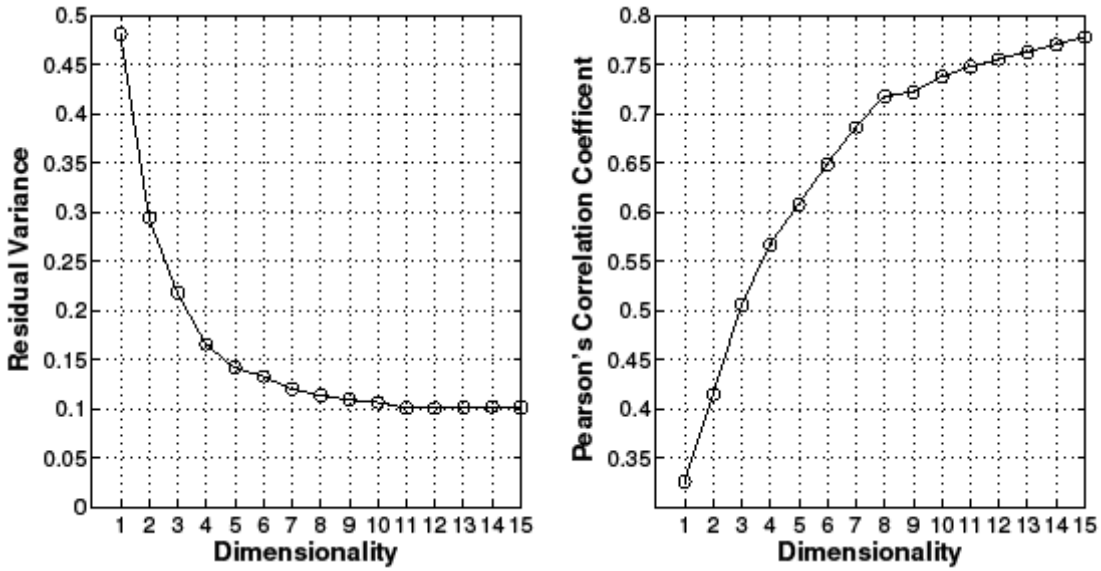


Figure 4.4: Performance of the computation of different Isomaps. (Left): residual variances for the computation of Isomaps at different dimensionalities; and (right): Pearson's correlation coefficients between the original similarity matrix and Isomaps at different dimensionalities when only the entries of two matrices corresponding to non-empty positions in the original similarity matrix are considered.

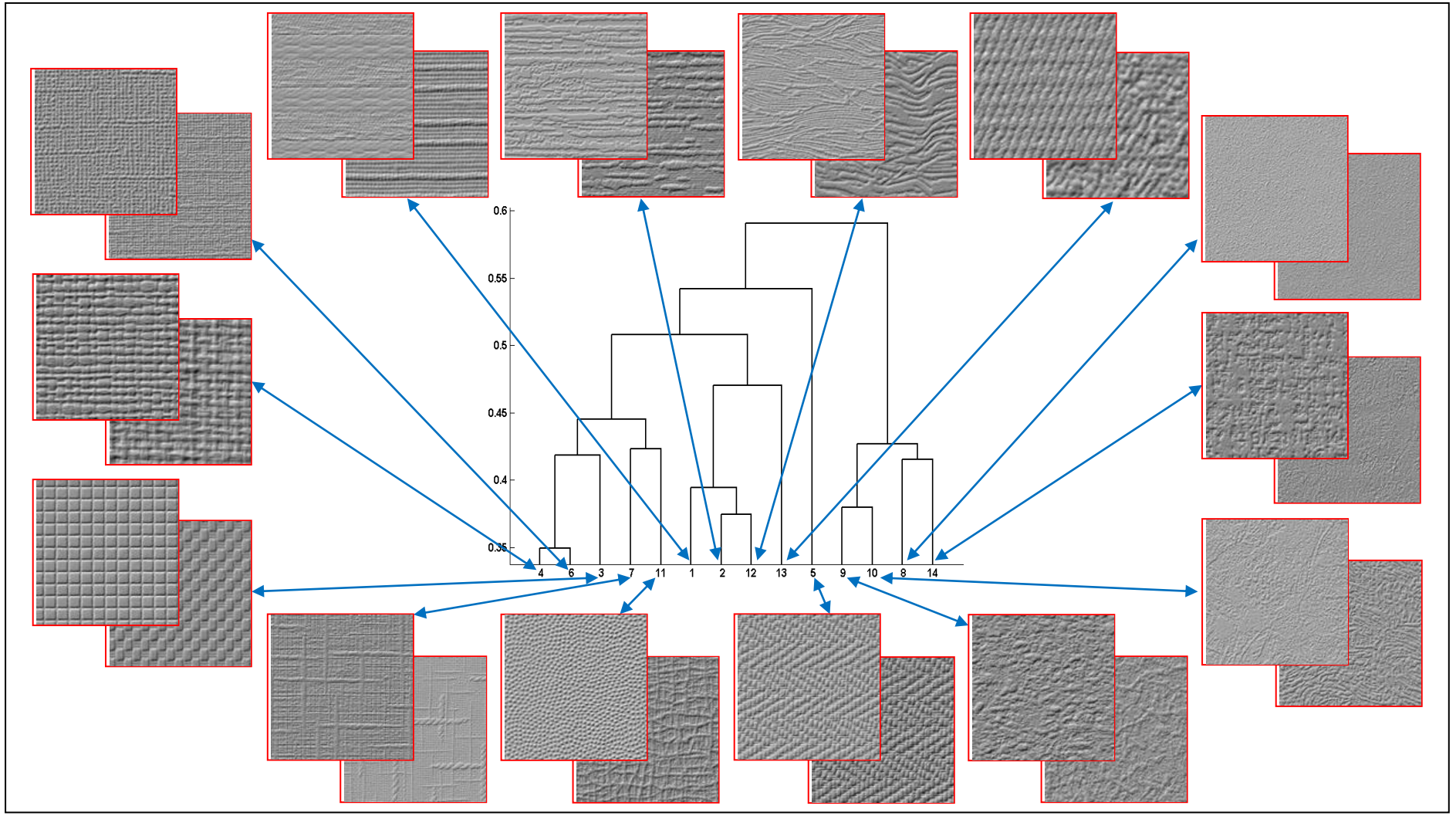


Figure 4.5: Dendrogram (cut at 0.337) obtained from 8D-ISO, along with two representative textures of each cluster.

Furthermore, hierarchical clustering analysis was also used to analyse the 8D-ISO data and 14 clusters were obtained by cutting the dendrogram at 0.337. Figure 4.5 shows the dendrogram obtained from 8D-ISO. The descriptions of the 14 clusters can be found in Appendix A. Although more clusters can also be obtained, the 14 clusters provide a reasonable insight into the organisation of the similarity of the 334 textures in *Pertex*. The 8D-ISO similarity matrix, sorted using the 14 clusters, is plotted in Figure 4.3 (d). It shows that the intra-cluster similarity is still retained while the inter-cluster similarity is not as sparsely represented as before.

Thus, it was decided that the 8D-ISO similarity matrix could be used to provide a valuable pair-of-pairs judgement set. Hence, Equations (4.3) and (4.4) were used to construct the third set of pair-of-pairs judgements, namely, $POPJ_{ISO}$.

4.2.4 Comparing the Three Pair-of-Pairs Judgement Sets

In the previous subsections, we obtained three sets of human-derived pair-of-pairs judgements: $POPJ_{POP}$, $POPJ_P$ and $POPJ_{ISO}$. Specifically, the $POPJ_{POP}$ was directly obtained using the pair-of-pairs experiment; the $POPJ_P$ was generated from the original perceptual similarity matrix obtained using the free-grouping experiment; and the $POPJ_{ISO}$ was constructed from the 8D-ISO similarity matrix derived by applying Isomap analysis to the original perceptual similarity matrix.

In the three sets of pair-of-pairs judgements, "0" can mean that the two pairs involved have the same "level" of similarity. This can be a function of the underlying human judgements but is also a function of the resolution of similarity data (the original perceptual similarity matrix or the 8D-ISO similarity matrix). Specifically, the numbers of the "zero-valued" judgements contained in the three pair-of-pairs judgement sets: $POPJ_{POP}$, $POPJ_P$ and $POPJ_{ISO}$ are 1, 712 and 0, respectively. Therefore, $POPJ_P$ is not suitable for the direct comparison with other pair-of-pairs judgement sets.

In this subsection, we will first introduce a method for comparing two sets of pair-of-pairs judgements and then use this method to pair-wise compare the three human-derived pair-of-pairs judgement sets.

The Method for Comparing Two Sets of Pair-of-Pairs Judgements

As discussed in Section 2.4.1, an accuracy-based measure is suitable for comparing two sets of pair-of-pairs judgements. In this case, we introduce "agreement rate" (%) to measure the consistency of two different pair-of-pairs judgement sets. The higher an agreement rate is, the more consistent the two pair-of-pairs judgement sets are.

The comparison of two different pair-of-pairs judgement sets is performed as below:

(1) Compute the criterion to decide whether or not the two pair-of-pairs judgements are consistent for each trial:

$$IsAgreed_i = (POPJ_A(i) == POPJ_B(i))?\,1:0, i = 1, 2, \dots, 1000, \quad (4.5)$$

where $POPJ_A(i)$ or $POPJ_B(i)$ could be a human-derived (perceptual) or computationally derived pair-of-pairs judgement; and

(2) Calculate the percentage agreement rate:

$$Agreement\ Rate\ (\%) = \sum_{i=1}^{1000} IsAgreed_i * 100/1000. \quad (4.6)$$

Comparing the Three Pair-of-Pairs Judgement Sets Pair-wise

As mentioned above, $POPJ_P$ contains 712 "zero-valued" judgements. However, not all of these judgements mean both pairs involved differ by the same level of similarity. Another possible reason is that many zero values in the perceptual similarity matrix derived using the free-grouping experiment were yielded due to the limited number of human observers. Hence, it is meaningless to directly compare $POPJ_P$ with other pair-of-pairs judgement sets (the "agreement" rates between $POPJ_{POP}$ and $POPJ_P$ and between $POPJ_{ISO}$ and $POPJ_P$ are only 27.8% and 27.9% respectively). In fact, only 26 out of the 1000 pairs of pairs constructed from that similarity matrix carry two non-zero similarity values. Thus, the low "agreement" rates are attributed to the sparseness of the similarity matrix derived from the free-grouping experiment.

As a result, only the 26 "valid" $POPJ_P$ judgements can be compared with 26 corresponding $POPJ_{POP}$ or $POPJ_{ISO}$ judgements. However, when $POPJ_{POP}$ and $POPJ_{ISO}$ were compared, all 1000 pair-of-pairs judgements were utilised. It is noteworthy that the only "zero-valued" $POPJ_{POP}$ judgement was produced from an equal number of human observers choosing "left" pair and "right" pair as more similar. Thus, it will be

kept for comparisons. The pair-wise comparison results are displayed in Table 4.1. It can be seen that the agreement rate between two sets of pair-of-pairs judgements: $POPJ_{POP}$ and $POPJ_{ISO}$ (see Equations (4.2) and (4.4)) obtained from the pair-of-pairs experiment and 8D-ISO respectively is 73.9%. It suggests that the two sets of judgements agree with each other well. Although only 1000 pairs of pairs used in the original pair-of-pairs experiment are examined, they were randomly constructed from the 334 textures in *Pertex*. Thus, the consistency between the $POPJ_{POP}$ and 8D-ISO indicates the effectiveness of 8D-ISO.

	$\boldsymbol{POPJ_{POP}}$	$\boldsymbol{POPJ_{P}}$	$\boldsymbol{POPJ_{ISO}}$	$\boldsymbol{POPJ_{R}}$
$\boldsymbol{POPJ_{POP}}$	100	80.8 (27.8)	73.9	48.25±1.55
$\boldsymbol{POPJ_{P}}$	80.8 (27.8)	100	84.6 (27.9)	14.40±0.85 (46.16±9.42)
$\boldsymbol{POPJ_{ISO}}$	73.9	84.6 (27.9)	100	50.00±1.58
$\boldsymbol{POPJ_{R}}$	48.25±1.55	14.40±0.85 (46.16±9.42)	50.00±1.58	100

Table 4.1: Pair-wise agreement rates (%) (see Equation (4.6)) obtained from three sets of perceptual pair-of-pairs judgements. Here, $POPJ_{POP}$, $POPJ_{P}$ and $POPJ_{ISO}$ denote pair-of-pairs judgements derived using the pair-of-pairs experiment directly, the free-grouping experiment, and both free-grouping and Isomap analysis, respectively. Especially, two figures are displayed with the use of only the 26 "valid" and all 1000 (inside the bracket) $POPJ_{P}$ judgements respectively. In addition, $POPJ_{R}$ stands for one million sets of pair-of-pairs judgements that we randomly generated (see text for more details).

When only the 26 "valid" $POPJ_{P}$ judgements were utilised, the agreement rates between these judgements and 26 corresponding $POPJ_{POP}$ judgements obtained from the pair-of-pairs experiment directly, and between these judgements and 26 corresponding $POPJ_{ISO}$ judgements obtained from the 8D-ISO similarity matrix, are 80.8% (21 out of 26) and 84.6% (22 out of 26), respectively. Both agreement rates suggest high consistencies. In addition, Figure 4.4 (right) shows that the value of the Pearson's correlation coefficient (between the original perceptual similarity matrix and the Isomap matrix) increases to 0.7180 when the dimensionality of the Isomap matrix approaches eight. This coefficient implies that 8D-ISO correlates well with the original similarity matrix derived from the

free-grouping experiment. However, 26 pair-of-pairs judgements are not sufficient for the benchmark dataset of an evaluation framework.

In addition, we also randomly generated one million sets of pair-of-pairs judgements ($POPJ_R$) and compared these data with $POPJ_{POP}$, $POPJ_P$ and $POPJ_{ISO}$. The means and standard deviations of achieved agreement rates are shown in Table 4.1. The low agreement rates (≈ 50%) suggest that the two different populations of human observers in the free-grouping and pair-of-pairs experiments [Halley, 2011A] [Clarke et al., 2012] did not make their judgements arbitrarily.

4.2.5 Summary

To summarise, the 8D-ISO similarity matrix not only retains most of the similarity information of the original perceptual similarity matrix obtained using free-grouping, but also agrees with the 1000 pair-of-pairs judgements obtained using the pair-of-pairs experiment well. Thus, $POPJ_{POP}$ and $POPJ_{ISO}$ (obtained using the pair-of-pairs experiment and free-grouping with Isomap analysis respectively) will be used as the ground-truth data in future evaluations.

4.3 Computationally Derived Pair-of-Pairs Judgements at Differing Resolutions

In this section, we derive pair-of-pairs judgements from a similarity matrix computed using a computational feature set. First of all, we choose distance measures for the computation of similarity matrices. Multi-pyramid analysis is then used to enlarge the spatial extent exploited by these features. Given one computational feature set, a series of similarity matrices are computed using a multi-pyramid scheme. Finally, a set of pair-of-pairs judgements is generated from each similarity matrix.

4.3.1 Distance Measures for Computing a Similarity Matrix

When texture similarity is estimated using computational feature vectors, a distance measure is required for estimating the dissimilarity between two textures. Given a texture database, after feature extraction is performed on each texture image, a distance

matrix can be pair-wise computed from all feature vectors using a distance measure. It is then converted into a similarity matrix.

As a simple but effective metric, the *Euclidean* distance [Deza and Deza, 2009] (see Equation (4.7)) is normally thought to be unsuitable for measuring the distance between two histograms. The most popular histogram-wise distance metrics include the *Chi-square* (χ^2) statistic [Press et al., 1992] (see Equation (4.8)), the *G* statistic (see Equation (4.9)) [Sokal and Rohlf, 1969], the *Bhattacharyya* distance (see Equation (4.10)) [Thacker et al., 1997] and histogram intersection (see Equation 4.11) [Swain and Ballard, 1991]. However, only the *Euclidean* distance and the *Chi-square* (χ^2) statistic are used in this thesis due to their popularity and simplicity. The *Chi-square* statistic is used as the distance measure for histogram-based features (see Table 2.1) while the *Euclidean* distance is chosen for all other features (see Table 2.2).

$$Euclidean(x,y) = \sqrt{\sum_i (x_i - y_i)^2} \tag{4.7}$$

$$\chi^2(x,y) = \frac{1}{2}\sum_i \frac{(x_i - y_i)^2}{x_i + y_i} \tag{4.8}$$

$$G(x,y) = 2\sum_i [x_i \log x_i - x_i \log y_i] \tag{4.9}$$

$$B(x,y) = 1 - \sum_i \sqrt{x_i} \cdot \sqrt{y_i} \tag{4.10}$$

$$HI(x,y) = \sum_i \min(x_i, y_i) \tag{4.11}$$

4.3.2 The Importance of Multi-pyramid

As discussed in Section 3.8, the 51 computational feature sets, excluding those filtering-based feature sets, only compute higher order statistics on small local neighbourhoods and do not consider the aperiodic spatial relationship of these statistics. In addition, we concluded that long-range higher order statistics are required to encode aperiodic long-range interactions. Thus, the spatial extent of the first stage is an important impact factor for the 51 feature sets.

Changing the size of local neighbourhoods can adjust the spatial extent exploited by local neighbourhood-based computational features. However, this does not apply to those feature sets which do not use local neighbourhoods. In addition, the computational cost will increase correspondingly. More imporatantly, it might produce an "averaging effect" and decrease the discriminatory power of features [Mao and Jain, 1992]. Hence, it is not practicable to enlarge the spatial extent exploited by all of the 51 feature sets simply by changing the size of their local neighbourhoods.

It is likely that human visual perception processes images using multiple levels of resolutions simultaneously and that this fact is important to the research of human perception [Koenderink, 1984]. In the fields of computer vision and pattern recognition, multi-resolution analysis is usually used to enhance the performance of features because such techniques allow larger spatial extent to be considered. Pyramid decompositions [Simoncelli, 2009], such as the Gaussian pyramid, Laplacian pyramid, wavelet pyramid and steerable pyramid decompositions, have been utilised for multi-resolution texture analysis, and out of these the Gaussian pyramid [Burt, 1981] is a popular tool for this type of approach. We therefore use this method in this thesis. During the process of the Gaussian pyramid decomposition, an input image is repeatedly filtered by low-pass filters and downsampled to generate a sequence of ever smaller and more abstract images. In Figure 4.6, four smaller images in the pyramid are low-pass filtered versions of the top-most image (i.e. the original texture image).

4.3.3 Computing Texture Similarity Matrices Using a Multi-pyramid Scheme

Given one computational feature set, six similarity matrices are calculated using a multi-pyramid scheme. The implementation is as follows (also see Figure 4.7):

(1) Each texture image is decomposed into five Gaussian pyramid sub-bands [Simoncelli, 2009] (see Figure 4.6) corresponding to five individual resolutions of 1024×1024, 512×512, 256×256, 128×128 and 64×64;

(2) Each sub-band is individually normalised to have an average intensity of 0 and standard deviation of 1 in order to remove the influence of 1st- and 2nd-order grey level (moment) statistics;

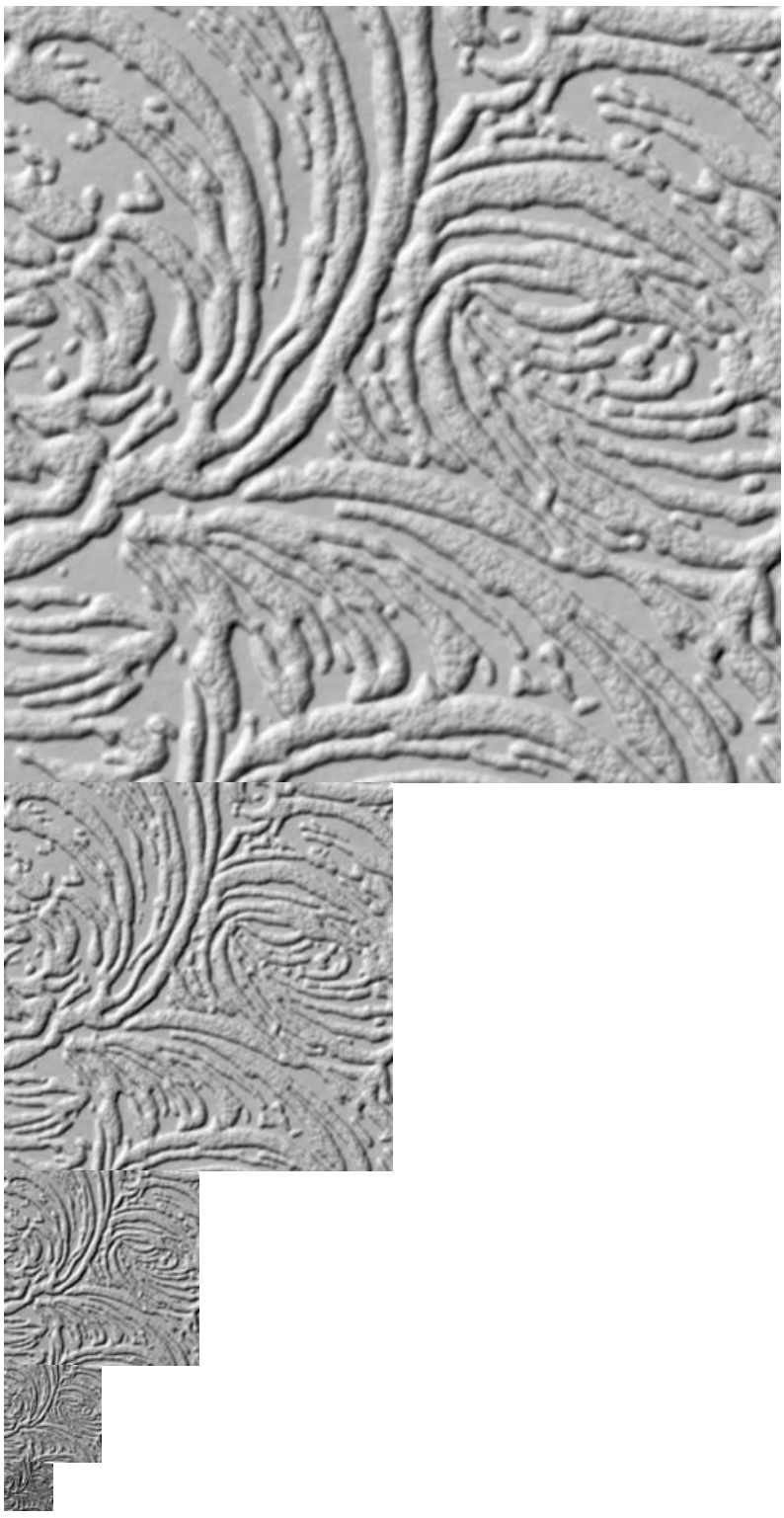


Figure 4.6: Five pyramid levels: level 0 (the original image), 1, 2, 3 and 4 of the top-left quarter of Texture "026" in Pertex obtained using the Gaussian pyramid decomposition.

(3) Feature extraction is performed to obtain a feature vector from each sub-band independently, and in addition all five feature vectors are combined into an additional feature vector. Thus, in total six feature vectors are generated for each texture. In this thesis, the combination of the five individual resolution feature vectors is referred to as the

"multi-resolution" feature vector. We also use the term "six resolutions" to refer to the five individual resolutions and the multi-resolution scheme together; and

(4) One pair-wise distance matrix is computed from all 334 sub-band images at each pyramid level. Each distance matrix is normalised to the range of [0, 1] and is then converted into a similarity matrix by subtracting 1. Hence, six computational similarity matrices are obtained for each feature set and are used as the computational estimates of the perceptual texture similarity.

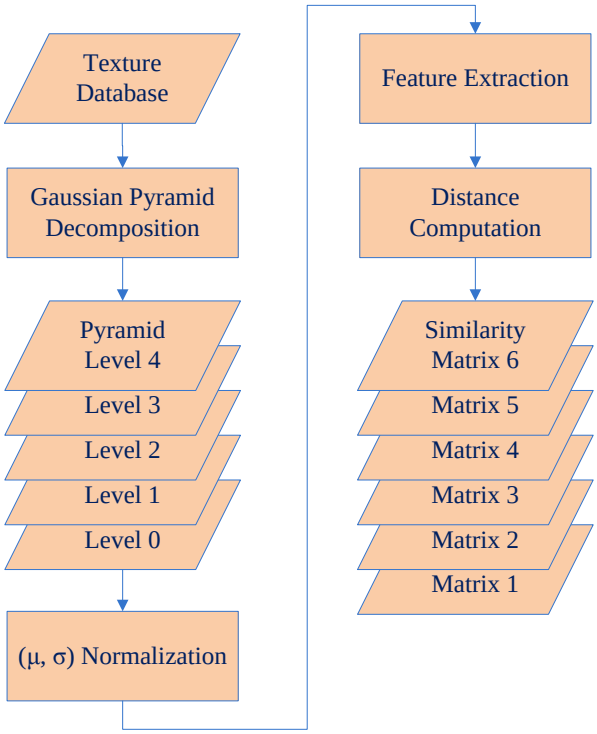


Figure 4.7: The pipeline of the computation of texture similarity using a multi-pyramid scheme.

4.3.4 Deriving Pair-of-Pairs Judgements Computationally

Given a computational similarity matrix, corresponding to the i-th ($i = 1, 2, \ldots, 1000$) trial of the original pair-of-pairs experiment, we will label the computational similarities of the left and right pairs as $CS_L(i)$ and $CS_R(i)$ respectively. The computational estimated judgement corresponding to the i-th trial, i.e. $J_E(i)$, is computed based on the difference between these values:

$$J_E(i) = CS_L(i) - CS_R(i), i = 1, 2, \ldots, 1000. \quad (4.12)$$

The computationally estimated pair-of-pairs judgement set, i.e. $POPJ_E$, is derived from a set of $J_E(i)$, as below:

$$POPJ_E(i) = \begin{cases} 1, & J_E(i) > 0 \\ 0, & J_E(i) = 0 \\ -1, & J_E(i) < 0 \end{cases}, i = 1, 2, \dots, 1000, \tag{4.13}$$

where "1" means, as before, that the left pair is more similar than the right one, "0" suggests that both pairs differ by the same level of similarity, and "-1" implies that the right pair is more similar than the left one.

4.3.5 Summary

To summarise, multi-pyramid analysis is utilised to expand the spatial extent exploited by computational features. When one computational feature set is applied, six similarity matrices are computed using a multi-pyramid scheme. Correspondingly, six sets of pair-of-pairs judgements are obtained from these similarity matrices.

4.4 Comparing Human and Computationally Derived Pair-of-Pairs Judgement Sets

As described in Section 4.2.4, "agreement rate" (%) was used to measure the consistency between human-derived and computationally derived pair-of-pairs judgement sets. The comparison process is performed for each feature set on each pyramid level as below:

(1) Compute the criterion to decide whether or not the pair-of-pairs judgements obtained using computational features and human observers are consistent for each trial:

$$IsAgreed_i = (POPJ_H(i) == POPJ_E(i))?\,1:0, i = 1, 2, \dots, 1000, \tag{4.14}$$

where $POPJ_H$ is $POPJ_{POP}$ or $POPJ_{ISO}$, and $POPJ_E$ is a computationally estimated pair-of-pairs judgement set. It should be noted that the normalisation operation (see step (4) in Section 4.3.3) might yield different resolutions of resultant data from different sources of distance matrices. As a result, some "zero-valued" $POPJ_E$ judgements might be produced because of this operation, which impairs the reliability of Equation (4.14). Fortunately, none of the computational pair-of-pairs judgement sets obtained in this

study contain "zero-valued" $POPJ_E$. Otherwise, all "zero-valued" $POPJ_E$ judgements and corresponding $POPJ_H$ judgements should be excluded from the comparison in Equation (4.14); and

(2) Calculate the percentage agreement rate using Equation (4.6).

Given one human derived ground-truth dataset, in total, six agreement rates are computed for each computational feature set at six different pyramid resolutions (including multi-resolution) respectively.

4.5 Conclusions

In this chapter, we introduced a pair-of-pairs based evaluation framework for benchmarking computational features. Compared with the existing evaluation frameworks for texture segmentation [Randen and Husøy, 1999], classification [Zhang et al., 2007] and retrieval [Khelifi and Jiang, 2011], the framework that we have proposed does the following:

(1) exploits the higher resolution (non-binary or non-Boolean-valued) perceptual texture similarity data obtained from a large texture database of 334 textures;

(2) enhances the spatial extent exploited by computational features using a multi-pyramid approach;

(3) is able to compare computationally derived pair-of-pairs similarity judgements and their perceptual counterparts obtained by human observers; and

(4) introduces a new performance measure: "agreement rate" (defined in Equation (4.6)). Thus, this framework is more suitable for the task of comparing higher resolution similarity data than the existing texture segmentation, classification and retrieval evaluation frameworks.

In the next chapter the results of two pair-of-pairs based evaluation experiments will be reported that investigate the ability of the computational features to estimate human-derived pair-of-pairs texture similarity.

Chapter 5

Pair-of-Pairs Based Evaluation Experiments

5.1 Introduction

In this chapter, we use the evaluation framework described in Chapter 4 to perform two evaluation experiments in order to examine the ability of computational features to estimate perceptual pair-of-pairs judgements derived from human observers.

Specifically, we use two complementary human derived pair-of-pairs judgement sets as the ground-truth data for the two experiments: $POPJ_{POP}$ (obtained using the original pair-of-pairs experiment) and $POPJ_{ISO}$ (constructed from the perceptual similarity matrix 8D-ISO). The "agreement rate" defined by Equation (4.6) is used as the performance measure. (This was designed to measure the consistency between computational and perceptual pair-of-pairs judgements). In each experiment, we investigate:

(1) whether or not the 51 feature sets perform well compared with the human derived pair-of-pairs judgements;

(2) whether or not there is a "best feature set" or "best feature category" (see Chapter 3); and

(3) which resolution (including a multi-resolution scheme) is the optimal one.

In this chapter, therefore, Sections 5.2 and 5.3 introduce the results of the two evaluation experiments. Section 5.4 discusses the consistency of these results, while in Section

5.5 we discuss the importance of long-range interactions to human perception. Finally, we present our conclusions in Section 5.6.

5.2 Evaluation Experiment Using $POPJ_{POP}$

In this section, the pair-of-pairs based evaluation framework introduced in Chapter 4 is applied. The perceptual pair-of-pairs judgement set: $POPJ_{POP}$ obtained in the "original" pair-of-pairs experiment (POP_O) is used as the ground-truth data. The agreement rate as defined by Equation (4.6) is utilised to measure the consistency between computational pair-of-pairs results and the $POPJ_{POP}$ judgements. The evaluation experiment was conducted on 51 computational feature sets under five individual resolutions and a multi-resolution scheme (see Section 4.3).

5.2.1 Overall Performance

Figure 5.1 shows the agreement rates (%) obtained using the 51 feature sets. It is observed that the average agreement rate over all 51 computational feature sets and six resolutions is 57.65%. However, the perceptual pair-of-pairs judgement set: $POPJ_{ISO}$ obtains a higher agreement rate of 73.9%. This provides limited validation of the perceptual pair-of-pairs judgement set $POPJ_{POP}$ with the difference being due to either the variability between different observer groups or the difference between the methods of deriving these two ground-truth datasets. In addition, the average agreement rate 48.25% calculated between one million randomly generated pair-of-pairs judgement sets (also see Section 4.2.4) and $POPJ_{POP}$ suggests that (1) the human observers did not make their judgements arbitrarily and (2) the judgements obtained using those computational features are also not random. However, the most obvious observation is that the performance of the 51 feature sets differs from that of human observers.

Particularly, Table B.1 (in Appendix B) illustrates all the agreement rates (%) displayed in Figure 5.1 in more detail. The highest agreement rate is obtained using LM [Leung and Malik, 2001] at the resolution of 128×128 while the lowest agreement rate 46.0% is derived using SRDM [Kim and Park, 1999] at the resolution of 512×512. If multi-resolution is only considered, the best agreement rate, 66.3%, is obtained using MRSAR [Mao and Jain, 1992]. In this case, SVR [Harwood et al., 1995] are outperformed by the other feature sets as it only obtains an agreement rate of 46.9%. Considering the varia-

tion of the performance of the 51 feature sets over the six resolutions, we cannot determine the best feature set. We will, therefore, examine their average performance across the six resolutions in the next subsection.

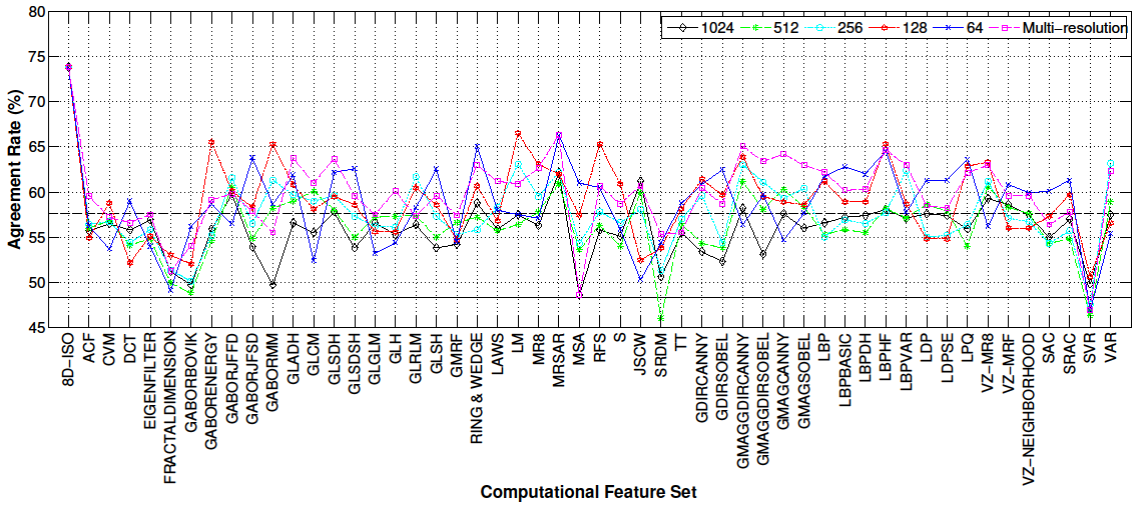


Figure 5.1: Agreement rates (%) obtained using 51 sets of computational features at five individual resolutions and the multi-resolution scheme against the perceptual pair-of-pairs judgement set $POPJ_{POP}$. The agreement rate 73.9% between the $POPJ_{ISO}$ ("8D-ISO") and $POPJ_{POP}$ is also shown. Additionally, the black dashed line shows the average agreement rate 57.65% (calculated over the 51 feature sets and six resolutions). The black solid line displays the average agreement rate 48.25% computed between one million randomly generated pair-of-pairs judgement sets and $POPJ_{POP}$.

5.2.2 Average Performance across Resolutions

In order to remove the effect of the resolution on performance, the average agreement rates (%) for the 51 sets of computational features and the corresponding 95% confidence intervals are computed over the different resolutions and displayed in Figure 5.2. It can be seen that MRSAR [Mao and Jain, 1992] outperforms its counterparts in this case with an agreement rate of 63.32% and the worst performance (48.00%) is obtained using SVR [Harwood et al., 1995]. However, the 95% confidence interval of the performance obtained using MRSAR is ±1.89% which means an unstable performance. The title of "the best feature set" could possibly be "granted" to other feature sets by varying the resolution. Thus, we cannot decide the best feature set. In this situation, it is not possible to determine the best feature category either.

Considering the performance produced using the 51 feature sets spread out across different resolutions, in the next subsection we will investigate the effect of the resolution on the performance of these feature sets in order to determine whether or not an optimal resolution exists.

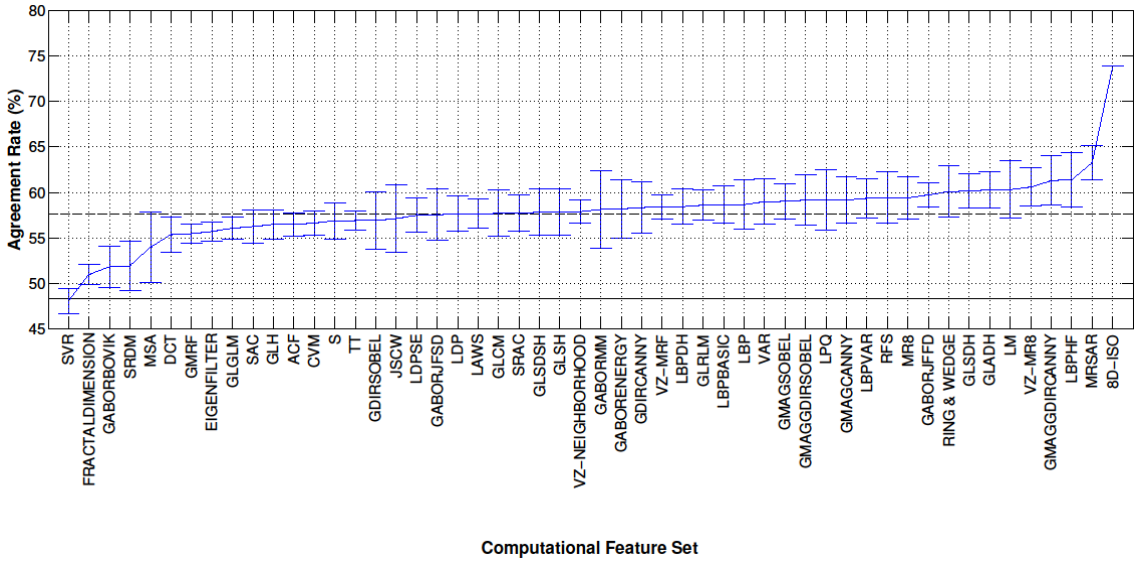


Figure 5.2: Average agreement rates (%, sorted in an ascending order) and 95% confidence intervals (error bars) over five individual resolutions and the multi-resolution scheme. Agreement rates are derived between the pair-of-pairs judgments obtained using 51 sets of computational features and the $POPJ_{POP}$ judgements. The agreement rate 73.9% between the $POPJ_{ISO}$ ("8D-ISO") and $POPJ_{POP}$ judgement sets is also shown. The black dashed line suggests the average agreement rate 57.65% (computed over the 51 feature sets and six resolutions). In addition, the black solid line displays the average agreement rate 48.25% computed between one million randomly generated pair-of-pairs judgement sets and $POPJ_{POP}$.

5.2.3 Performance at Different Resolutions

We investigate the performance of the 51 computational feature sets obtained at different resolutions in order to determine whether or not an optimal resolution exists for these feature sets.

Optimal Performance across Six Resolutions

Figure 5.3 shows the highest agreement rate obtained for each feature set, across five individual resolutions and the multi-resolution scheme. It can be observed that the 51

feature sets obtained their optimal performance over a variety of resolutions. We will therefore examine the average performance over the 51 feature sets at each resolution.

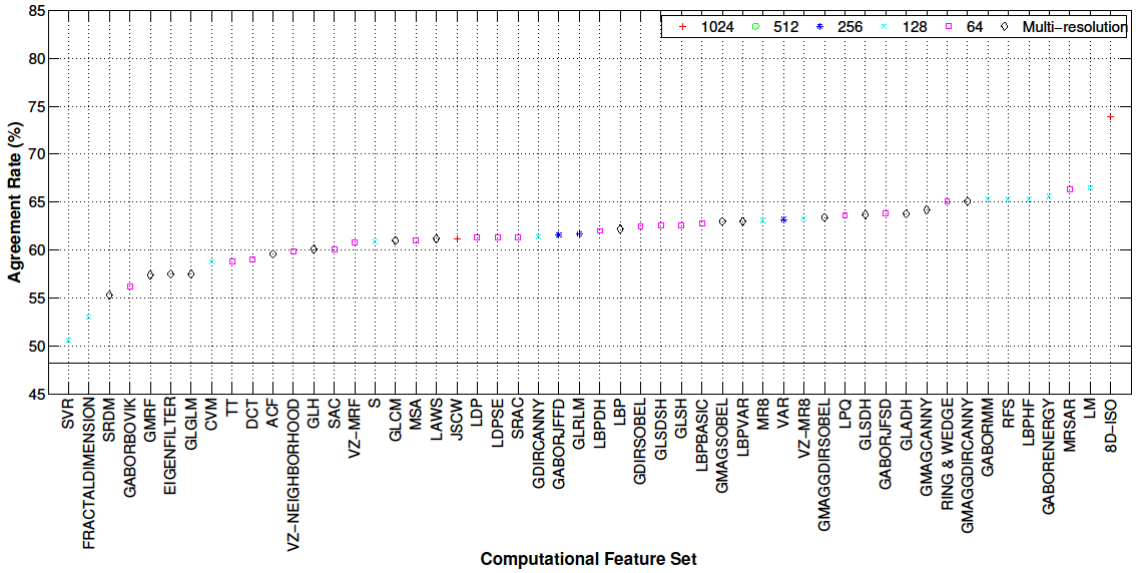


Figure 5.3: The optimal agreement rates (sorted in an ascending order) derived using 51 computational feature sets against $POPJ_{POP}$ and the corresponding best resolutions for these feature sets, across five individual resolutions and the multi-resolution scheme. Besides, the agreement rate 73.9% obtained using another set of perceptual pair-of-pairs judgements $POPJ_{ISO}$ ("8D-ISO") is also presented. The black solid line displays the average agreement rate 48.25% computed between one million randomly generated pair-of-pairs judgement sets and $POPJ_{POP}$.

Average Performance over 51 Feature Sets at Six Resolutions

Figure 5.4 shows the average agreement rates and 95% confidence intervals obtained using five individual resolutions and the multi-resolution scheme over the 51 computational feature sets. In order to test the significance of the effect of the resolution on the agreement rates obtained using those feature sets, a one-way repeated-measures ANOVA (Analysis of Variance) was conducted. Given that we regard the 51 feature sets as a population, the agreement rate obtained using different feature sets and the resolution can be considered as the dependent variable and the independent variable of a one-way repeated-measures ANOVA respectively. By using Mauchly's test [Field, 2009], it is indicated that the assumption of sphericity was violated, $\chi^2(14) = 78.63$, $p < 0.05$. Hence, degrees of freedom were corrected using Greenhouse-Geisser estimates of sphericity [Field, 2009] ($\varepsilon = 0.623$). The results show that the agreement rates obtained us-

ing the 51 feature sets were significantly affected by the resolution, $F(3.11, 155.71) = 15.97$, $p < 0.05$.

Furthermore, it can be seen that the average agreement rate obtained at the multi-resolution scheme is better than the other resolutions. However, the agreement rate varies much at the multi-resolution scheme with a 95% confidence interval of ±1.06%. In this situation, it is not feasible to take the multi-resolution scheme as the optimal resolution directly.

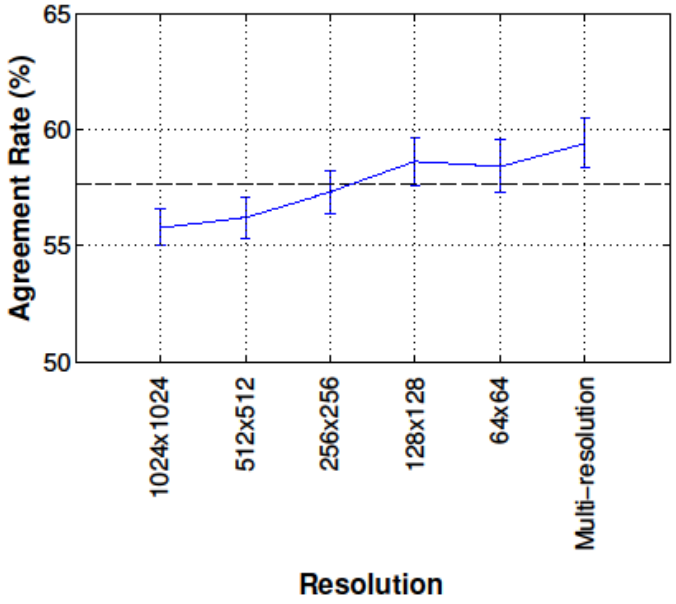


Figure 5.4: Average agreement rates and 95% confidence intervals (error bars) computed using five individual resolutions and multi-resolution against $POPJ_{POP}$ over 51 feature sets. The black dashed line suggests the average agreement rate 57.65% (calculated over the 51 feature sets and six resolutions).

The post hoc tests were then performed using the Bonferroni correction. The results reveal that the agreement rates obtained at resolutions of 256×256, 128×128, 64×64 and multi-resolution are significantly different from those obtained at the resolution of 1024×1024 (p = 0.0092, 0.0002, 0.0017 and 0.0000 < 0.05). However, there is no significant difference between the agreement rates derived using multi-resolution and the resolution of 128×128 (p = 1.0000), and using multi-resolution and the resolution of 64×64 (p = 1.0000). In this context, 128×128, 64×64 and multi-resolution could be the candidates of the optimal resolution.

Effect of Using Multi-resolution

In this experiment, the multi-resolution scheme is used to enhance the spatial extent exploited by the 51 computational feature sets in order to improve their performance compared with that they "achieved" at the original resolution (i.e. 1024×1024). When we compare the performance of each feature set at the resolution of 1024×1024 and multi-resolution, it is found that 49 of the 51 feature sets performed better at multi-resolution than at 1024×1024. This number is 31, 34, 40 and 38 when the resolutions of 512×512, 256×256, 128×128 and 64×64 are considered respectively. To be specific, only JSCW [Portilla and Simoncelli, 2000] and SVR [Harwood et al., 1995] behaved slightly worse at multi-resolution with the agreement rates: 60.7% and 46.9% than that they performed at the resolution of 1024×1024 with the agreement rates: 61.2% and 49.8%. As a result, the multi-resolution scheme is chosen for further research.

5.2.4 Summary

In this section, an evaluation experiment was conducted using the pair-of-pairs based evaluation framework proposed in Chapter 4. The perceptual pair-of-pairs judgement set $POPJ_{POP}$ was used as the ground-truth data. The agreement rates obtained using 51 computational feature sets are generally distributed within the range from 46.0% to 66.5%. Meanwhile, the average agreement rate over all 51 feature sets and six resolutions (including multi-resolution) is 57.65%. Obviously, none of the performance is comparative with the agreement rate 73.9% computed between the two sets of perceptual pair-of-pairs judgements: $POPJ_{POP}$ and $POPJ_{ISO}$. Furthermore, the performance of the 51 feature sets is not stable across different resolutions. Thus, it is not practical to determine the best feature set. In this case, we can also not decide the best feature category. The results of the one-way repeated-measures ANOVA, corrected using Greenhouse-Geisser estimates of sphericity, show that the agreement rates obtained using the 51 feature sets were significantly affected by the resolution. However, each of resolutions of 128×128, 64×64 and multi-resolution could be chosen as the optimal resolution. Nevertheless, only the multi-resolution scheme was chosen for further research because it can improve the performance of more (49 of the 51) feature sets than the other resolutions compared with that obtained at the original resolution (i.e. 1024×1024).

5.3 Evaluation Experiment By Means of $POPJ_{ISO}$

Similar to the experiment introduced in the previous section, a second pair-of-pairs based evaluation experiment is conducted in this section. The experimental setup is the same as that used in Section 5.2 except that the perceptual pair-of-pairs judgement set: $POPJ_{ISO}$ generated from the 8D-ISO similarity matrix is used as the ground-truth data.

5.3.1 Overall Performance

Figure 5.5 shows agreement rates (%) between the pair-of-pairs judgements obtained using the 51 sets of computational features at five individual resolutions and the multi-resolution scheme and the perceptual pair-of-pairs judgement set $POPJ_{ISO}$. Clearly, the agreement rate of 73.9% between two sets of perceptual pair-of-pairs judgements: $POPJ_{POP}$ and $POPJ_{ISO}$ is the highest. In contrast, the average agreement rate 50% computed between one million randomly generated pair-of-pairs judgement sets (also see Section 4.2.4) and $POPJ_{ISO}$ indicates that (1) human observers did not make their judgements arbitrarily and (2) the judgements obtained using the 51 computational feature sets are not random. Similar to that displayed in Figure 5.1, the performance of the 51 feature sets is much lower than this agreement rate. Compared with human perceptual judgements, the performance of the 51 feature sets is inferior.

In addition, Table B.2 (in Appendix B) lists all the agreement rates shown in Figure 5.5 in more detail. It can be seen that, for the computational feature sets, the highest agreement rate 60.7% is obtained using MRSAR [Mao and Jain, 1992] while the lowest agreement rate 46.5% is derived using JSCW [Portilla and Simoncelli, 2000] at the resolution of 64×64. When only the multi-resolution condition is considered, the best performance (60.0%) is obtained using MRSAR. However, SVR [Harwood et al., 1995] performs worst at an agreement rate of 49.3% among the 51 feature sets in the same conditions. As we observed in Section 5.2, the performance of the 51 feature sets varied much across the six resolutions. As a result, it is not practical to decide which feature set is the best one. Thus, we will investigate their average performance across the six resolutions in the next subsection.

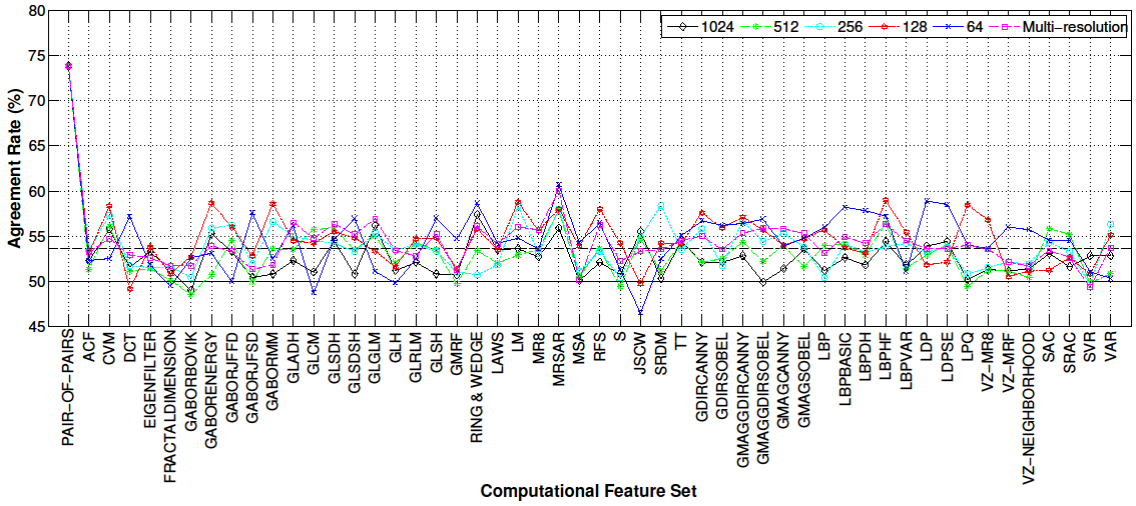


Figure 5.5: Agreement rates (%) between the pair-of-pairs judgements obtained using 51 sets of computational features at five individual resolutions and the multi-resolution scheme and the perceptual pair-of-pairs judgement set $POPJ_{ISO}$. The agreement rate 73.9% between two sets of perceptual pair-of-pairs judgements: $POPJ_{POP}$ ("PAIR-OF-PAIRS") and $POPJ_{ISO}$ is also reported. In addition, the black dashed line suggests the average agreement rate 53.59% (computed over the 51 feature sets and six resolutions). In addition, the black solid line displays the average agreement rate 50.00% computed between one million randomly generated pair-of-pairs judgement sets and $POPJ_{ISO}$.

5.3.2 Average Performance across Resolutions

Average agreement rates (%) and 95% confidence intervals obtained using the 51 feature sets are computed over five individual resolutions and multi-resolution in order to remove the effect of the resolutions. Figure 5.6 displays this information in detail. It can be observed that MRSAR [Mao and Jain, 1992] obtains the highest agreement rate: 58.4% and SVR [Harwood et al., 1995] are outperformed by all its counterparts at an agreement rate of 50.55%. Although MRSAR produced the best performance, the 95% confidence interval of its performance is ±1.89% which suggests an unstable performance. We, hence, cannot take it as the best feature set because it might be outperformed by certain other feature sets when the resolution changes. In this case, it is also not possible to determine the best feature category.

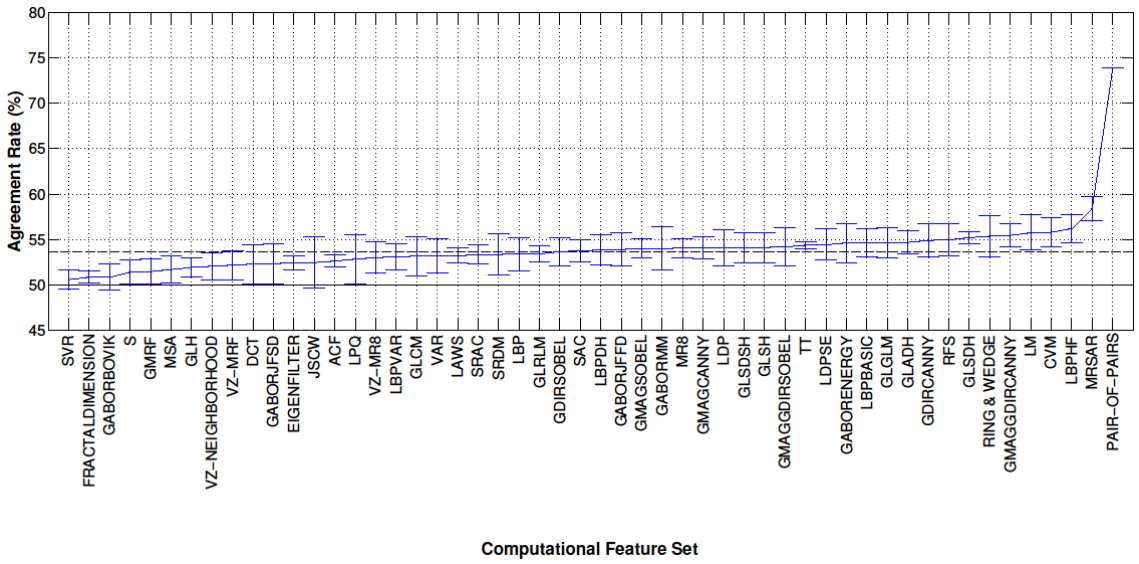


Figure 5.6: Average agreement rates (%, sorted in an ascending order) and 95% confidence intervals (error bars) over five individual resolutions and multi-resolution. Agreement rates are obtained between the pair-of-pairs judgments derived using 51 sets of computational features and $POPJ_{ISO}$. The agreement rate 73.9% between two sets of perceptual pair-of-pairs judgements: $POPJ_{POP}$ ("PAIR-OF-PAIRS") and $POPJ_{ISO}$ is also reported. The black dashed line suggests the average agreement rate 53.59% (calculated over the 51 feature sets and six resolutions). In addition, the black solid line displays the average agreement rate 50.00% computed between one million randomly generated pair-of-pairs judgement sets and $POPJ_{ISO}$.

Since the agreement rates obtained using the 51 feature sets spread out over the six resolutions, we will further examine the effect of the resolution on the performance of these feature sets in order to decide whether or not an optimal resolution exists in the next subsection.

5.3.3 Performance at Different Resolutions

In this subsection, we examine the performance of the 51 computational feature sets obtained at different resolutions in order to decide the optimal resolution for these.

Optimal Performance across Six Resolutions

When only the highest agreement rate obtained using one computational feature set across five resolutions and multi-resolution is considered, the optimal agreement rates achieved are plotted in Figure 5.7. The results show that the optimal performance of the

51 feature sets was obtained at various resolutions. As a result, in order to make sure whether or not there is an optimal resolution, we will investigate the average performance over the 51 feature sets.

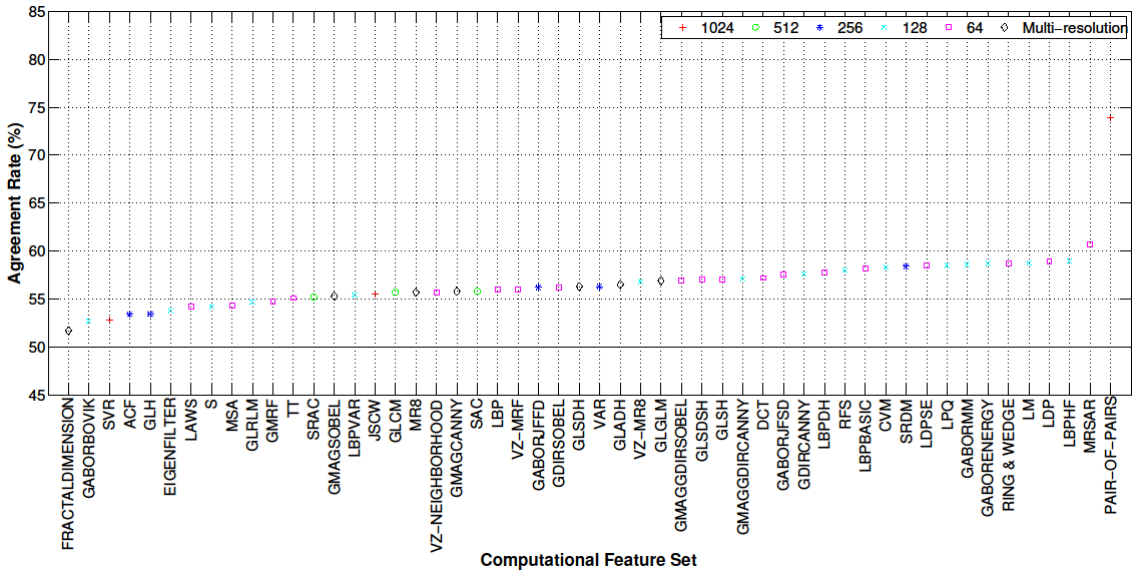


Figure 5.7: The optimal agreement rates (sorted in an ascending order) achieved using 51 computational feature sets against $POPJ_{ISO}$ and the corresponding best resolutions for these feature sets, across five individual resolutions and the multi-resolution scheme. In addition, the agreement rate 73.9% obtained using the perceptual pair-of-pairs judgement set $POPJ_{POP}$ ("PAIR-OF-PAIRS") is also shown. The black solid line displays the average agreement rate 50.00% computed between one million randomly generated pair-of-pairs judgement sets and $POPJ_{ISO}$.

Average Performance over 51 Feature Sets at Six Resolutions

Figure 5.8 shows the average agreement rates and 95% confidence intervals derived using five individual resolutions and the multi-resolution scheme over the 51 computational feature sets. As we did in Section 5.2.3, we performed a one-way repeated-measures ANOVA (Analysis of Variance) in order to test the significance of the effect of the resolution on the agreement rates obtained using these feature sets. The results of Mauchly's test [Field, 2009] show that the assumption of sphericity was violated, $\chi^2(14) = 64.39$, $p < 0.05$. Degrees of freedom were therefore corrected using Greenhouse-Geisser estimates of sphericity [Field, 2009] ($\varepsilon = 0.651$). The results show that the agreement rates obtained using the 51 feature sets were significantly affected by the resolution, $F(3.25, 162.66) = 9.61$, $p < 0.05$.

It can also be seen that the average performance obtained at 128×128 is better than that obtained at the other resolutions. However, the agreement rate for the resolution of 128×128 has a wide 95% confidence interval of ±0.7052%. Therefore, we cannot directly choose 128×128 as the optimal resolution because it might not be the optimal one any more when different feature sets are considered.

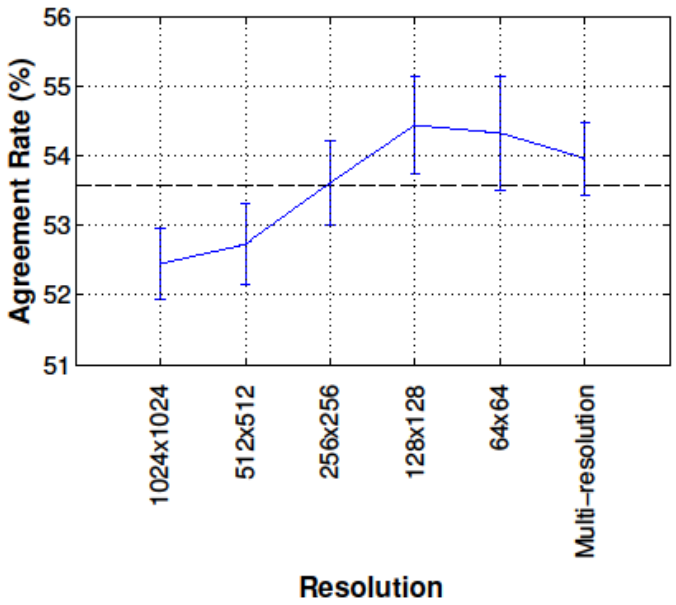


Figure 5.8: Average agreement rates and 95% confidence intervals (error bars) computed using five individual resolutions and the multi-resolution scheme against $POPJ_{ISO}$ over 51 feature sets. The black dashed line shows the average agreement rate 53.59% (computed over the 51 feature sets and six resolutions).

Furthermore, the post hoc tests were conducted using the Bonferroni correction. It is revealed that the agreement rates obtained using resolutions of 256×256, 128×128, 64×64 and multi-resolution are significantly different from those obtained using the resolution of 1024×1024 (p = 0.0141, 0.0001, 0.0035 and 0.0000 < 0.05). However, no significant differences between the agreement rates derived using multi-resolution and the resolution of 256×256 (p = 1.0000), using multi-resolution and the resolution of 128×128 (p = 1.0000), and using multi-resolution and the resolution of 64×64 (p = 1.0000) are found. As a result, 256×256, 128×128, 64×64 and multi-resolution are regarded as the candidates for the optimal resolution.

Effect of Using Multi-resolution

Since we cannot choose the optimal resolution according to the average performance across the six resolutions, as we did in Section 5.2.3, we compare the performance of

each feature set derived at the original resolution (i.e. 1024×1024) with that obtained at the other resolutions. It has been found that 27, 37, 39, 39 and 43 of the 51 feature sets perform better at 512×512, 256×256, 128×128, 64×64 and multi-resolution than at 1024×1024, respectively. In this case, the multi-resolution scheme is more suitable for the optimal resolution than 128×128 although the average performance obtained at the former is slightly lower than that derived at the latter. Specifically, eight feature sets worked worse at the multi-resolution scheme than at the resolution of 1024×1024. Table 5.1 lists the corresponding agreement rates obtained using these eight feature sets at the resolution of 1024×1024 and multi-resolution. Hence, the multi-resolution can be used to enhance the performance of the 51 feature sets and is chosen for further investigation.

Method	**Resolution**	
	1024×1024	**Multi-resolution**
CVM	55.8	54.7
EIGENFILTER	53.3	52.6
GABORENERGY	55.3	53.9
RING & WEDGE	57.4	56.1
JSCW	55.5	53.3
LDP	53.9	53.6
LDPSE	54.4	53.6
SVR	52.8	49.3

Table 5.1: Agreement rates (%) obtained using eight sets of computational features at the resolution of 1024×1024 and multi-resolution against the perceptual pair-of-pairs judgement set $POPJ_{ISO}$. These feature sets performed worse when multi-resolution was considered than that they performed when the resolution of 1024×1024 was used.

5.3.4 Summary

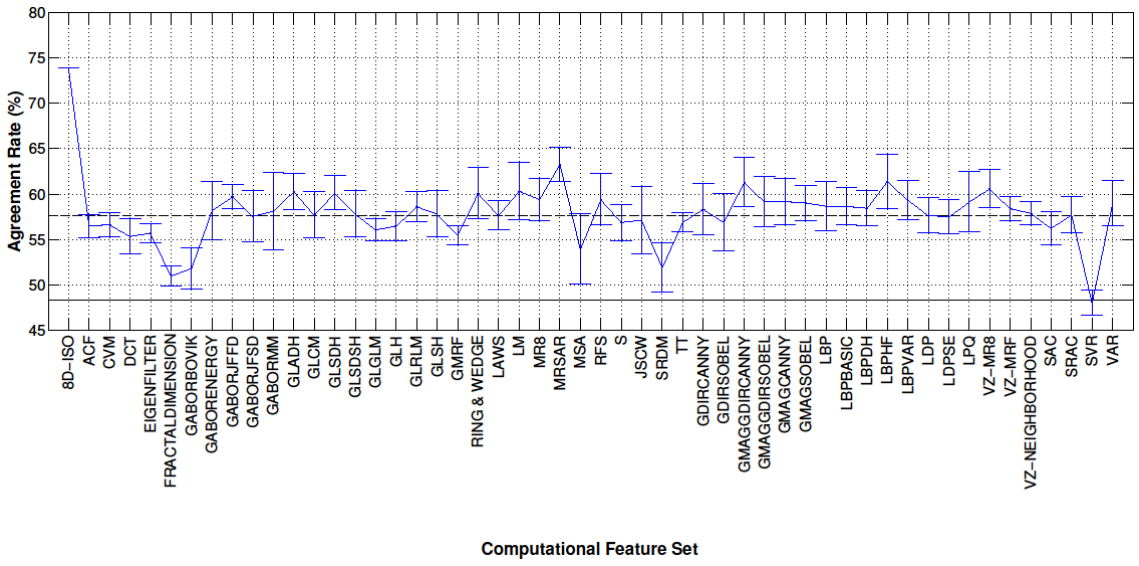
In this section, an evaluation experiment was carried out against the ground-truth pair-of-pairs judgement set $POPJ_{ISO}$ by means of the pair-of-pairs based evaluation framework proposed in Chapter 4. However, the agreement rates between the pair-of-pairs judgements obtained using 51 computational feature sets and $POPJ_{ISO}$ are only between 46.5% and 60.7%. These agreement rates are lower than the agreement rate of 73.9% which is obtained using another set of perceptual pair-of-pairs judgements $POPJ_{POP}$ against $POPJ_{ISO}$. In addition, the average agreement rate over all 51 feature sets and six resolutions (including multi-resolution) is only 53.59%. Clearly, the 51 feature sets performed poorly compared with human observers. Since the 51 feature sets performed unstably throughout the six resolutions, we cannot determine the best feature set nor the

best feature category. In addition, the results of the one-way repeated-measures ANOVA, corrected using Greenhouse-Geisser estimates of sphericity, show that the agreement rates obtained using the 51 feature sets were significantly affected by the resolution. However, each of resolutions of 256×256, 128×128, 64×64 and multi-resolution could be chosen as the optimal resolution. Nevertheless, the multi-resolution scheme was eventually chosen for further investigation because it improved the performance of more (43 of the 51) feature sets than the other resolutions compared with the original resolution of 1024×1024.

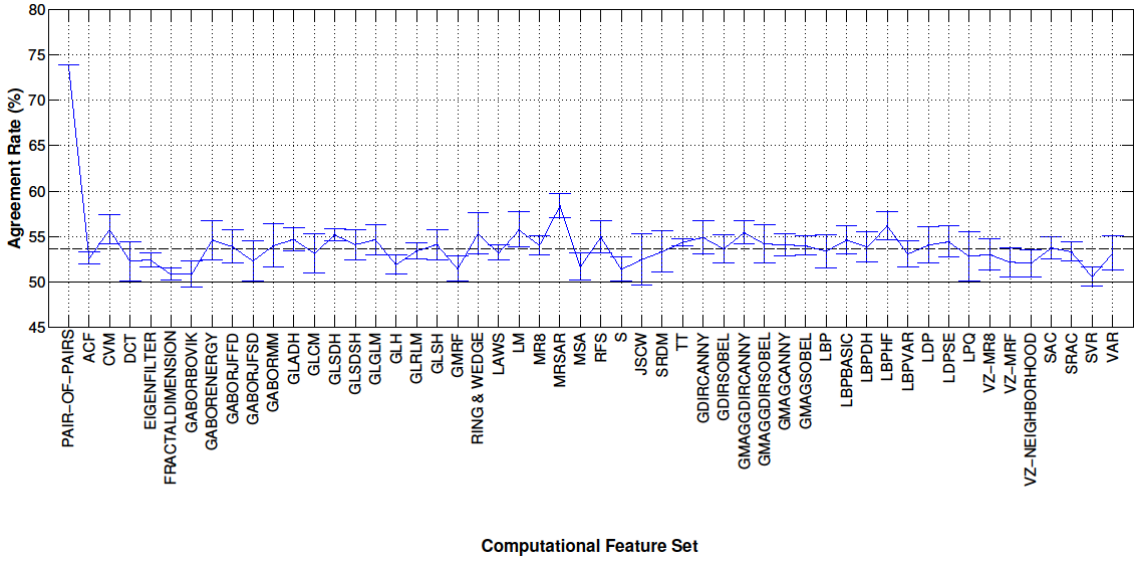
5.4 Consistency of the Results of Two Experiments

Figure 5.9 (a) displays the average agreement rates (%) between the pair-of-pairs judgments derived using the 51 computational feature sets over six resolutions compared against the perceptual pair-of-pairs judgement set $POPJ_{POP}$. In addition, the agreement rate obtained using $POPJ_{ISO}$ is also shown for comparison purposes. The $POPJ_{ISO}$ data produces the highest agreement rate at 73.9% providing validation of the perceptual pair-of-pairs judgement set $POPJ_{POP}$. However, the performance of the 51 sets of computational features is weak (the average agreement rates lie in the range from 48.00% to 63.32%). Meanwhile, Figure 5.9 (b) provides a similar plot in which the performance of the same 51 feature sets are compared against the other perceptual pair-of-pairs judgement set $POPJ_{ISO}$. The highest and lowest performances: 58.40% and 50.55% are provided by using MRSAR [Mao and Jain, 1992] and SVR [Harwood et al., 1995] respectively. It can be seen that the two curves in Figures 5.9 (a) and (b) are similar. In both cases the performance of the 51 feature sets is poor when compared against the two sets of human data.

Furthermore, Table 5.2 shows Spearman's correlation coefficients [Field, 2009] (α = 0.05), (columns 2-7) between the two agreement rate curves in Figures 5.1 and 5.5, and (column 8) between the two average agreement rate curves in Figures 5.9 (a) and (b). What is reflected is that the two groups of agreement rates obtained using the same 51 sets of computational features against two sets of different perceptual pair-of-pairs judgements are closely correlated with each other. It also indicates that the two sets of human perceptual data are consistent with each other as discussed in Section 4.2.4.



(a)



(b)

Figure 5.9: Average agreement rates and standard deviations (error bars) over five individual resolutions and multi-resolution. Agreement rates are calculated between the pair-of-pairs judgements derived using 51 feature sets and human data: $POPJ_{POP}$ (a) and $POPJ_{ISO}$ (b) respectively. The black dashed lines show the average agreement rates: 57.65% and 53.59% (computed over the 51 feature sets and six resolutions). In addition, the black solid lines display the average agreement rates: 48.25% and 50.00% computed between one million randomly generated pair-of-pairs judgement sets and $POPJ_{POP}$(a) and $POPJ_{ISO}$ (b) respectively.

	1024	512	256	128	64	Multi-	Mean
Correlation Coefficient	0.576	0.492	0.558	0.852	0.822	0.698	0.611

Table 5.2: Spearman's correlation coefficients (α = 0.05): (columns 2-7) between the two agreement rate curves in Figures 5.1 and 5.5; and (column 8) between the two curves in Figures 5.9 (a) and (b). Here, "1024", "512", "256", "128", "64" and "Multi-" mean the five individual resolutions and multi-resolution, and "Mean" denotes the case when the average agreement rates obtained over the six resolutions is considered.

5.5 Discussion

Generally speaking, the majority of the 51 computational feature sets examined in this study, excluding those filtering-based feature sets, do not consider the spatial relationship of the local features computed in the first stage of feature extraction. Thus, these "orderless" feature sets cannot capture a shape or segment an object from its surrounding scene [Lazebnik et al., 2006]. In this situation, spatial extent is important to these feature sets for encoding long-range image structure. However, only limited spatial extent is normally utilised by those feature sets. Even though a larger local neighbourhood can also be chosen for a number of feature sets, the computational cost will increase correspondingly. In addition, it possibly produces an "averaging effect" and decreases the discriminatory power of features [Mao and Jain, 1992]. A multi-pyramid scheme has been used in order to enlarge the spatial extent that is exploited by the 51 feature sets (also see Section 4.3). No matter whether five pyramid resolutions were used separately or together, the computational results never agreed well with humans' perceptual similarity judgments. Obviously, enhancing spatial extent in this way is not sufficient for capturing the complexity of human perception.

As discussed in Section 3.8, none of the 51 feature sets can encode aperiodic long-range interactions. As an energy density measure, the power spectrum is normally used to measure the periodicity of an input signal, e.g. the periodic texture structure [Liu, and Picard, 1998]. However, the phase information encodes most of the aperiodic structure information within an image [Oppenheim and Lim, 1991]. Thus, we hypothesise that the power spectrum is more important to the 51 feature sets than the phase spectrum. This hypothesis is tested in Appendix C. It was found that the pair-of-pairs judgements

obtained using those feature sets from phase-randomised (power-only) images were significantly more in agreement with the pair-of-pairs judgements obtained from the original images than those obtained from power-uniformised (phase-only) images. This indicates that the 51 feature sets exploit the power spectrum significantly more than they exploit the phase information. This may explain why these feature sets cannot exploit aperiodic long-range interactions.

On the other hand, classical receptive-field models are, generally, thought to account for local perceptual effects, e.g. border contrast and Mach bands, however, they are not suitable for explaining global perceptual effects, for instance, the perception of the spatially unconnected areas from colour, illusory contours, depth, brightness, motion and texture [Spillmann and Werner, 1996]. Correspondingly, the relationships between the perception and centre-surround antagonism of retinal receptive fields are limited to short-range interactions. On the contrary, the global perceptual effects which usually require neurophysiological mechanisms within the cortex are attributed to long-range interactions. In this situation, if human also utilise long-range interactions for judging the similarity of textures, this will account for the disagreement between the 51 computational feature sets and human observers.

5.6 Conclusions

In this chapter, we evaluated the ability of computational features to estimate perceptual similarity using two sets of ground-truth data: $POPJ_{POP}$ and $POPJ_{ISO}$. It was found that none of the pair-of-pairs judgements estimated using the computational features agreed with either ground-truth datasets more than 66.5% even when calculated over five individual resolutions and one multi-resolution scheme. However, the agreement rate between the two human-derived datasets ($POPJ_{POP}$ and $POPJ_{ISO}$) was 73.9%, which provides at least a limited validation of the two perceptual judgement sets with each other. The difference between these two datasets could be due to either the variability between human populations or the difference between the methods for deriving the two datasets. However, the most significant result is that none of the 51 feature sets performed well against either perceptual pair-of-pairs judgement sets compared with the 73.9% agreement of the two human derived judgement sets.

Since the best performance was obtained using different feature sets cross the six resolutions, we were unable to determine the best individual feature set. In Chapter 3, we divided these feature sets into four categories: filtering-based, statistical, structural and model-based features. However, again, it was not practical to determine which feature category provides the best results overall.

The average performance over the 51 feature sets was computed at five individual resolutions and the multi-resolution scheme along with their 95% confidence intervals. The results of the one-way repeated-measures ANOVA, corrected using Greenhouse-Geisser estimates of sphericity, indicate that the agreement rates obtained using the 51 feature sets were significantly affected by the resolution. However, using the ANOVA, we cannot determine an optimal resolution out of the six resolutions. Nevertheless, compared with the performance obtained at the original resolution (i.e. 1024×1024) the multi-resolution scheme improved the performance of more feature sets than any of the other four resolutions. The post hoc test using the Bonferroni correction also revealed that the agreement rates obtained using multi-resolution are significantly different from those obtained using the resolution of 1024×1024. In addition, the multi-resolution scheme is able to partially encode both short- and long-range interactions because it utilises multiple scales simultaneously. As a result, the multi-resolution scheme was chosen for further investigation.

It could be claimed that the assessment reported in this chapter is based upon an uncommon task: that of judging which pair appears more similar given two pairs of textures. In the next chapter, therefore, another evaluation experiment based on a more popular application, i.e. texture retrieval, is reported.

Chapter 6

Retrieval-Based Evaluation Experiment

6.1 Introduction

In order to augment the results obtained in Chapter 5, another evaluation experiment based on a more common task, namely, texture retrieval, is reported in this chapter. In this experiment, we aim to discover:

(1) whether or not the 51 computational feature sets produce similar results to humans when performing retrieval operations;

(2) whether or not certain feature sets or feature categories (see Chapter 3) generally perform better than their counterparts;

(3) whether or not the multi-resolution approach is better suited to the task than the original resolution (as its advantages over the original resolution have been illustrated in Chapter 5); and

(4) whether or not the results obtained are consistent with those of the pair-of-pairs based evaluation experiments.

More specifically, we first propose a retrieval-based evaluation method in which ranking data, derived by applying the Isomap analysis [Tenenbaum et al., 2000] to the results of a free-grouping experiment, is used as the ground-truth data. The comparison performance measures: G and M ($G, M \in [0, 1]$), defined in Equations (2.10) and (2.12), are used to assess ranking performance. These measures consider not only the number of the relevant textures retrieved but also the rankings of these textures. They therefore provide a more informed measure than the more commonly used metrics, such as Preci-

sion, Recall, Normalised Precision and Normalised Recall (see Section 2.4.2). The experiment is conducted at five different resolutions and in addition we present results from using a multi-resolution scheme. The retrieval set sizes N are set at 10, 20, 40 and 60.

The rest of this chapter is organised as follows: the retrieval-based evaluation method is introduced in Section 6.2. Section 6.3 then reports the overall performance of the computational feature sets on the texture retrieval task. Furthermore, Section 6.4 investigates the effect of the resolution on the performance of the feature sets. Section 6.5 illustrates the performance of the feature sets obtained using the multi-resolution scheme. We also discuss the relationship between the results of this experiment and those obtained in the pair-of-pairs based experiments in Section 6.6. Finally, we draw our conclusions in Section 6.7.

6.2 Retrieval-Based Evaluation Method

Perceptual rankings obtained from human observers have been used to evaluate the performance of search engines [Bar-Ilan et al., 2007] [Metaxas et al., 2009] [Hariri, 2011]. Inspired by this, we introduce a second evaluation method as a complement to the pair-of-pairs based evaluation framework in order to obtain results on a different task (retrieval). This method compares the top N rankings sorted by humans with the top N rankings sorted using computational features. It allows the similarity of different numbers of textures and one query texture to be examined by changing the size of N.

Compared with the other performance measures surveyed in Section 2.4, G and M measures can not only compare two identical (i.e. complete) rankings but also two non-identical (i.e. partial) rankings. Most importantly, both measures utilise actual ranks of the input. Thus, the two measures are suitable for measuring the consistency between a computational ranking and a perceptual ranking. In this evaluation method, the 8D-ISO similarity matrix (see Section 4.2.3) is used as the source of the ground-truth data. The approach for computing texture similarity introduced in Section 4.3 is also used in this method. Each texture is considered in turn as a query texture and the other 333 textures are sorted in descending order of the similarity in the 8D-ISO similarity matrix to provide a ranked list. The evaluation on computational texture rankings against their perceptual counterparts differs from that described in Section 4.4. The evaluation process is conducted as follows:

1) For a computational feature set, $f \in$ {51 computational feature sets}; a resolution, $r \in$ {1024×1024, 512×512, 256×256, 128×128, 64×64 and multi-resolution}; and a value of the retrieval set size, $N \in$ {10, 20, 40, 60}; do:
   i. For each query texture, $q \in$ {334 textures in the *Pertex* database};
      a. Determine "ground-truth" retrieval set by using 8D-ISO to obtain the ranked list (R_{ISO}) of the first *N* textures;
      b. Determine retrieval set by using the *feature set*, *f*, to derive the ranked list (R_f) of the first *N* textures;
      c. Note that the query texture *q* is excluded from the retrieval;
      d. Compute the *G* and *M* measures for comparing the ranked lists: R_{ISO} and R_f;
   ii. Calculate "average *G*" and "average *M*" over all 334 queries.
2) Repeat for all values of *f*, *r* and *N*.

6.3 Overall Performance

The evaluation experiment was conducted using the method introduced in the previous section. Table 6.1 reports the best *G* and *M* performances for different retrieval set sizes and different resolutions (see Figures D.1 and D.2 in Appendix D for more details). When compared to 8D-ISO, the most significant result is that none of the 51 feature sets did well, with the best performing ones providing average *G* and *M* measures of 0.41 and 0.27.

Furthermore, it can be seen that it is not practical to determine the "best" feature set because the best performance is dependent on the resolution. Similarly, the best feature category can also not be decided.

N	Measure	Resolution					
		1024×1024	**512×512**	**256×256**	**128×128**	**64×64**	**Multi-**
10	***G***	VZ-NEIGHBORHOOD	VZ-MRF	VZ-MRF	LBPBASIC	LBPBASIC	LBPHF
		0.21	0.21	0.20	0.20	0.16	0.23
	M	VZ-NEIGHBORHOOD	VZ-MRF	VZ-MRF	LBPBASIC	LBPBASIC	LBPBASIC
		0.19	0.20	0.19	0.18	0.13	0.20
20	***G***	VZ-NEIGHBORHOOD	VZ-MRF	MRSAR	LBPBASIC	LBPBASIC	MRSAR
		0.25	0.25	0.24	0.24	0.20	0.28
	M	VZ-NEIGHBORHOOD	VZ-MRF	VZ-MRF	LBPBASIC	LBPBASIC	LBPHF
		0.20	0.21	0.20	0.20	0.15	0.22
40	***G***	VZ-NEIGHBORHOOD	VZ-MRF	MRSAR	MRSAR	MRSAR	MRSAR
		0.30	0.30	0.32	0.32	0.28	0.36
	M	VZ-NEIGHBORHOOD	VZ-MRF	VZ-MRF	LBPBASIC	LBPBASIC	MRSAR
		0.22	0.23	0.22	0.22	0.18	0.25
60	***G***	RING & WEDGE	RFS	MRSAR	MRSAR	MRSAR	MRSAR
		0.35	0.36	0.38	0.38	0.34	0.41
	M	VZ-NEIGHBORHOOD	VZ-MRF	MRSAR	LBPBASIC	MRSAR	MRSAR
		0.24	0.25	0.24	0.24	0.20	0.27

Table 6.1: The best feature sets determined using G and M measures are shown above for retrieval set sizes $N \in \{10, 20, 40 \text{ and } 60\}$ textures are retrieved at six resolutions $r \in \{1024\times1024, 512\times512, 256\times256, 128\times128, 64\times64 \text{ and multi-resolution}\}$.

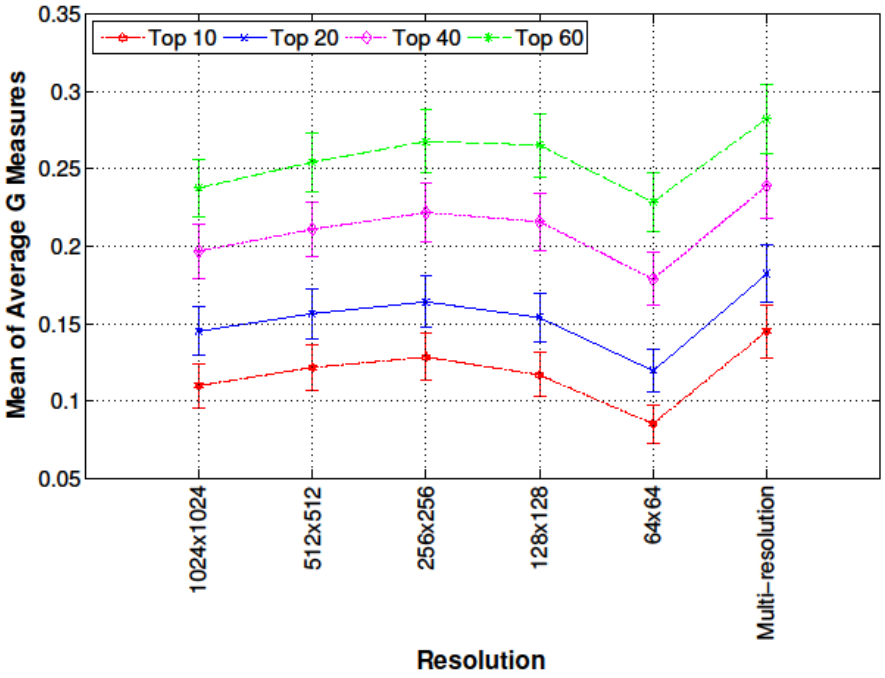
6.4 Effect of the Resolution

Since the significant advantages of the multi-resolution scheme over the original resolution have been observed in Chapter 5, we examine the effect of the resolution on the performance of the feature sets in order to determine whether or not it is the case when the retrieval-based evaluation is conducted. We first apply an ANOVA (Analysis of Variance) to test the significance of the effect of the resolution on the *G* measure and the *M* measure. Then, the effect of using the multi-resolution scheme is analysed empirically.

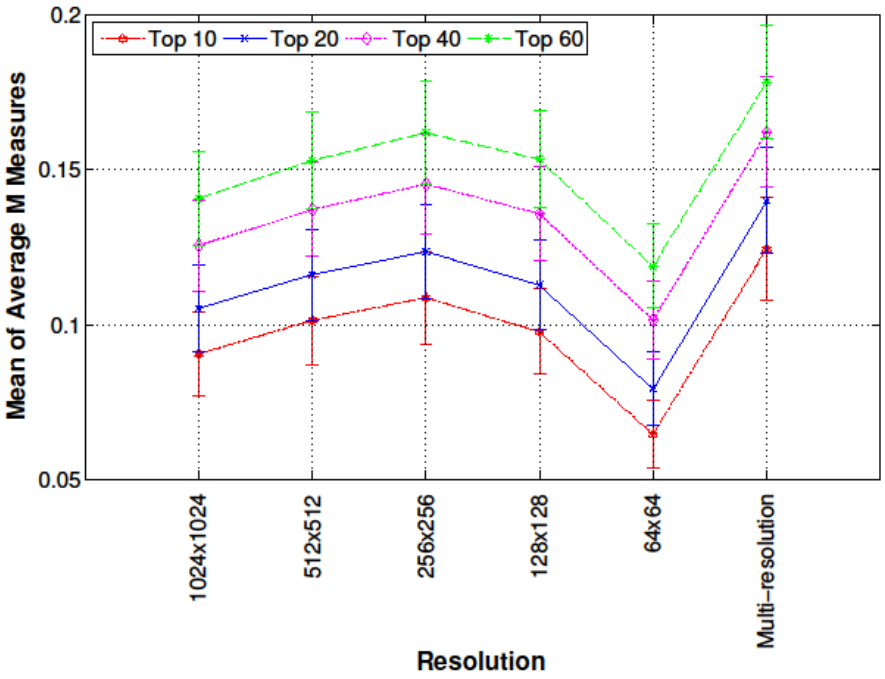
6.4.1 Significance Tests Using ANOVA

Given that the 51 feature sets are considered as a population, the *G* or *M* measure obtained using these can be considered as the dependent variable while the resolution and the retrieval set size *N* are regarded as the independent variable of a factorial repeated-measures ANOVA [Field, 2009]. Since we are not interested in the interactions between *G* and *M* measures, we performed two factorial repeated-measures ANOVAs on *G* and *M* measures separately. However, the family-wise error needs to be controlled by adjusting the level of significance for each individual ANOVA in order to ensure that the

overall Type I error rate (∂) throughout all ANOVAs stays at 0.05. Hence, ∂ = 0.025 is utilised for Bonferroni correction [Abdi, 2007]. Figures 6.1 (a) and (b) show the means and 97.5% confidence intervals of the average *G* and average *M* measures derived using five individual resolutions and the multi-resolution scheme over the 51 feature sets.



(a)



(b)

Figure 6.1: Means and 97.5% confidence intervals (error bars) of the average G (a) and average M (b) measures obtained using five individual resolutions and the multi-resolution scheme over the 51 feature sets.

ANOVA on the *G* Measure

The results of Mauchly's test [Mauchly, 1940] [Field, 2009] show that the assumption of sphericity was violated for the main effects of the resolution, $\chi^2(14) = 186.38$, $p < 0.05$, the retrieval size, $\chi^2(5) = 367.91$, $p < 0.05$, and the interaction between the resolution and the retrieval size, $\chi^2(119) = 1169.39$, $p < 0.05$. Degrees of freedom were therefore corrected using Greenhouse-Geisser estimates of sphericity [Greenhouse and Geisser, 1959] ($\varepsilon = 0.46$ and 0.34 for the main effects of the resolution and the retrieval size, and $\varepsilon = 0.20$ for the interaction effect between the resolution and the retrieval size). We, therefore, report the three effects derived from this analysis as below:

(1) The results show a significant main effect of the resolution on the *G* measure, $F(2.28, 114.03) = 41.87$, $p < 0.025$. Contrasts revealed that the *G* measures obtained using resolutions of 1024×1024 ($F(1, 50) = 93.31$, $r = 0.81$), 512×512 ($F(1, 50) = 44.15$, $r = 0.68$), 256×256 ($F(1, 50) = 21.72$, $r = 0.55$), 128×128 ($F(1, 50) = 27.52$, $r = 0.60$) and 64×64 ($F(1, 50) = 92.20$, $r = 0.81$) were significantly lower than those obtained using the multi-resolution scheme;

(2) The significant main effect of the retrieval size on the *G* measure is also found, $F(1.03, 51.41) = 1945.95$, $p < 0.025$. Contrasts revealed that the *G* measures obtained when top 10 ($F(1, 50) = 2000.40$, $r = 0.99$), top 20 ($F(1, 50) = 2194.28$, $r = 0.99$) and top 40 ($F(1, 50) = 2645.83$, $r = 0.99$) textures were retrieved were significantly lower than those obtained when top 60 textures were retrieved; and

(3) There was a significant interaction effect between the resolution and the retrieval size, $F(3.03, 151.43) = 17.40$, $p < 0.025$. It is indicated that the retrieval size generated different effects on *G* measures with the changing of the resolution.

ANOVA on the *M* Measure

Mauchly's test [Mauchly, 1940] [Field, 2009] was performed to examine the sphericity of the *M* measure data. Its results show that the assumption of sphericity was violated for the main effects of the resolution, $\chi^2(14) = 169.54$, $p < 0.05$, the retrieval size, $\chi^2(5) = 450.20$, $p < 0.05$, and the interaction between the resolution and the retrieval size, $\chi^2(119) = 1562.49$, $p < 0.05$. Thus, degrees of freedom were corrected using Greenhouse-Geisser estimates of sphericity [Greenhouse and Geisser, 1959] ($\varepsilon = 0.45$ and 0.34 for the main effects of the resolution and the retrieval size, and $\varepsilon = 0.22$ for the in-

teraction effect between the resolution and the retrieval size). The three effects derived from this analysis are described as follows:

(1) The significant main effect of the resolution on the *M* measure is observed, $F(2.25, 112.46) = 51.77$, $p < 0.025$. Contrasts show that the *M* measures obtained using resolutions of 1024×1024 ($F(1, 50) = 91.56$, $r = 0.80$), 512×512 ($F(1, 50) = 49.80$, $r = 0.71$), 256×256 ($F(1, 50) = 27.61$, $r = 0.60$), 128×128 ($F(1, 50) = 42.01$, $r = 0.68$) and 64×64 ($F(1, 50) = 122.89$, $r = 0.84$) were significantly lower than those obtained using the multi-resolution scheme;

(2) The main effect of the retrieval size on the *M* measure is also significant, $F(1.01, 50.62) = 1869.54$, $p < 0.025$. Contrasts revealed that the *M* measures obtained when top 10 ($F(1, 50) = 1898.48$, $r = 0.99$), top 20 ($F(1, 50) = 1991.85$, $r = 0.99$) and top 40 ($F(1, 50) = 2276.91$, $r = 0.99$) textures were retrieved were significantly lower than those obtained when top 60 textures were retrieved; and

(3) A significant interaction effect between the resolution and the retrieval size is found, $F(3.29, 164.45) = 7.48$, $p < 0.025$.

Summary of the Significance Tests

No matter whether the *G* or the *M* measure is used, the main effect of the resolution on both measures is significant. The *G* and *M* measures obtained using the multi-resolution scheme were significantly higher than those obtained using the other five resolutions. The significant main effect of the retrieval size on both measures is also observed. In addition, a significant interaction effect between the resolution and the retrieval size is found. However, this section's aims are to investigate the effect of the resolution on the performance obtained using computational feature sets. In the next subsection, we, therefore, investigate the effect of the multi-resolution scheme on the performance of the 51 feature sets using the same approach as that used in Sections 5.2.3 and 5.3.3.

6.4.2 Examining the Number of Feature Sets that Can be Enhanced Using Multi-resolution

The average *G* and average *M* measures obtained using the 51 feature sets at five resolutions were compared with those obtained at the original resolution (1024×1024) when

top 10, 20, 40 and 60 textures were retrieved. Table 6.2 reports these results in detail. In fact, the performance of more feature sets (≥ 46) out of the 51 feature sets is enhanced using the multi-resolution scheme than the other four resolutions. Particularly, Table 6.3 lists the exceptions produced using five feature sets when multi-resolution is considered. However, the difference between the performance obtained using these feature sets except JSCW [Portilla and Simoncelli, 2000] and VZ-MR8 [Varma and Zisserman, 2005] at the resolution of 1024×1024 and multi-resolution is tiny. Thus, the advantages of the multi-resolution scheme over the other resolutions are also found empirically.

N	**Measure**	**Resolution**				
		512×512	**256×256**	**128×128**	**64×64**	**Multi**
10	***G***	42	40	30	14	***47***
	M	38	44	29	11	***46***
20	***G***	42	40	31	14	***47***
	M	41	43	31	12	***46***
40	***G***	44	41	34	20	***48***
	M	44	43	32	13	***46***
60	***G***	45	44	38	23	***49***
	M	44	43	32	16	***46***

Table 6.2: The numbers of the feature sets whose G and M measures were improved using the other five resolutions ("Multi" denotes the multi-resolution scheme) compared with the measures obtained using the same feature sets at the resolution of 1024×1024, when top 10, 20, 40 and 60 textures were retrieved. In each row, the bold italic font indicates the largest number of the feature sets.

N	**Measure**	**Resolution**	**Method**				
			FRACTALDIMENSION	**JSCW**	**SVR**	**TT**	**VZ-MR8**
10	***G***	**1024**	0.09	0.22	0.07	N/A	0.22
		Multi	0.08	0.19	0.07	N/A	0.22
	M	**1024**	0.05	0.14	0.03	0.08	0.18
		Multi	0.04	0.12	0.03	0.08	0.17
20	***G***	**1024**	0.06	0.18	0.04	N/A	0.21
		Multi	0.06	0.16	0.04	N/A	0.20
	M	**1024**	0.04	0.13	0.02	0.06	0.17
		Multi	0.03	0.11	0.02	0.06	0.17
40	***G***	**1024**	0.10	0.25	0.08	N/A	N/A
		Multi	0.09	0.22	0.08	N/A	N/A
	M	**1024**	0.05	0.16	0.03	0.09	0.19
		Multi	0.05	0.14	0.03	0.09	0.18
60	***G***	**1024**	0.14	0.29	N/A	N/A	N/A
		Multi	0.13	0.27	N/A	N/A	N/A
	M	**1024**	0.07	0.18	0.05	0.12	0.20
		Multi	0.06	0.15	0.05	0.12	0.20

Table 6.3: The G and M measures obtained using five sets of computational features at the resolution of 1024×1024 and multi-resolution ("Multi"), when top 10, 20, 40 and 60 textures were retrieved.

6.4.3 Summary of Effect of the Resolution

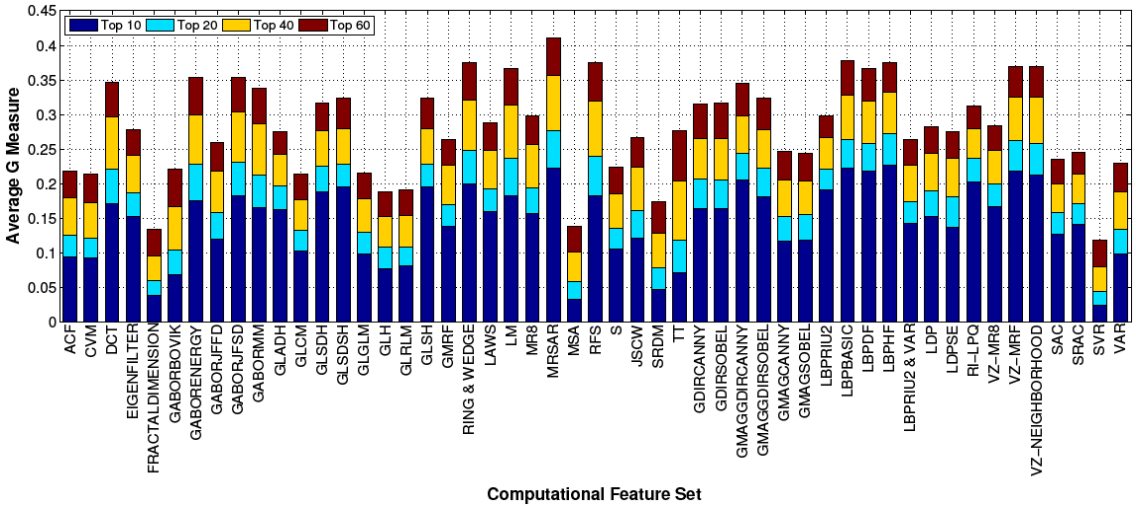
In this section, the advantages of the multi-resolution scheme over the other resolutions, especially, the original resolution (1024×1024), have been observed statistically and empirically. As a result, the multi-resolution scheme was chosen for further investigation.

6.5 Detailed Examination of Feature Performance for the Multi-resolution Case

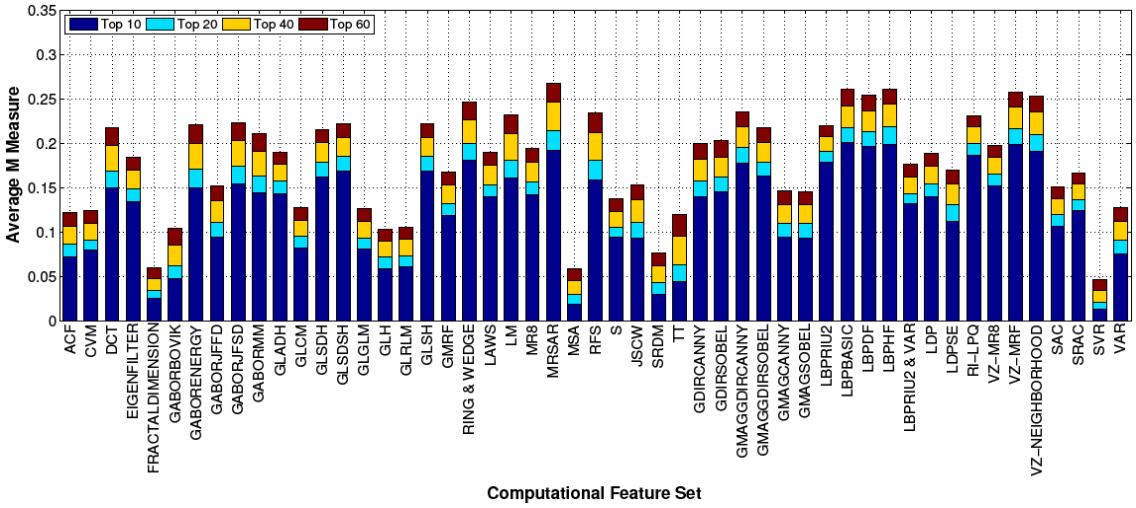
This section first reports the performance obtained using each of the 51 feature sets when the multi-resolution scheme is considered. The textures involved in the extreme cases: "failed" and "relatively-successful" are then examined.

6.5.1 Average *G* and Average *M* Measures

Figure 6.2 presents the average *G* (a) and average *M* (b) measures derived using the 51 feature sets when the multi-resolution scheme is considered. The highest average *G* and *M* measures of 0.41 and 0.27 were obtained using MRSAR [Mao and Jain, 1992] when 60 textures are retrieved. Specifically, at the same conditions, the average *G* measure is greater than 0.41/0.43 when over 14/15 relevant textures are in the same orders and exceeds 0.40/0.42 when 15/16 textures are in the opposite orders. The average *G* of 0.41 suggests that less than 16 textures are relevant no matter what orders these are in, for some query textures. On the other hand, the average *M* measure is over 0.27 when one or more relevant textures are in the same order or at least 22 relevant textures are in the opposite order. In addition, if we ignore the difference between computational and perceptual ranked lists of relevant textures, the average proportion of the number of the relevant textures with respect to 60, i.e. average Precision (see Equation (2.2)), is 0.48. It means that only 29 (≈ 0.48×60) retrieved textures are relevant when 60 textures are retrieved. To summarise, even when considering only the best performance, (1) no more than one half of retrieved textures are relevant, and (2) the orders of the relevant textures between computational rankings and perceptual rankings are different. Therefore, even the best performance obtained here is disappointing.



(a)



(b)

Figure 6.2: Stacked bar charts of the average G (a) and M (b) measures obtained using 51 feature sets when multi-resolution is only considered compared to 8D-ISO. Each bar shows four different, colour-coded results for four retrieval set sizes $N \in \{10, 20, 40 \text{ and } 60\}$.

6.5.2 "Failed" and "Relatively-Successful" Textures

The textures associated with two extreme cases that we term: "failed" and "relatively-successful" can provide us with more insights. We consider "failed" textures to be the textures that cannot be accurately retrieved using the majority of the 51 feature sets. The "relatively-successful" textures are defined as the textures that can be retrieved using the majority of the 51 feature sets better than the other textures. Since the multi-

resolution scheme was regarded as the optimal one, it is used in this investigation. In addition, only $N = 10$ retrievals are considered.

It is found from Equations (2.10) and (2.12) that both *G* and *M* arrive at 0 when there is no relevant retrieved texture after a retrieval operation is performed for one query texture. In this case, the query texture is termed as a "failed" texture. In addition, if the *G* or *M* measure is small, the current query texture can also be regarded as "failed". The worst average *G* and average *M* measures (0.02 and 0.01) obtained using multi-resolution when top 10 textures are retrieved are used to threshold all *G* and *M* measures. For each feature set, if the *G*/*M* measure obtained on one query is greater than 0.02/0.01, this texture will be left out. The occurrence frequencies of all remaining query textures obtained using the 51 feature sets are accumulated vs. 334 textures. In essence, the occurrence frequencies are the numbers of feature sets. $T_n = 30$ is used to threshold the occurrence frequencies. After the thresholding operation is conducted, the textures whose occurrence frequencies are over T_n are taken as "failed" textures. Figures 6.3 (a) and (b) present top 15 "failed" textures for no less than 43 feature sets, selected using the *G* and *M* measures.

The "relatively-successful" textures, were also selected by thresholding *G*/*M* values. The best average *G* and average *M* (0.23 and 0.20) obtained when the top 10 textures are retrieved using the multi-resolution approach are used to threshold all *G* and *M* obtained using the 51 feature sets in the same conditions. For each feature set, if the *G*/*M* measure obtained on one query texture is less than 0.23/0.20, this texture will be left out. Then we accumulate the occurrence frequencies of 334 textures from the remaining textures obtained using the 51 feature sets. Again, $T_n = 30$ is used to threshold the occurrence frequencies. The textures whose occurrence frequencies are over T_n are considered as "relatively-successful" textures. Figures 6.4 (a) and (b) display top 15 "relatively-successful" textures for at least 37 feature sets, selected using the *G* and *M* measures.

It can be observed from Figure 6.3 that the majority of the 51 sets of computational features are unable to accurately capture perceptual rankings between aperiodic textures (e.g. "040", "312", "148", "131" and "034"), although some of these textures are also well-ordered (e.g. "148" and "034"). However, these feature sets are able to better encode perceptual rankings between periodic (regular) or nearly-periodic textures (see Figure 6.4, e.g. "168", "171", "172", "121", "061" and "308"). In fact, no matter whether a texture is periodic or nearly-periodic, it shows strong periodicity which is normally

associated with power spectra. In contrast, aperiodic (structural or stochastic) textures are believed to be encoded by the phase information [Oppenheim and Lim, 1991]. This indicates that the 51 feature sets exploit power spectra more than phase spectra.

Figures 6.5 and 6.6 list the top 10 retrieval textures: (a) ranked by the human observers in the free-grouping experiments and (b) retrieved using GLH [Mirmehdi et al., 2009] and GDIRSOBEL [Ojala et al., 1996], of two "failed" textures "003" and "131" (see Figure 6.3) selected using both *G* and *M* measures. Furthermore, Figures 6.7 and 6.8 display the top 10 retrieval textures: (a) ranked by the human observers in free-grouping and (b) retrieved using GMAGGDIRCANNY [Ojala et al., 1996] and LBPBASIC [Ahonen and Pietikäinen, 2009], of two "relatively-successful" textures "047" and "121" (see Figure 6.4) selected using the *M* measure.

From Figures 6.5 (a) to 6.8 (a), it can be seen that humans are able to rank either aperiodic (e.g. structural or stochastic textures) or periodic (regular) textures. However, none of the retrieval results obtained using computational features for the "failed" or "relatively-successful" textures are highly consistent with those perceptual rankings. Even the "optimal" retrieval results for two "relatively-successful" textures are not completely satisfactory.

6.5.3 Summary for the Multi-resolution Case

When the multi-resolution scheme was used to retrieve 60 textures, the best average *G* and *M* measures obtained were 0.41 and 0.27, respectively. However, less than one half of retrieved textures are relevant and the rankings of the relevant textures differ in the computational and perceptual based retrievals. In addition, we also examined two extreme cases: "failed" and "relatively-successful". Even when the "relatively-successful" textures are considered, their retrieved texture rankings are not satisfying (only a small part of textures are relevant) compared with the texture rankings sorted by humans.

As we discussed in Section 5.5, the 51 computational feature sets do not exploit aperiodic long-range interactions. However, humans have been found to be able to utilise these in other tasks [Field et al., 1993] [Polat and Sagi, 1994] [Spillmann and Werner, 1996]. We, therefore, hypothesise that humans can employ aperiodic long-range interactions when judging the similarity of textures. If this is the case, it may account for the disagreement between computational and perceptual rankings.

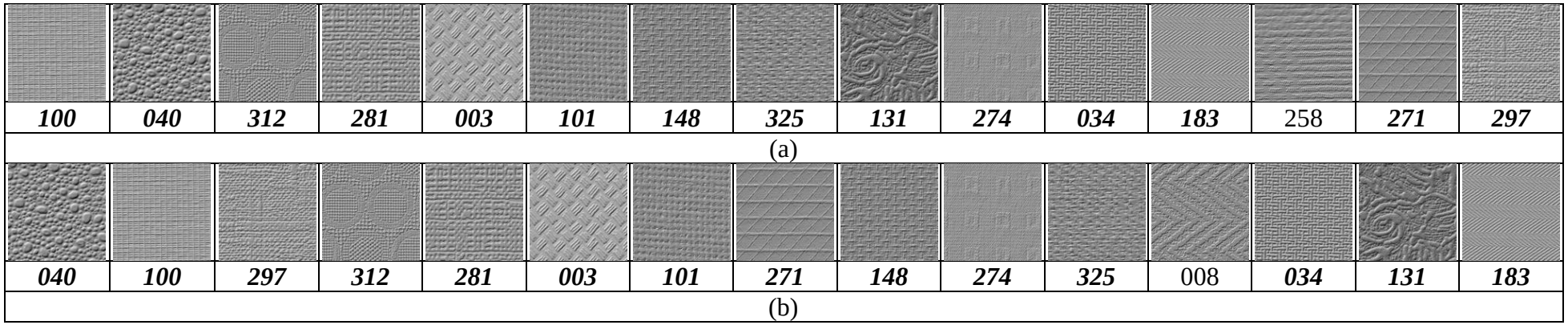


Figure 6.3: Top 15 "failed" textures (central quarters are shown only) for no less than 43 sets of computational features, chosen using the G (a) and M (b) measures respectively. The bold italic font means the intersection of the two groups of textures.

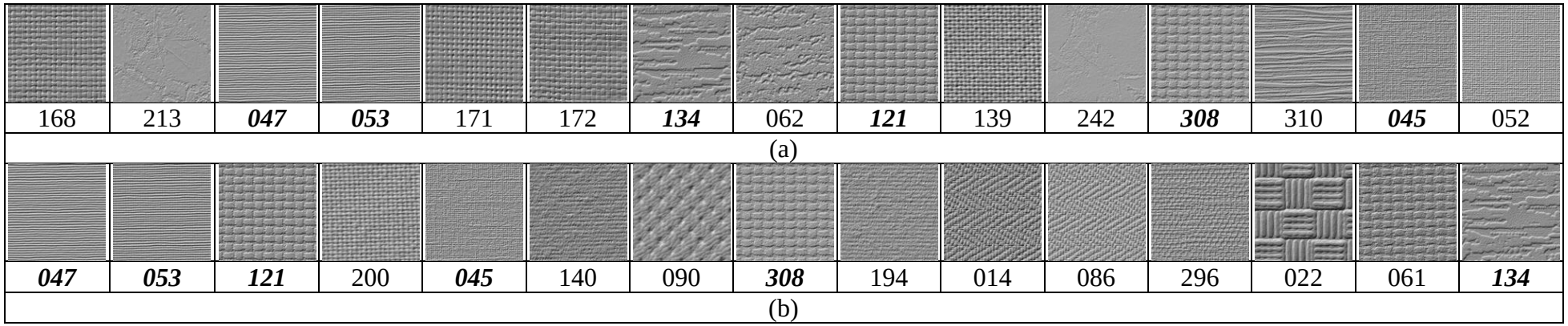
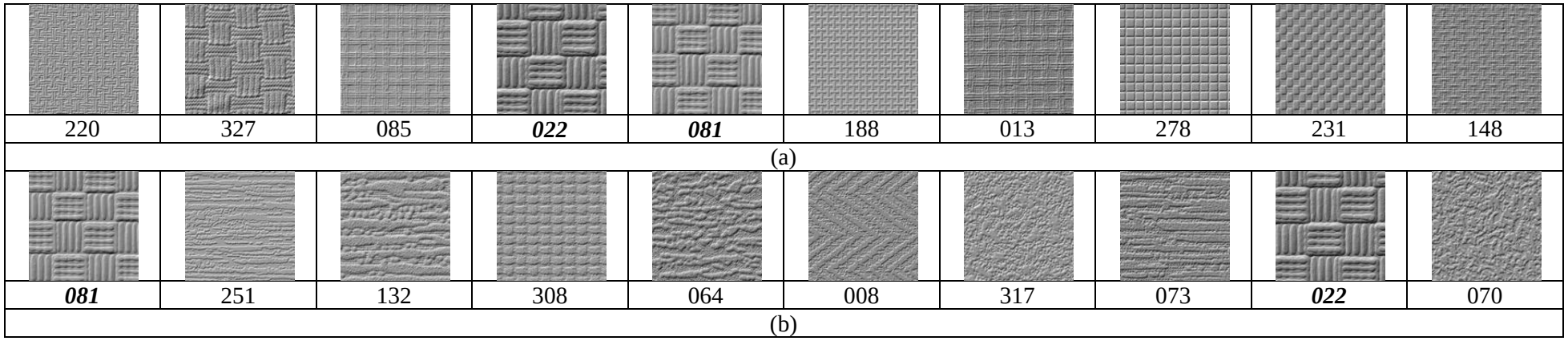
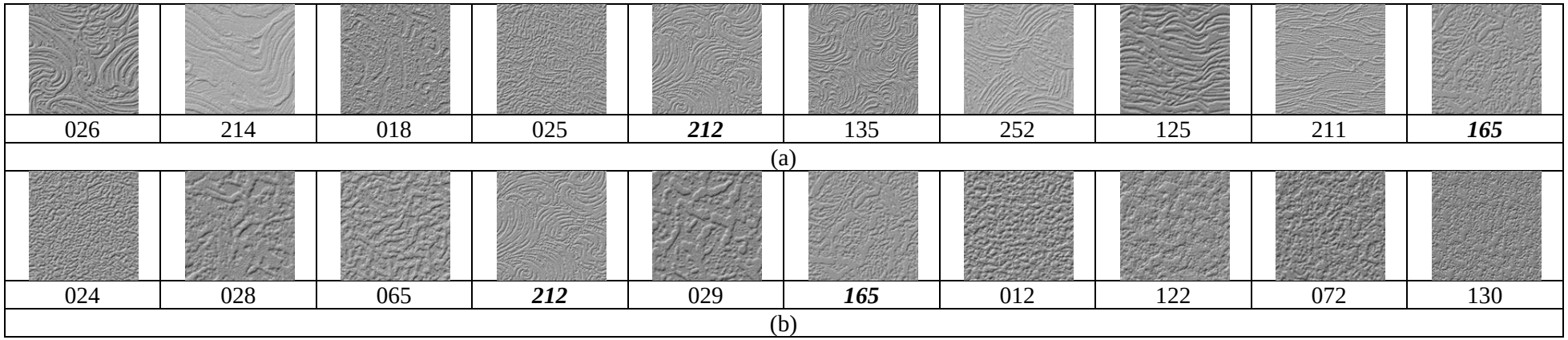


Figure 6.4: Top 15 "relatively-successful" textures (central quarters are shown only) for at least 37 sets of computational features, chosen using the G (a) and M (b) measures respectively. The bold italic font means the intersection of the two groups of textures.



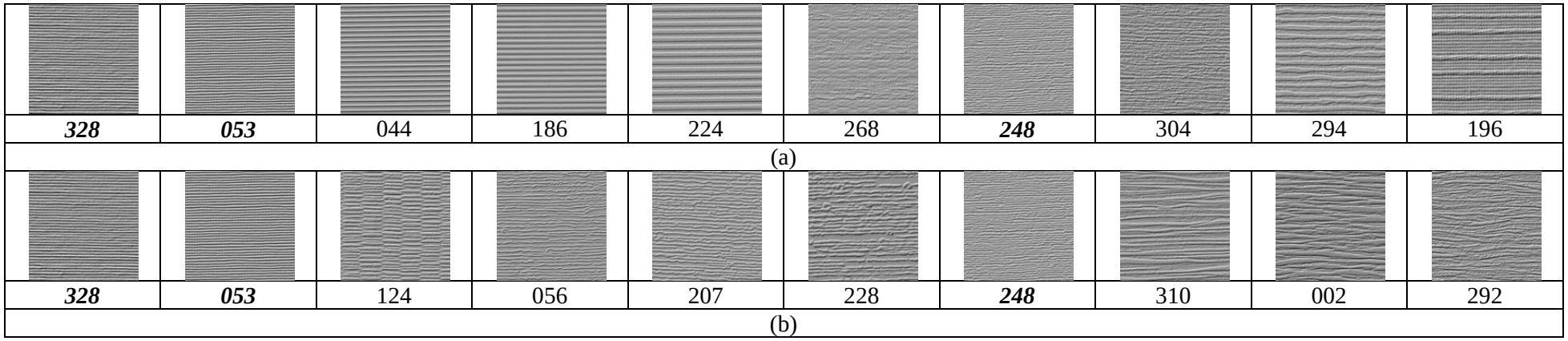
220	327	085	***022***	***081***	188	013	278	231	148
(a)									
081	251	132	308	064	008	317	073	***022***	070
(b)									

Figure 6.5: Top 10 textures (central quarters are shown only): (a) ranked by the observers in the free-grouping experiments and (b) retrieved using GLH of the "failed" texture "003" (see Figure 6.3) chosen using both G (0.15) and M (0.06). The bold and italic font suggests relevant textures.



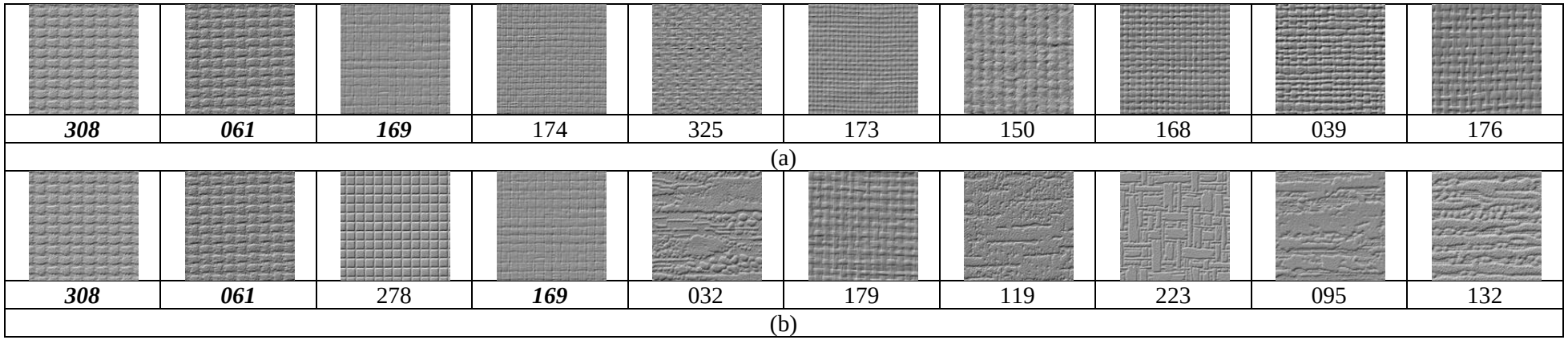
026	214	018	025	***212***	135	252	125	211	***165***
(a)									
024	028	065	***212***	029	***165***	012	122	072	130
(b)									

Figure 6.6: Top 10 textures (central quarters are shown only): (a) ranked by the observers in the free-grouping experiments and (b) retrieved using GDIRSOBEL of the "failed" texture "131" (see Figure 6.3) chosen using G (0.13) and M (0.06). The bold and italic font suggests relevant textures.



328	***053***	044	186	224	268	***248***	304	294	196
(a)									
328	***053***	124	056	207	228	***248***	310	002	292
(b)									

Figure 6.7: Top 10 textures (central quarters are shown only): (a) ranked by the observers in the free-grouping and (b) retrieved using GMAGGDIR-CANNY of the "relatively-successful" texture "047" (see Figure 6.4) chosen using M (0.68). The bold and italic font suggests relevant textures.



308	***061***	***169***	174	325	173	150	168	039	176
(a)									
308	***061***	278	***169***	032	179	119	223	095	132
(b)									

Figure 6.8: Top 10 textures (central quarters are shown only): (a) ranked by the human observers in the free-grouping and (b) retrieved using LBPBASIC of the "relatively-successful" texture "121" (see Figure 6.4) chosen using M (0.73). The bold and italic font suggests relevant textures.

6.6 Relationship to the Pair-of-Pairs Evaluation

In this experiment, none of the 51 feature sets performed well when compared against perceptual texture rankings. This further supports the conclusions drawn in Chapter 5. However, there are also some differences between the results obtained in the two evaluations. For example, VZ-NEIGHBORHOOD and VZ-MRF [Varma and Zisserman, 2009] performed similarly to the mean in two pair-of-pairs based evaluation experiments, while they outperformed the majority of the 51 feature sets in the retrieval-based evaluation experiment.

Given that the 334 textures in *Pertex* can be divided into 14 clusters (see Section 4.2.3), it is interesting to examine both the intra-cluster (within a cluster) and inter-cluster (between clusters) perceptual similarities among the 1000 pair-of-pairs in the original pair-of-pairs experiment. However, only the top N (10, 20, 40 and 60) most similar textures rather than all 334 textures are generally retrieved in the retrieval-based evaluation experiment. As the retrieval-based evaluation experiment only considers the consistency of two top N rankings, it is likely that intra-cluster perceptual similarity is tested more than inter-cluster perceptual similarity. The smaller N is, the more the intra-cluster similarity is examined. Consequently, when small numbers of textures are retrieved, the retrieval-based evaluation experiment mainly examines the ability of the computational feature sets to estimate intra-cluster perceptual texture similarity. This should account for the slight difference between the performances of the 51 feature sets obtained in the pair-of-pairs based and the retrieval-based evaluation experiments.

6.7 Conclusions

In this chapter, we first proposed a retrieval-based evaluation method and then reported the results of the evaluation experiment that used the same computational feature sets as those examined in Chapter 5 using a pair-of-pairs approach. The top 10, 20, 40 and 60 textures were retrieved, giving the highest average G and M measures of 0.41 and 0.27 compared to 8D-ISO. In this situation, no more than one half of retrieved textures are relevant and the orders of the relevant textures between computational rankings and perceptual rankings are different. These results are not as good as the best average G and M measures: 0.434 and 0.293 reported by Bar-Ilan et al. [2007]. The retrieved tex-

tures are also not consistent with the textures retrieved by humans (no matter whether their rankings are considered or not), even if one "relatively-successful" texture is queried using its corresponding "best" feature set. Thus, we conclude that the computational feature sets do not perform well against human observers on ranking the textures in the *Pertex* database.

Furthermore, it is unclear as to which is the best feature set overall because it varies with the resolution. In this situation, it is also unclear as to what the best feature category is.

The results of two factorial repeated-measures ANOVA corrected using Greenhouse-Geisser estimates of sphericity show that the *G* or *M* measures obtained using the 51 feature sets were significantly affected by the resolution. The *G* or *M* measures obtained using the multi-resolution scheme were significantly higher than those obtained using the other five resolutions. In addition, the advantages of the multi-resolution scheme over the other resolutions when being compared with the original resolution (i.e. 1024×1024) were also observed empirically. Thus, the multi-resolution scheme was chosen for further investigation.

Beyond the above, the results derived in this experiment are slightly different from those obtained in the pair-of-pairs based evaluation experiments. We attribute it to the difference between the two experimental setups.

As we discussed in Section 5.5, the 51 computational feature sets cannot exploit aperiodic long-range interactions. However, humans have been found to be able to use long-range interactions in other tasks [Field et al., 1993] [Polat and Sagi, 1994] [Spillmann and Werner, 1996]. We, therefore, hypothesise that humans exploit long-range interactions for judging the similarity of textures and that this is why the computational features do not perform well when tested against human data.

Chapter 7

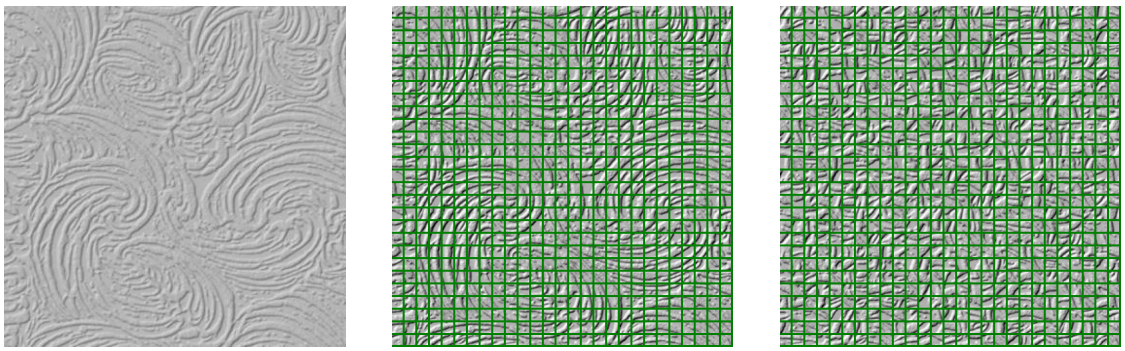
The Importance of Long-Range Interactions to Perceptual Texture Similarity

7.1 Introduction

The goal of this thesis is to investigate and develop a perceptually-motivated computational measure of texture similarity. It is well-known that humans exploit long-range visual interactions for other tasks [Field et al., 1993] [Polat and Sagi, 1994] [Spillmann and Werner, 1996], however, in Chapter 3, we have shown that very few computational features exploit aperiodic long-range interactions in imagery. We therefore hypothesise that the poor performance of the 51 computational feature sets reported in Chapters 5 and 6 could be that they do not exploit long-range aperiodic information.

Unfortunately, it is difficult to test this hypothesis directly by "adding" long-range interactions to textures because such actions invariably change local characteristics, i.e. short-range interactions. It is not practical, therefore, to design an experiment to investigate the effect of the addition of long-range interactions. As an alternative, we can remove (or at least reduce) long-range interactions by randomising the position of texture patches. However, the boundary between two randomised patches may introduce new short-range interactions which affect human judgements. This is likely to be the case even if we use texture synthesis techniques [Efros and Freeman, 2001] [Nealen and Alexa, 2003] to make the "boundary area" change gradually. In order to remove (or at least reduce) long-range interactions while inhibiting the perceived changes in short-range interactions, we first overlay the texture image with a grid (see Figure 7.1 (b)) and

then randomise the position of the blocks (see Figure 7.1 (c)). The grid is designed to reduce the visual effect of local discontinuities at the randomised blocked edges.



(a) (b) (c)

Figure 7.1: An original texture image and its non-randomised blocked and randomised blocked images. Both the middle (b) and right (c) images are generated from Texture "026" (a) in Pertex by overlaying, or "blocking" it with a green grid while the blocks in the right image are randomised. Although the grid in (b) obscures a portion of the texture, an observer can still associate it with the original image (a). In other words, the human visual system is able to integrate the information in different blocks in (b). However, the texture (or pattern) in (c) looks different from that in (a) or (b).

In order to test our hypothesis, we performed two modified pair-of-pairs experiments using the 80 "most inconsistent" pairs of pairs (see Figures E.2-E.5), those where the disagreements between the majority of the 51 feature sets and human observers were greatest. Appendix E describes the selection of these 80 pairs of pairs from the 1000 pairs of pairs used in the "original" pair-of-pairs experiment (POP_O) [Clarke et al., 2012]. The two modified pair-of-pairs experiments (POP_N and POP_R) were designed using "non-randomised blocked" images (see Figure 7.1 (b)) and "randomised blocked" images (see Figure 7.1 (c)) respectively. The former was designed to provide a "control" investigating the effect of the "blocking" of images. The latter was used to examine the effect of "removing" or "reducing" long-range interactions. As a result, if human observers in this randomised blocked experiment (POP_R) agree less significantly with the majority of the observers in the original experiment (POP_O) compared with the observers in the non-randomised blocked experiment (POP_N), it suggests that long-range interactions significantly affect humans' judgements of texture similarity.

The rest of this chapter is organised as follows. The two modified pair-of-pairs experiments are described in detail in Section 7.2. In Section 7.3, the experimental results are reported and analysed. Finally, conclusions are drawn in Section 7.4.

7.2 Experimental Design

In this section, two modified pair-of-pairs experiments (POP_N and POP_R) are described that used non-randomised blocked and randomised blocked images respectively. The objective is to investigate whether or not long-range interactions affect humans' judgements of texture similarity.

7.2.1 Hypothesis

Field et al. [1993] used the concept of the "association field" to explain how continuity may be represented by a visual system. It was shown that humans can still recognise an object in one image even though a grid has been imposed on top of it (see Figures 7.1 (a) and (b)). This phenomenon is normally attributed to the effect of long-range interactions [Field et al., 1993]. Inspired by this result, we "blocked" texture images and removed, or at least reduced, the long-range interactions by randomising the position of these blocks (see Figure 7.1 (c)) in POP_R. However, we can still not guarantee that the short-range interactions are not impaired by the "blocking" operation. Thus, we conducted a second (modified) pair-of-pairs experiment (POP_N) using non-randomised blocked images (see Figure 7.1 (b)) in order to provide a "control" investigating the effect of the "blocking" of images. If human observers in the randomised blocked experiment (POP_R) agree with the majority of the observers in the original experiment (POP_O) significantly less than the observers in the non-randomised blocked experiment (POP_N) do, this indicates that long-range interactions affect humans' judgements of texture similarity.

The 51 computational feature sets tested in Chapters 5 and 6 normally do not exploit aperiodic long-range interactions, as we discussed in Chapter 3. Therefore, if human judgements agree more with the results of computational features when we "remove" long-range interactions while retaining short-range interactions, it seems likely that humans do exploit long-range interactions when judging texture similarity. Conversely, it also indicates that short-range interactions are used by humans even if when long-range interactions have been "removed".

7.2.2 Experimental Setup

Tools

Padilla [2008] investigated several display technologies in his research. He concluded that high specification LCD monitors are more suitable for psychophysical experiments than CRT monitors in view of the gamma correction, spatial modulation transfer function and luminance uniformity. Therefore, all stimuli were displayed on a calibrated NEC LCD2090UXi monitor at the resolution of 512×512 in the two modified pair-of-pairs experiments. The monitor has a resolution of 1600×1200 and pixel dimensions are 0.255mm×0.255mm (i.e. 100 dots per inch (dpi)). Thus, all stimuli were 130.56mm×130.56mm when displayed on the monitor. In addition, the monitor was linearly calibrated to gamma = 1, with a Gretag-MacBeth Eye-One, with a maximum luminance of 120$\mathrm{cd/m^2}$. In this case, the stimulus images look like they are lit by similar lighting conditions to those in a bright room.

Environment

Throughout the two modified pair-of-pairs experiments, observers were required to sit in front of the monitor. The responses to the stimuli displayed on the monitor were made by using the "Left Arrow (←)" or "Right Arrow (→)" key. The distance between the monitor and the observers was set to approximately 50 cm, providing an angular resolution of around 17 cycles per degree (cpd). As a result, the stimuli images subtended an angle of 14.89° in the vertical direction. The eyes of the observers were located approximately along the line of the centre of the screen. Both experiments were carried out in a dark room with opaque, matte, black curtains and matte walls without apparent specular reflections.

7.2.3 Stimuli

Restricted by the resolution of the monitor, 334 original texture images were first downsampled from the resolution of 1024×1024 to 512×512 by using Gaussian pyramid [Simoncelli, 2009]. Each downsampled image was then normalised to have an average intensity of 0 and standard deviation of 1 in order to remove the influence of 1st- and 2nd-order grey level properties. All normalised images were scaled to the range of [0,

255] by the global maximum and minimum grey level values throughout all 334 normalised images. In this chapter, we use "original texture images" to refer to these scaled images (see Figure 7.1 (a)).

In order to prevent human observers from using long-range interactions, all original texture images were first blocked with a green grid (see Figure 7.1 (b)) and the position of the blocks in the image were then randomised (see Figure 7.1 (c)). The reasons for using green rather than the other psychological primary colours [Natural Colour System] are that (1) it is gentler and more comfortable for the human eye and impairs human perception less; and (2) it makes the grid easy to distinguish from the grey texture. The thickness of the grid was set as three pixels. In addition, the size of the blocks was 19×19 pixels which is the largest size (see Table 3.2) of the neighbourhoods exploited by the 51 computational feature sets (excluding filtering-based features, see Section 3.8). It should be noted that different sizes of blocks could produce different effects on the perception of blocked images. This issue is also depedent on the scale of textures. However, to the best of our knowledge, there is no accurate measure for texture scale in the literature. Note that the texture scale issue is considered outwith of the scope of this research. Therefore, we ignore the effect of this issue in this chapter.

In one modified pair-of-pairs experiment (POP_R), the "randomised blocked" texture images (see Figure 7.1 (c)) were used to obtain the pair-of-pairs similarity judgements of humans. Furthermore, in another modified pair-of-pairs experiment (POP_N), the "non-randomised blocked" texture images (see Figure 7.1 (b)) were used as the "control" to determine the effect of blocking (superimposing the grid onto images) on perceptual similarity judgements. Note that the grid is provided in order to reduce the effect of local discontinuities at randomised blocked edges. All other conditions were kept the same as those in the original pair-of-pairs experiment (POP_O) [Clarke et al., 2012].

7.2.4 Observers

Ten participants with normal or corrected-to-normal vision were used, including four PhD students of the Texture Lab at Heriot-Watt University (regarded as naïve) and six other students from Heriot-Watt University (all naïve). None of the 10 participants had attended the original pair-of-pairs experiment. All participants signed a consent form before they performed the two experiments. Each participant was paid a 5 GBP Amazon voucher after they completed the two experiments.

7.2.5 Procedure

In the two modified pair-of-pairs experiments (POP_N and POP_R), only the 80 most inconsistent pairs of pairs selected in Appendix E were considered because: (1) we were interested in the commonality between the failures for the 51 computational feature sets; and (2) the experiments are time-consuming and 80 (8% of the 1000) trials (i.e. pairs of pairs) was a good trade-off between efficiency and accuracy.

The 10 participants were used in both of the modified pair-of-pairs experiments. The randomised blocked experiment was conducted at least one week earlier than the non-randomised blocked experiment in order to alleviate learning/training effects. The 80 trials were shown in random order to each participant in each experiment. Throughout all 80 trials, participants were simultaneously presented two texture image pairs (left and right) in which each image was blocked by using a green grid (and all grid blocks were further randomised in the POP_R). The participants were required to decide which pair was more similar. If they chose the left pair they pressed the "←" key; otherwise, they pressed the "→" key. Then the experiment automatically ran the next trial and continued until all 80 trials were done.

7.3 Experimental Results and Analysis

After the two modified pair-of-pairs experiments: POP_N and POP_R were completed, a dependent *t*-test was then conducted on the results of these experiments. In addition, a pair-of-pairs evaluation experiment was conducted on the results of the POP_R. The evaluation results were also compared with the evaluation results against the 80 corresponding pair-of-pairs judgements derived in the POP_O.

7.3.1 Experimental Results

Given that the pair-of-pairs judgement set: $POPJ_{POP}$ (see Equation (4.2)) made by the majority of the observers in the POP_O (only the 80 most inconsistent pairs of pairs are considered) are used as the baseline, we examined whether or not the human observers made significantly different judgements when they were still permitted to use long-range interactions (in the POP_N) from when they were prevented from using long-range interactions (in the POP_R). It should be noted that the single pair-of-pairs judgements

obtained using the original pair-of-pairs experiment cannot be used as the baseline because different populations of human observers were used in the original and modified pair-of-pairs experiments. In addition, there are no "zero-valued" judgements contained in $POPJ_{POP}$. In other words, either "left" or "right" pairs were chosen as more similar according to the 80 judgements in $POPJ_{POP}$.

The single pair-of-pairs judgement that observer m ($m = 1, 2, \dots, 10$) made in the i-th trial in one modified pair-of-pairs experiment is expressed as:

$$SPOPJ_M(m,i) = \begin{cases} 1, & \text{Choose "Left"} \\ -1, & \text{Choose "Right"} \end{cases}, m = 1, 2, \dots, 10, i = 1, 2, \dots, 80. \qquad (7.1)$$

where $SPOPJ_M$ is $SPOPJ_N$ or $SPOPJ_R$. The agreement rate (see Equation (4.6)) between the pair-of-pairs judgement set: $SPOPJ_N$ / $SPOPJ_R$ made by observer m in the POP_N/POP_R and the baseline ($POPJ_{POP}$) was computed. The two sets of agreement rates are labelled as: $AR_N(m)$ and $AR_R(m)$, $m = 1, 2, \dots, 10$. The means of the two sets of agreement rates are reported in Figure 7.2 and Table 7.1. It can be seen that the human observers in the non-randomised experiment agreed more with the observers in the original pair-of-pairs experiment than that they did in the randomised experiment, on average.

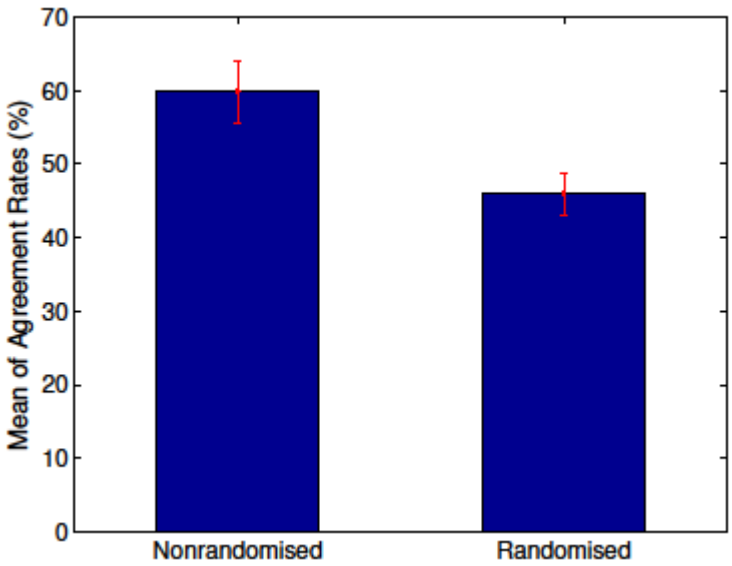


Figure 7.2: Means and 95% confidence intervals (error bars) of the agreement rate sets: AR_N and AR_R.

The agreement rates between the judgements made by the majority of observers in the non-randomised blocked and randomised blocked experiments ($POPJ_N$ and $POPJ_R$, obtained in the same way as $POPJ_{POP}$, see Equations (4.1) and (4.2)) with the judgements

made by the majority of observers in the original pair-of-pairs experiment are 68.75% and 46.25% respectively. This shows that the majority of the 10 observers agreed more with the observers in POP_O when the blocked images were used than when the randomised blocked images were used.

	AR_N	AR_R
Mean	59.88	46.00
Standard Error	2.17	1.48

Table 7.1: Means and standard errors of the agreement rate (%) sets: AR_N and AR_R.

7.3.2 Analysis of the Results

The well-known *t*-test [Joanl, 1987] is a statistical hypothesis test in which the test statistic satisfies a Student's *t* distribution if the null hypothesis holds true. The *t*-test is normally utilised to examine the significance of the difference between two sets of statistics which generally fit normal distributions. The *t*-test between the AR_N and AR_R is a dependent *t*-test because the same participants were used in both POP_N and POP_R. In this subsection, we first test the normality of the AR_N and AR_R and then examine the significance of the difference beteween these distributions.

K-S Tests

The Kolmogorov-Smirnov test (K-S test) [Kolmogorov, 1933] [Smirnov, 1948] was used to test the normality of the distributions. In addition to testing AR_N and AR_R, the K-S test was also applied to the difference of the AR_N and AR_R conditions because the *t*-test between the AR_N and AR_R is dependent [Field, 2009]. The results are reported in detail in Table 7.2. It shows that the three sets of statistics follow normal distributions according to the K-S tests. In addition, Figure 7.3 shows normal Q-Q plots [Wilk and Gnanadesikan, 1968] of the three sets of statistics.

K-S Test	Statistic	*df*	Sig. (*p*)	Is Normal
$\boldsymbol{AR_N}$	0.135	10	0.200	Yes
$\boldsymbol{AR_R}$	0.221	10	0.180	Yes
$\boldsymbol{AR_N - AR_R}$	0.247	10	0.086	Yes

Table 7.2: Results of three Kolmogorov-Smirnov (K-S) tests. The tests were conducted on three sets of distributions: AR_N, AR_R, and their difference.

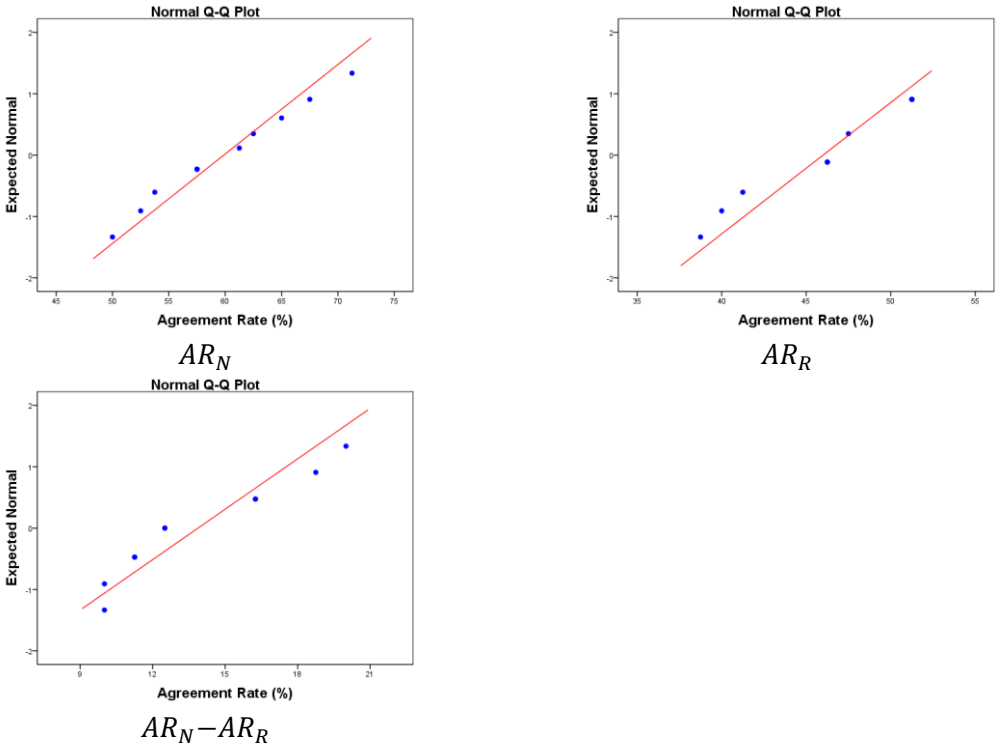


Figure 7.3: Normal Q-Q plots of the three sets of distributions: AR_N, AR_R, and their difference.

t-Test

A dependent *t*-test was conducted on the two sets of distributions: AR_N and AR_R. The results are reported in Table 7.3. These results show a significantly higher agreement with the observers in the original pair-of-pairs experiment when the 10 observers were presented non-randomised blocked pairs of pairs of images (M = 59.88, SE = 2.17) than when they were shown randomised blocked pairs of pairs of images (M = 46.00, SE = 1.48), $t(9) = 12.008$, $p < 0.05$, $r = 0.970$. This indicates that the randomisation processing which removed, or at least reduced, long-range interactions significantly affects human perceptual pair-of-pairs judgements.

α	*t*-test	*t*	*p*	*r*	*df*	*Sig.*
0.05	AR_N vs. AR_R	12.008	0.000	0.970	9	Yes

Table 7.3: Dependent t-test results ($\alpha = 0.05$), where $r \geq 0.5$ means that a strong effect was obtained.

7.3.3 Evaluation against $POPJ_R$

In the previous subsection, we analysed the effect of long-range interactions on human pair-of-pairs judgements. However, we have not answered the question as to why the 51 computational feature sets that we have tested do not agree with human observers. Considering each block of the grid that was superimposed on texture images only holds a small proportion of texture and the majority of the 51 computational feature sets extract features from overlapping neighbourhoods, the effect of blocking on the representation ability of these feature sets can be ignored. Furthermore, as discussed in Chapter 3, the majority of the 51 computational feature sets utilise short-range ($\leq$ 19×19 pixels) interactions and cannot exploit aperiodic long-range interactions. As a result, the effect of the randomisation process on the representation ability of those feature sets can also be ignored.

Based on the hypothesis above, the features extracted from the original texture images using the 51 sets of computational features were used again in this evaluation experiment. The perceptual pair-of-pairs judgement set ($POPJ_R$) obtained using the 80 pair-of-pairs in POP_R was used as the benchmark data. In addition, the results obtained using the corresponding 80 pair-of-pairs in the evaluation experiment in Section 5.2 were utilised for comparison. Figure 7.4 shows a scatter plot between the two sets of agreement rates obtained using the 51 computational feature sets against the perceptual pair-of-pairs judgements yielded in the original and the randomised, blocked pair-of-pairs experiments (POP_O and POP_R) in order to show the level of correlation. In addition, the Spearman's correlation coefficients between the two sets of agreement rates across six resolutions are -0.035, 0.134, 0.160, 0.224, 0.379 and 0.071 (p = 0.806, 0.350, 0.275, 0.114, 0.006 and 0.620) in turn. This suggests that the two sets of agreement rates do not correlate with each other well, no matter which resolution is considered.

Compared with the original pair-of-pairs experiment (POP_O), the randomised blocked pair-of-pairs experiment (POP_R) provides increased agreements (from 31.34% ± 0.1255 to 54.56% ± 0.0764 (mean ± standard deviation)) with the 51 computational feature sets on average. That is, when humans were permitted to exploit long-range interactions on images, they disagreed more with the computational results compared with when these long-range interactions had been removed or at least reduced. It indicates that humans exploit long-range interactions which are not normally available to the computational features that we have examined in this study.

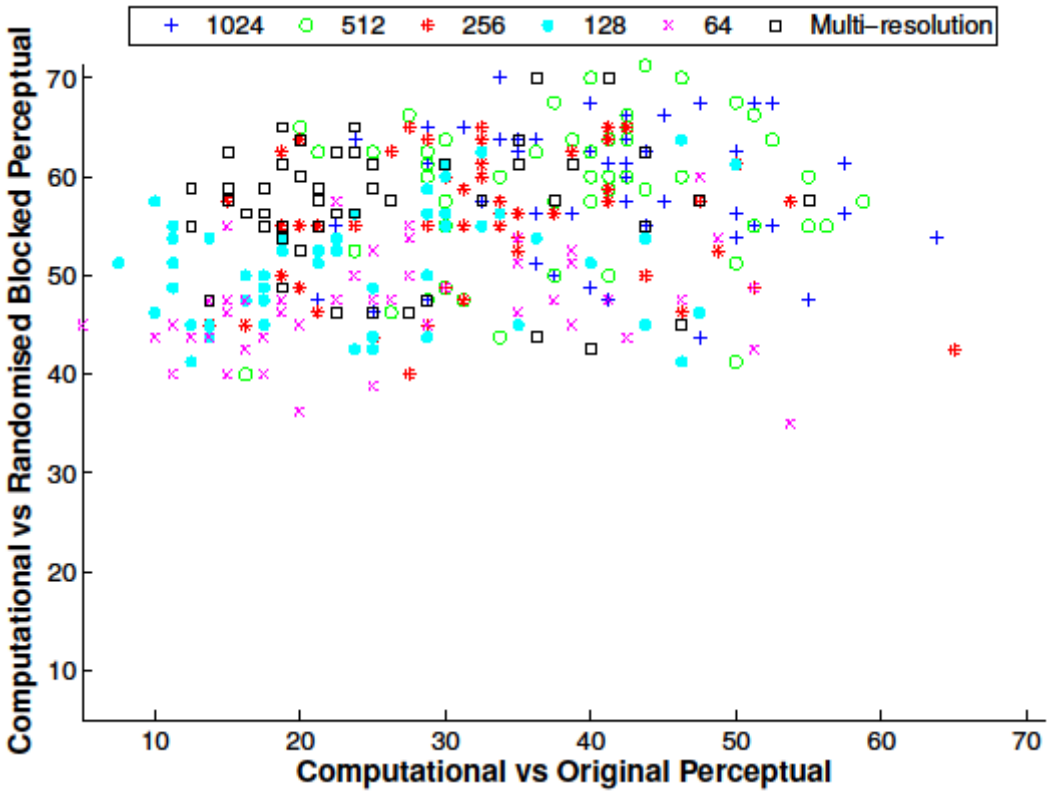


Figure 7.4: A scatter plot between two sets of agreement rates (%). The x-axis and y-axis show the agreement rates obtained using the 51 computational feature sets against the perceptual pair-of-pairs judgements yielded in the original and the randomised, blocked pair-of-pairs experiments respectively. The six different colours show the agreement rates derived at five individual resolutions and one multi-resolution scheme.

7.4 Conclusions

In this chapter, texture images in the 80 most inconsistent pairs of pairs (see Appendix E) were first "blocked" with a grid. The position of the blocks within an image was randomised in order to inhibit, or at least reduce, the exploitation of long-range interactions

to observers. One pair-of-pairs experiment (POP_R) was conducted using these randomised blocked images in order to investigate whether or not long-range interactions affect humans' judgements of texture similarity. In addition, in order to provide a "control" for the effect of the "blocking" operation on human texture similarity judgements, we carried out a second pair-of-pairs experiment (POP_N, without "randomisation").

The results of these experiments show that the experiment that utilised randomised blocked images (POP_R) produced significantly less agreement with the original experiment (POP_O) than the non-randomised blocked experiment (POP_N). What this suggests is that:

(1) the "randomisation" of image blocks does affect human perception of long-range interactions; and

(2) it is likely, therefore, that humans exploit long-range interactions for judging the similarity of textures.

Furthermore, as discussed in Section 3.8, the 51 computational feature sets that we examined cannot exploit aperiodic long-range interactions. In this situation, as the randomised blocked experiment provided increased agreement with the results obtained from the computational features, it seems likely that this increased agreement between humans and computational features arises because we have removed (or at least reduced) the long-range interactions that human exploit.

Thus, we conclude that long-range interactions are important to perceptual texture similarity.

Chapter 8

What Property Is Important to the Perception of Texture?

8.1 Introduction

The 334 textures in *Pertex* [Halley, 2011B] exhibit various texture characteristics, for example, directionality, regularity, periodicity and granularity. These characteristics can be modelled using different image properties such as power spectra, phase spectra, image exemplars and contours (see Section 2.5).

As discussed in Chapter 7, it is likely that humans exploit short-range and long-range interactions for judging the similarity of textures. However, not all categories of image properties will encode both short-range and long-range interactions. Power spectra, for example, can only encode periodic interactions, while phase spectra are able to encode aperiodic long-range interactions. However, as phase unwrapping is still an open problem for the use of phase spectra [Ying, 2006], we will not be considering that property in this thesis. Furthermore, image exemplars (blocks or patches) are able to encode only short-range interactions (see Chapter 7) whereas contours can encode periodic and aperiodic long-range interactions. In this chapter, we therefore ignore phase spectra and only consider the other three image properties: power spectra, image exemplars and contours, in order to determine which of these three properties is most important to the perception of texture.

The remainder of this chapter is organised as follows. Section 8.2 describes an experiment for examining the importance of three image properties to the perception of tex-

ture. Results are reported and analysed in Section 8.3. Finally, conclusions are drawn in Section 8.4.

8.2 Experimental Design

In this section, we present an experiment designed to determine which image property is most important among power spectra, image exemplars, and contours to texture perception. Correspondingly, three different sets of texture "property images", i.e. phase-randomised (power-only) images, randomised blocked images and contour maps, were used in three sessions of the experiment. A 2AFC (two-alternative forced choice) experimental design [Bogacz et al., 2006] was utilised. During each trial, the participant was required to compare an original texture image and a property image and then decide whether or not the property image represents the original one.

8.2.1 Hypothesis

Generally speaking, filtering-based features, except quadrature filters based features, only exploit the power spectrum and ignore the phase information (see Section 3.3). Since periodicity is generally associated with the power spectrum [Liu, and Picard, 1998], the power spectrum should be able to represent periodic textures well. On the other hand, aperiodic image structure is normally encoded by the phase spectrum [Oppenheim and Lim, 1991]. Since long-range interactions are normally associated with global perceptual phenomenon [Spillmann and Werner, 1996], the power spectrum is only able to encode periodic long-range interactions while the phase spectrum can encode aperiodic long-range interactions. However, very few features have been developed using global phase information. One possible reason is that the phase information is difficult to unwrap [Ying, 2006]. Hence, we ignored the phase spectrum in this study. We randomised the phase spectrum in each texture image in order to remove the effect of the phase information (see Section 2.5.1). The power spectrum is retained for each texture image. If the majority of observers think a texture image can be represented by its phase-randomised image, then the power spectrum is important to its perception.

As stated in Section 3.8, texton-based, statistical and model-based feature sets normally employ short-range ($\leq 19\times19$ pixels) interactions and cannot exploit aperiodic "long-range" interactions. Therefore, the effect of the "blocking" and "randomisation" on the

representation ability of those feature sets can be ignored. The randomised blocked images can only encode short-range interactions, as we discussed in Chapter 7. In this case, if the majority of observers judge that a texture image can be represented by its randomised blocked image, local image exemplars or patches are important for its perception.

In addition, as a popular visual cue, the contour (outline) was found to play an important role in the identification of objects [De Winter and Wagemans, 2004, 2008A, 2008B] [Panis et al., 2008] [Sassi et al., 2010]. Although several feature sets are designed to encode edge information in images, they only compute simple global statistics from the magnitudes or directions of the image gradients [Ojala et al., 1996]. However, the global statistics computed in the second stage are normally 1st-order statistics (see Table 3.2) and such statistics cannot capture spatial relationships. Hence, they can only encode higher order statistics in a small spatial extent. However, the contour captures higher order information in a longer range and thus encodes (both periodic and aperiodic) long-range interactions. Consequently, if one texture is chosen by the majority of observers as being represented by its contour map, contours are important to its perception.

Since different types of textures are represented by different image properties and global phase is difficult to unwrap [Ying, 2006], we only investigate power spectra, image exemplars, and contours in order to find out which of these more easily computed properties are likely to be important to the perception of texture.

8.2.2 Experimental Setup

The setup employed in this experiment is the same as that introduced in Section 7.2.2.

8.2.3 Stimuli

Corresponding to three image properties, three sets of "property images" obtained from the 334 textures in *Pertex* were used in the three sessions of this experiment. Some post-processing was also conducted on these property images.

Phase-Randomised (Power-Only) Images

Phase-randomised images were generated from the original images of the 334 textures in *Pertex* by using the method introduced in Section 2.5.1. Figure 8.1 presents four original texture images and their phase-randomised counterparts.

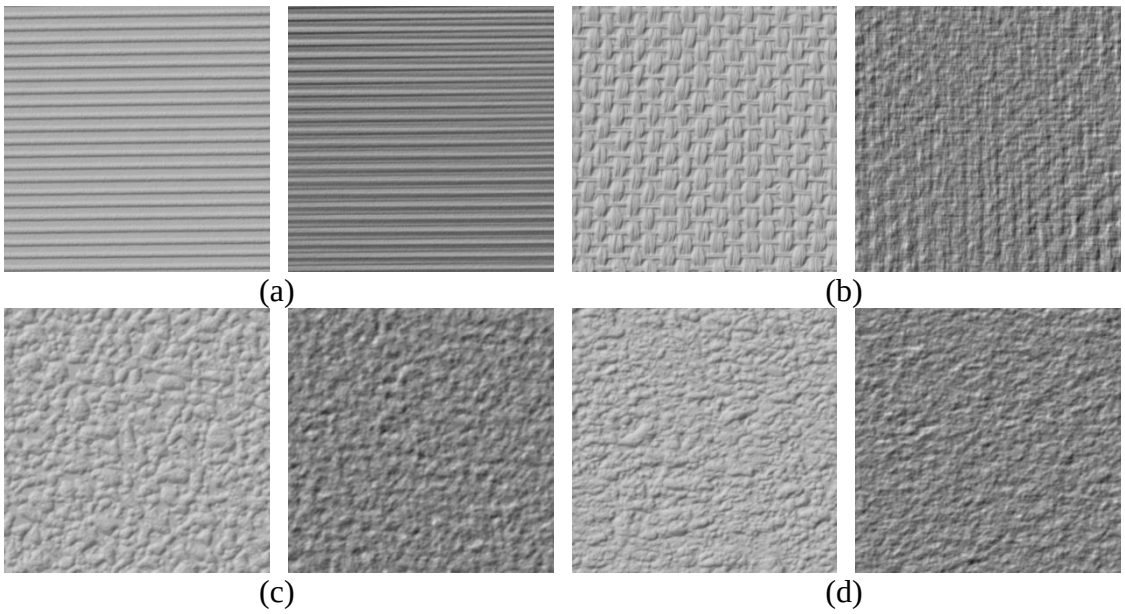


Figure 8.1: Original and phase-randomised texture images. The first and third columns display four original texture images: "044", "019", "012" and "042" (only top-left quarters are shown), and the second and forth columns show their phase-randomised counterparts (only bottom-right quarters are shown).

Randomised Blocked Images

As described in Chapter 7, randomised blocked images (see Figure 8.2) were obtained by first blocking the image with a green grid and then randomising the position of the blocks (i.e. image exemplars) in the grid. The thickness of the grid was set as three pixels. In addition, the size of the block was set to 19×19 pixels which is the largest neighbourhood exploited by the 51 computational feature sets (excluding filtering-based features, see Table 3.2).

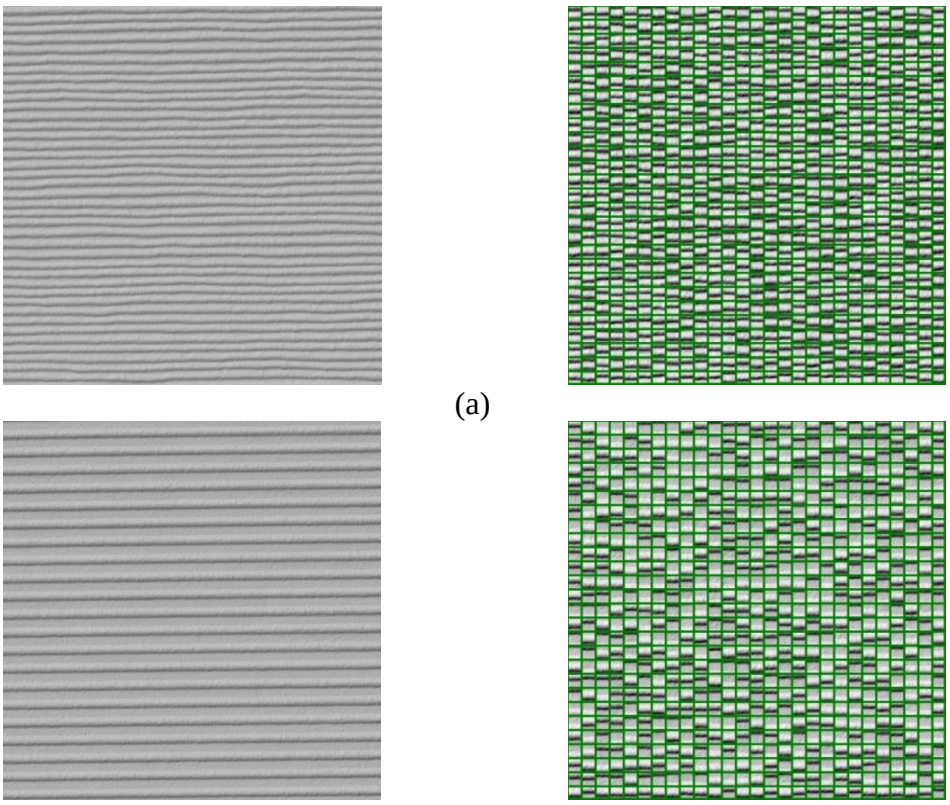


Figure 8.2: Original and randomised blocked texture images. The first column presents two original images: "047" and "044" (only top-left quarters are shown), and the second column displays corresponding randomised blocked images (only bottom-right quarters are shown).

Contour Maps (also Edge Maps)

The Canny edge detector [Canny, 1986] was used to extract contours from the 334 textures in *Pertex*. However, their height maps were utilised for this purpose, rather than the original texture images, as the effect of the illumination is reduced when height maps are used. In addition, the skeleton was obtained for each individual contour. This skeleton map (see Figure 8.3 (columns 2 and 4)) was then used instead of the contour map. In the rest of this chapter, the term "contour map" is used to refer to the skeleton map unless there are special descriptions or the contour map and skeleton map are mentioned together.

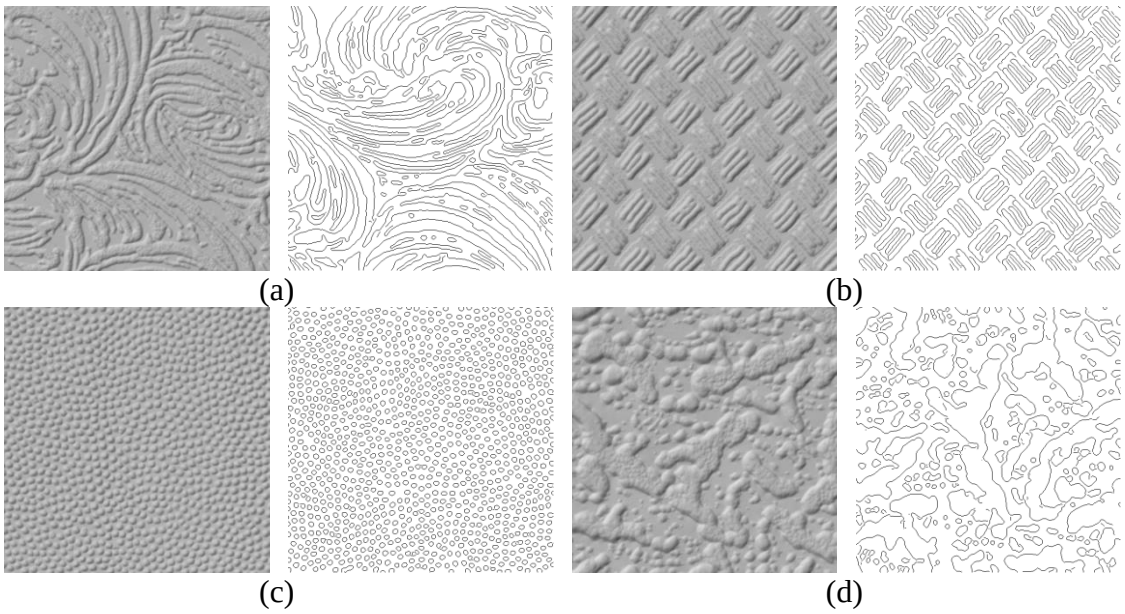


Figure 8.3: Original texture images and contour maps. The first and third columns display four original images: "026", "003", "020"and "029" (only top-left quarters are shown), and the second and forth columns show corresponding contour maps (only bottom-right quarters are shown).

Post-Processing

All 334 1024×1024 original and phase-randomised images were first normalised to have an average intensity of 0 and standard deviation of 1, in order to remove the influence of 1st- and 2nd-order grey level (moment) properties. The normalised images were scaled to the range of [0, 255] by the global maximum and minimum grey level values throughout all 334 normalised images in order to prevent observers from comparing images using grey level information when the phase-randomised images were used. In addition, randomised blocked images were obtained from the normalised and scaled original texture images. In this chapter, we use "original texture images" or "original images" to refer to the normalised and scaled original texture images (see Figures 8.1-8.3). The same 334 original images were used throughout all three sessions to keep the experimental conditions as constant as possible.

8.2.4 Observers

Throughout the three sessions of this experiment, 10 PhD students (all naïve to the experiment) at Heriot-Watt University with normal or corrected-to-normal vision were used. All 10 participants signed a consent form before they started the experiment. Each

participant was paid a 15 GBP Amazon voucher after they completed all three sessions of the experiment.

8.2.5 Reducing Biases

In order to obtain more reliable results, we used three processes to reduce any bias. First, we did not always show participants an original texture image with its property image as it might make them think that each image pair was from the same texture. We therefore presented one original texture image and the property image of a *different* texture to each participant in half the trials in each session. To be more specific, each participant was required to perform 334 trials in each session. All 334 textures were randomly divided into 2 groups (i.e. Groups 1 and 2) equally. In half of the trials (referred to as "correct trials") in each session, the original images in one group (e.g. Group 1) and their property images were used. But in the other half of the trials (referred to as "wrong trials"), the original images and the property images obtained from texture samples from the other group (Group 2) were employed. In other words, one original texture image was shown along with a property image generated from one of the other 166 images (excluding the current image) in the same group. All 334 trials were randomly shuffled for each participant.

	Name	**Session 1**		**Session 2**		**Session 3**	
		Group 1	**Group 2**	**Group 1**	**Group 2**	**Group 1**	**Group 2**
Instance 1	JC	√	×	×	√	√	×
	JN	√	×	×	√	√	×
	SW	√	×	×	√	√	×
	XJ	√	×	×	√	√	×
	YH	√	×	×	√	√	×
Instance 2	IK	×	√	√	×	×	√
	JL	×	√	√	×	×	√
	SQ	×	√	√	×	×	√
	XL	×	√	√	×	×	√
	YF	×	√	√	×	×	√

Table 8.1: "Correct" (√) and "wrong" (×) trials were generated from different groups in Instances 1 and 2. In addition, they were also generated from different groups for each observer in Session 1 (or 3) and Session 2.

Second, the 10 participants were divided into two equal-sized teams. In each session, the "correct trials" and "wrong trials" were generated from different groups for the two teams. Thus, two instances were created for two teams, in each session (see Table 8.1).

As listed in Table 8.1, correct property images and their original texture images were chosen from Group 1 in Session 1 or Session 3 while they were chosen from Group 2 in Session 2, when Instance 1 was performed. However, the reverse combinations were used in Instance 2. In addition, the three sessions were conducted at an interval of no less than seven days. With the help of these strategies, the learning effect should be reduced.

Finally, considering textures are normally regarded as homogeneous phenomena, each original or property image was divided into four equal-sized 512×512 quarters (see Figure 8.4). Throughout the three sessions, the top-left quarters of original images and the bottom-right quarters of property images were employed, in order to avoid, or at least inhibit, that participants comparing the original and property images pixel-by-pixel.

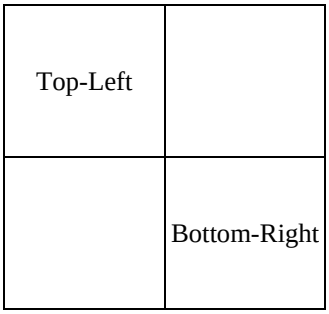


Figure 8.4: Selection of different image quarters for original and property images. The top-left quarters of original images and the bottom-right quarters of property images were used throughout all three sessions for the purpose of preventing participants from comparing original images with property images pixel-by-pixel.

8.2.6 Procedure

The experiment was divided into three sessions which were carried out separately with an interval of at least seven days. Each session was divided into two different instances (see Table 8.1). Correspondingly, two teams of participants (each team consists of five participants) were required to attend different instances of one session. In each session, a participant conducted 334 trials. Two images were presented in each trial: one original texture image and its, or another texture's, property image. Phase-randomised images, contour maps and randomised blocked images were utilised in the three sessions in turn.

In each trial, the participant was required to compare one original texture image and one property image and decide whether or not the property image represents the original. A

2AFC (two-alternative forced choice) experimental design [Bogacz et al., 2006] was employed. If the participant chose "yes", they pressed the left key "←"; otherwise, they pressed the right key "→". The system always waited for a response. Participants were allowed to take breaks as long as desired. The system automatically exited after all 334 trials were performed.

8.3 Experimental Results and Analysis

Three texture subsets were derived from 334 *Pertex* textures.

8.3.1 Results

A voting process was used for each texture in order to decide which property is most important to the perception of texture. Given an instance of one session (see Table 8.1), for each texture, if (1) the current original image and property image are from the same texture and (2) the majority of participants think the property image can represent the original, then we record that the texture can be represented by its property image; otherwise, it is assumed that the texture cannot be adequately represented by the property image. The threshold $T_{Population} = 4$ was used for thresholding the number of the participants (in total five participants were used in an instance).

Considering the condition when an original image and a property image of another texture were displayed, if the majority of participants indicate that the property image represents the original, then this could be due to: (1) the textures are "similar" (as indicated by the 8D-ISO similarity matrix) and the original image can therefore still be represented by its actual property image. In this case, another threshold $T_{Similarity} = 0.75$ (over 93% of the entries in 8D-ISO are less than this value) was applied for thresholding the similarity of two different textures; otherwise, (2) the texture was considered not to be well represented by its property image.

The textures shown to be well represented by their property images in the two instances of one session were merged into a subset. In total, three texture subsets were obtained using the three sessions. Table 8.2 reports the sizes of the three texture subsets. It can be seen that the results of the experiment suggest that the contour map is the most representative of the three types of property images. In addition, the sizes (see the overlap-

ping area) of the intersections of texture subsets are also shown in Figure 8.5. It can be observed, for instance, that 92 out of 334 textures can be represented by all three types of properties. Furthermore, the three texture subsets were divided into 14 clusters (also see Appendix A for more details) according to the dendrogram in Figure 4.5. The percentage sizes of each cluster for each of the three texture subsets are displayed in Figure 8.6. It can be observed that (1) the contour map can represent not only periodic or nearly-periodic textures but also aperiodic (random) textures; (2) the phase-randomised images are generally able to represent periodic, nearly-periodic, or aperiodic but well-ordered textures; and (3) the randomised blocked images can represent both periodic, or nearly-periodic, and aperiodic textures but are the least perceptually-representative property images.

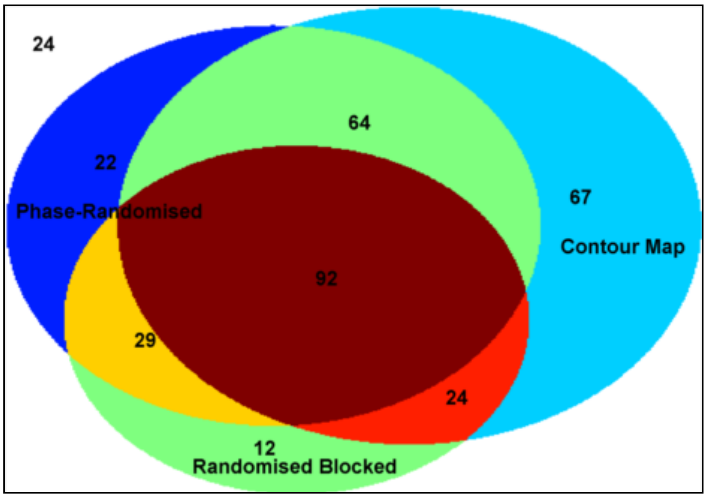


Figure 8.5: The three ellipsoids show the texture subsets chosen using three different types of property images respectively. The sum of the four numbers in each ellipsoid denotes the size of the current texture subset. In addition, the number in an overlapping area shows the size of the intersection of involved texture subsets (i.e. ellipsoids).

	Subset			**Intersection**
	Contour Map	**Phase-Randomised**	**Randomised Blocked**	
Size	247	207	157	92

Table 8.2: The three texture subsets chosen using three different types of property images respectively and the size of their intersection.

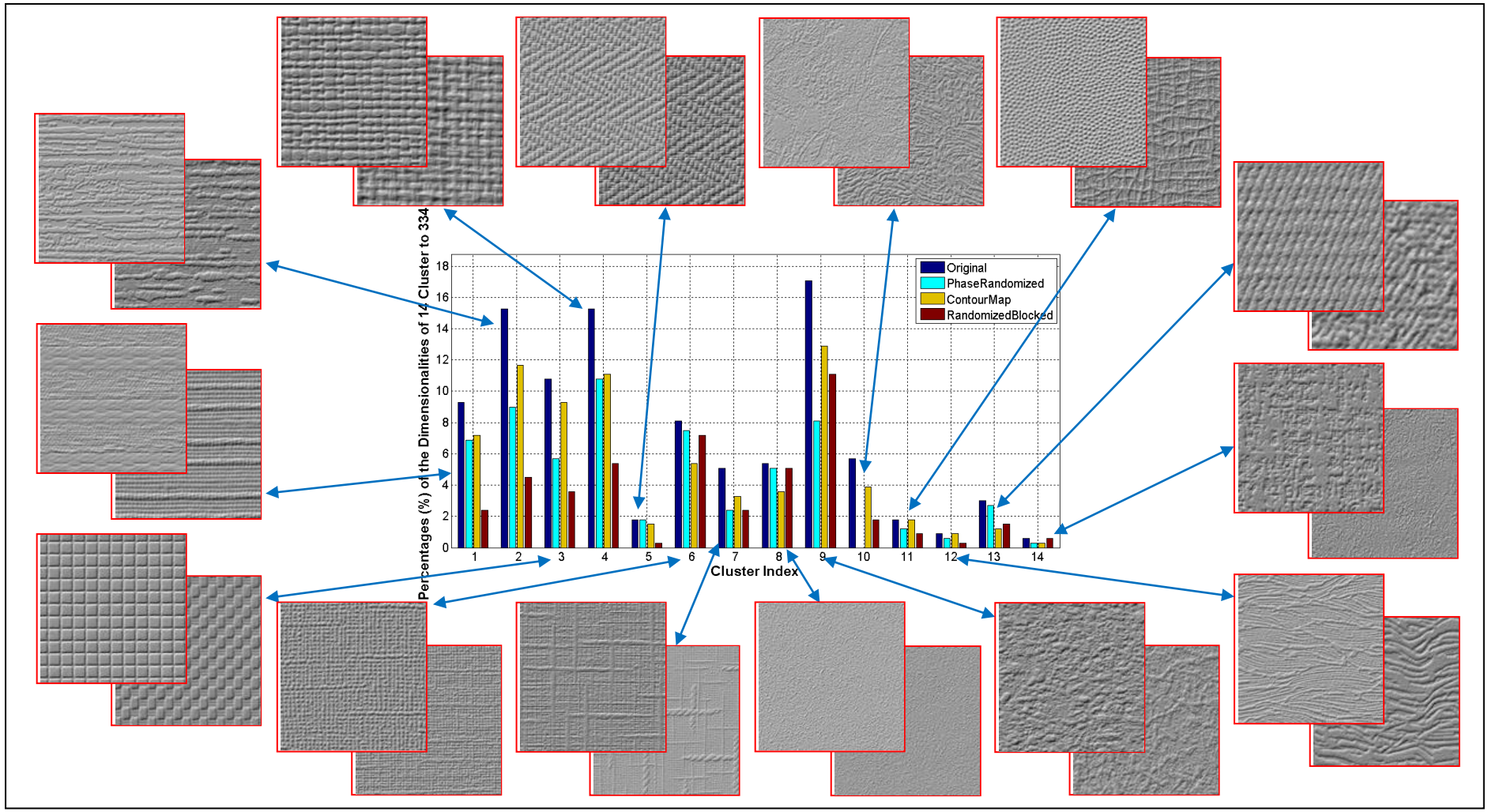


Figure 8.6: The percentages of the sizes of 14 clusters (see Appendix A) in 4 sets of textures (the Pertex dataset and its 3 subsets which can be represented by 3 types of property images) relative to the full 334 texture dataset are shown above together, with two representative textures of each cluster.

8.3.2 Evaluation on the Largest Subset

Since 247 textures have been successfully represented by their contour maps, in this subsection, the retrieval-based evaluation method introduced in Section 6.2 is used with these textures in order to examine the performance of the 51 computational feature sets to exploit the contour information for estimating perceptual texture similarity. The same evaluation method was used as that utilised in Chapter 6 except that only 247 textures were used rather than 334 textures.

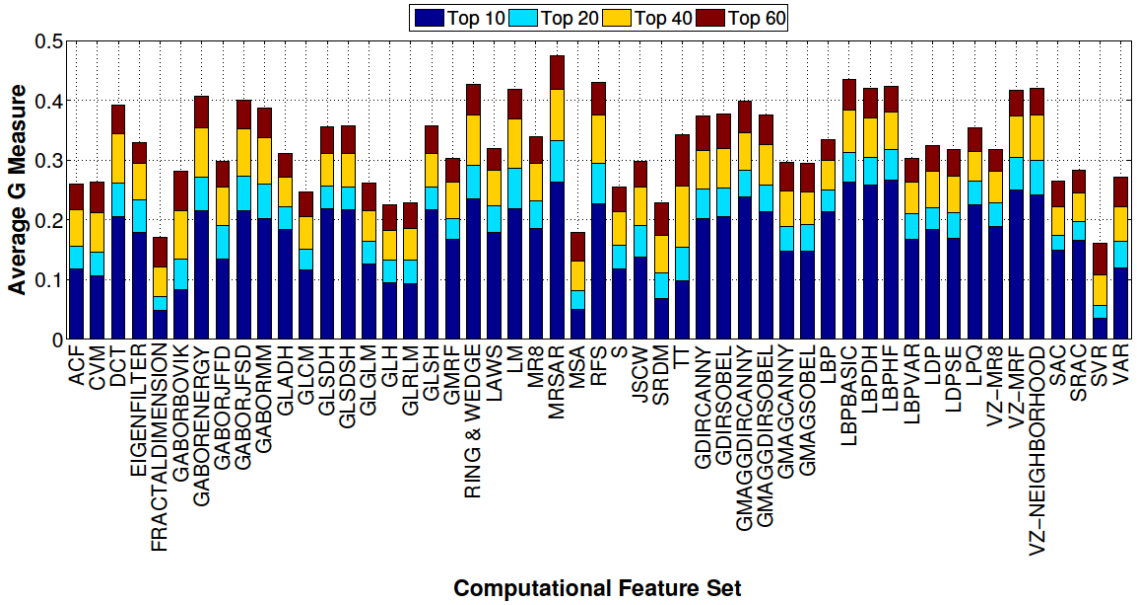
Since the advantages of the multi-resolution scheme have been illustrated in Chapters 5 and 6, we only report the average *G* and average *M* measures obtained using the 51 feature sets at this scheme. Figure 8.7 displays average *G* and average *M* measures for the four retrieval set sizes and 51 feature sets. The best performance obtained using these feature sets provides average *G* and average *M* measures of 0.48 and 0.31, which were obtained using MRSAR [Mao and Jain, 1992] when 60 textures were retrieved. These results imply that none of these feature sets exploit the contour information as well as humans do.

8.3.3 Summary

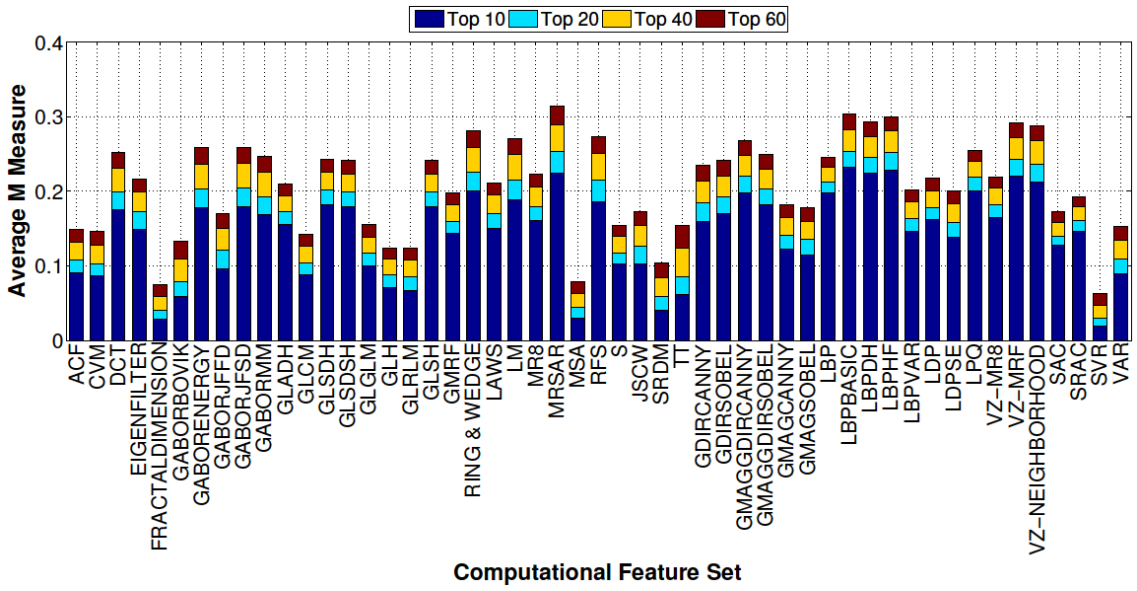
In summary, the contour map is the most representative one among the three types of property images examined and can represent 247 out of the 334 textures in *Pertex*. When the textures in *Pertex* are divided into 14 clusters (see Appendix A for more details), the contour map is the most important to the perception of nine clusters (see Figure 8.6).

In addition, the 80 most inconsistent pairs of pairs of textures were chosen in Appendix E, which contain 104 individual textures. Among these textures, 77 can be represented by their contour maps, according to the results obtained in Section 8.3.1. The proportion of 77/104 (74.04%) is close to the 247/334 (73.95%) seen in the full *Pertex* database. It implies that the contour map is equally important for the perceptual judgements of the 80 most inconsistent pair-of-pairs of textures.

However, none of the 51 feature sets examined in this study utilised this information as well as human observers did.



(a)



(b)

Figure 8.7: Average G (a) and average M (b) measures obtained using 51 computational feature sets at the multi-resolution scheme when 247 textures which can be perceptually represented by their contour maps were considered. Each bar shows four different, colour-coded results for the four values of $N \in \{10, 20, 40, 60\}$.

8.4 Conclusions

In this chapter, we asked participants to compare 334 texture images with their three "property images" corresponding to three different image properties. The objective was to determine which property is most important for the perception of texture. Experimental results show that the contour map is the most important property for the percep-

tion of texture and is able to represent 247 out of the 334 *Pertex* textures. The strong representation ability of the contour map agrees with results obtained for the identification of everyday objects [De Winter and Wagemans, 2004, 2008A, 2008B] [Panis et al., 2008] [Sassi et al., 2010]. Compared to the contour map, the power spectrum (as represented by a phase-randomised image) and randomised local image exemplars (randomised blocked image) can only represent 207 and 157 texture images respectively.

In addition, Figure 8.6 (also see the descriptions in Appendix A) shows that the contour map is able to represent periodic/nearly-periodic textures, aperiodic but well-ordered textures, blob-like textures, swirly textures, and even other types of random textures. In the field of vision science, it is also well-known that humans are extremely adept at exploiting the long-range visual interactions evident in contour information [Field et al., 1993] [Pettet et al., 1998]. The contour map, therefore, is able to encode both periodic and aperiodc long-range interactions. However, the phase-randomised image can only capture periodic long-range interactions while the randomised blocked image can only encode short-range interactions. This probably explains why the contour map is the most representative among the three image properties. It is also noteworthy that a subset of nearly-periodic textures can be represented by their corresponding randomised blocked images as well. In this context, the importance of short-range interactions cannot be ignored. However, the contour map also encodes short-range interactions.

Finally, in addition to the experiments performed using human observers, a retrieval-based evaluation experiment using computational features was conducted on the 247 textures that can be represented by their contour maps. However, the weak results suggest that none of these features utilise the contour information as well as humans do.

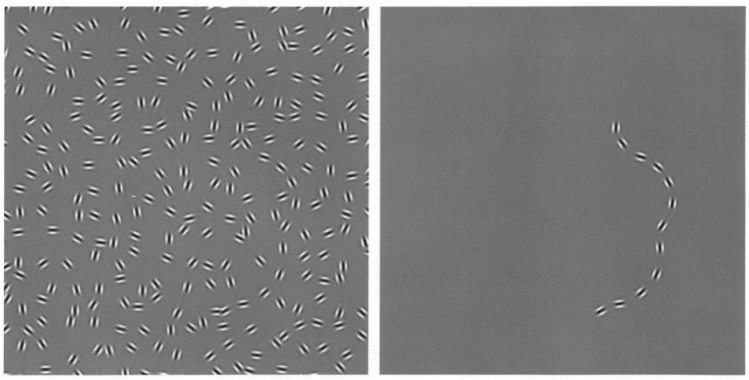
Chapter 9

Texture Features Using the Spatial Distributions and Orientations of Contour Segments

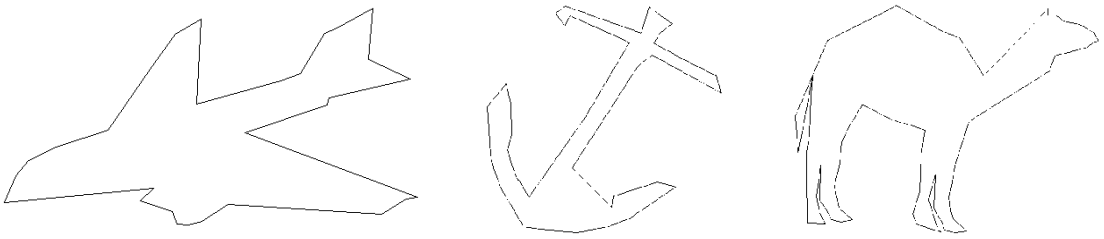
9.1 Introduction

As discussed in Chapter 3, few feature sets encode aperiodic, long-range characteristics of texture; however, it is well-known that such data are critical to human perception of imagery [Oppenheim and Lim, 1991] [Field et al., 1993] [Pettet et al., 1998] [De Winter and Wagemans, 2008B]. It is also well-known that humans are extremely adept at exploiting the long-range visual interactions evident in contour information [Field et al., 1993] [Pettet et al., 1998] [Hansen and Hess, 2006].

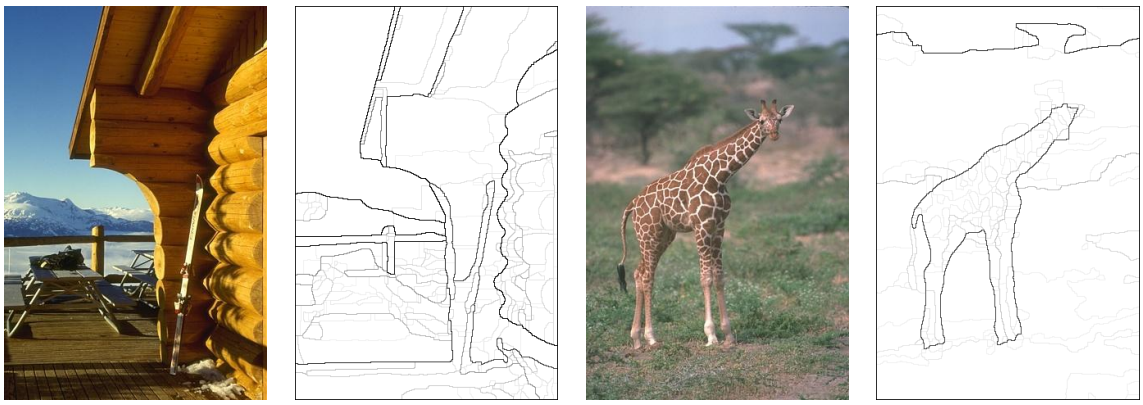
A contour is commonly thought of as "an outline or silhouette, or a contour line on a contour map, or the corresponding line on the ground or sea bed" [Contour]. In mathematics, a contour is also described as a directional curve which consists of a finite series of directional smooth curves whose endpoints are matched to display a single direction. Dakin and Hess [1999] defined contours to "consist mainly of edges which are spatially broadband and whose (cosinusoidal) components have arrival phases close to ±90°". In addition, Papari and Petkov [2011] proposed a further definition of contour: "the set of lines that human observers would concent on to be the contours in that image". Figure 9.1 shows three different groups of contours used in vision science and computer vision.



(a)



(b)



(c)

Figure 9.1: Examples of contour from the literature. (a) A contour used for contour integration research [Field et al., 1993]; (b) three straight-line versions of contours of everyday objects [De Winter and Wagemans, 2008B]; and (c) original images and their contour maps in computer vision [Arbelaez et al., 2011].

In psychophysics, contours (outlines) have been found to play an important role in the identification of objects [De Winter and Wagemans, 2004, 2008A, 2008B] [Panis et al., 2008] [Sassi et al., 2010]. In the previous chapter, we presented information that suggests that the contour map encodes significant information that is used for the perception of texture. However, to the best of our knowledge, no research has been reported that exploits contour information directly for texture analysis.

In this chapter, first of all, we review a set of related contour representation techniques in order to investigate whether or not there exist suitable approaches for encoding the

contours extracted from a texture image. We then introduce a feature set that exploits the long-range HOS (higher order statistics) encoded in the spatial distributions and orientations of contour segments. The proposed feature set is compared with the 51 feature sets that we tested in Chapters 5 and 6 as well as one conventional shape recognition type feature set.

To be more specific, Section 9.2 reviews 15 types of contour representation methods. A new contour-based feature set is proposed in Section 9.3. Furthermore, in Section 9.4, the proposed feature set is compared with 52 feature sets under the pair-of-pairs based and retrieval based evaluation methods introduced in Chapters 4 and 6 respectively. Conclusions are finally drawn in Section 9.5.

9.2 Survey of Contour Representation Approaches

Contour representation approaches normally extract features from the boundary of a shape. Generally speaking, these approaches are classified into two classes: structural (discrete) and global (continuous) approaches [Zhang and Lu, 2004]. In this section, we will review a series of existing contour representation approaches in order to investigate whether or not these approaches can be used for encoding the contours extracted from a texture image.

9.2.1 Criteria for the Survey

Four criteria that are considered important are introduced below.

- **Generative**

Generative features allow the contours to be recreated thereby providing insight into the information that they encode. More compact features could be obtained from the generative features in the following stages. Thus, we only require the features extracted in the first stage to be generative.

- **Noise-Insensitive**

Basically, contour detection algorithms might introduce noise (small points or contour variations). Although most noise could be removed using a post-processing, it is not practical to remove all of the noise. Considering the discriminatory power of contour representation algorithms might be impaired by noise, noise-insensitive algorithms are preferred.

- **Encoding the Spatial Distribution of Contours**

Normally, many contours are detected in an image and one feature vector is extracted from each contour. The spatial distribution of different contours is important for encoding the global structure of the image. Therefore, contour representation approaches should be able to encode the spatial distribution of contours.

- **Simple Computation and Matching**

As mentioned before, a texture image generally produces a large number of contours, hence, the computation for encoding one individual contour should be simple in order to achieve a high computational efficiency. Furthermore, the comparison of two feature vectors will need to be simple, and so the direct matching of contours should be avoided as it is time-consuming.

9.2.2 Structural Methods

Structural contour representation approaches fragment a contour into a set of segments which are normally referred to as primitives and then encode these segments into a feature vector or compare two segment sets directly.

Chain Code Histogram

A chain code represents a contour using a series of unit-size directional line segments [Freeman, 1961] [Freeman and Saghri, 1978]. Since the shape of a contour can be regenerated from its chain code vector, it is a generative method. However, it is sensitive to noise. A histogram can be accumulated from the chain code vector to reduce its dimensionality [Iivarinen and Visa, 1996]. Chain code histograms cannot encode the spa-

tial distribution of contours. However, the computation and matching of chain code histograms are simple.

Curve Interpolation and Approximation

Generally speaking, curve fitting [Arlinghaus, 1994] constructs a curve to optimally fit a set of points under some constraints. Curve fitting can be divided into two methods: curve interpolation [Glass, 1966] and curve approximation [Speer et al., 1998]. Both of these methods can restore the appoximate shape of a curve and are thus generative. The former requires a precise fit to the data points while the latter normally applies a smoothing function to approximately fit the data. As a result, curve interpolation is sensitive to noise but curve approximation is not. However, neither curve interpolation nor curve approximation can capture the spatial distribution of contours. Since interpolation or linear regression algorithms are usually involved, the computation of curve interpolation and approximation is not efficient. Nevertheless, the matching of two sets of coefficients is simple.

Polygon Decomposition

Polygon decomposition first approximates a contour as a polygon and then extracts features from its vertices [Groskey et al., 1990 & 1992] or a set of interest points [Mehrotra and Gary, 1995]. Since the original shape of a contour can be roughly restored when the number of vertices or interest points is large, polygon decomposition is generative. Furthermore, it is insensitive to noise. However, it cannot encode the spatial distribution of contours. In addition, the computational efficiency of polygon decomposition is low while the matching process is simple.

Curve Decomposition

Berretti et al. [2000] further extended the approach proposed by Groskey et al. [1990] and used the curvature zero-crossing points from a Gaussian smoothed boundary as primitives (or tokens). Each primitive was encoded by its maximum curvature and its orientation. Similarly, Dudek and Tsotsos [1997] first derived shape primitives from a contour and then represented each segment with its length, ordinal position, and the curvature tuning value. The contour is finally encoded by a series of segment descriptors. Since curve primitives can retain the original shape, curve decomposition is

generative. In addition, it is insensitive to noise. However, this method cannot encode the spatial distribution of contours. The decomposition of contours is not efficient but the matching of two sets of descriptors is simple.

Syntactic Analysis

Syntactic methods normally describe a contour with a series of predefined primitives. Syntactic shape analysis [Chomsky, 1957] is designed to encode the structural and hierarchical characteristics of human vision mechanisms. However, a priori knowledge for the image dataset is required in order to obtain primitives, which means that this kind of method is not practical [Zhang and Lu, 2004]. Syntactic shape analysis can be taken as generative only if all primitives are representative. However, it is not sensitive to noise. The syntactic analysis method cannot encode the spatial distribution of contours. In addition, the direct comparison between a contour and primitives are complicated.

Shape Invariants

Shape invariants can be classified into geometric invariants [Huang et al., 1998] [Li, 1999], algebraic invariants [Squire and Caelli, 2000] and so on. Since shape invariants are obtained from primitives, they are generative. However, invariants based contour representation approaches have two critical problems: (1) the invariants cannot resist to the influence of the boundary noise and errors; and (2) obtaining a suitable solution for the feature matching in acceptable time is difficult [Zhang and Lu, 2004]. Furthermore, they cannot encode the spatial distribution of contours.

9.2.3 Global Methods

Global methods integrally consider a contour and directly extract a feature vector from it.

Simple Shape Descriptors

Simple shape descriptors normally consist of area, bending energy, circularity, eccentricity and major axis orientation [Yong et al., 1974]. One problem with these descriptors is that they are unable to discriminate contours with inconspicuous differences. In addition, Peura and Iivarinen [1997] also proposed a set of global shape descriptors,

including circular variance, convexity, elliptic variance and ratio of the principle axes. None of these descriptors are generative but they are insensitive to noise. Furthermore, they cannot encode the spatial distribution of contours. However, the computation and matching of these descriptors are simple.

Slope Representation

Slope representation (using a ψ-s plot) normally starts from a terminal point of a contour and plots the tangent ψ between the current point and the previous point versus the current arc length s to represent the contour in a ψ-s plane [Jain et al., 1995]. It is generative because the original shape can be restored from its ψ-s plot. Slope representation is sensitive to noise and cannot encode the spatial distribution of contours. However, the computation and matching of this method are simple.

Shape Signatures

A shape signature is computed as a function of the points of a contour. The popular shape signatures consist of area, centroid distance, centroidal profile, chord-length, complex coordinates, cumulative angle, curvature and tangent angle [Freeman, H., 1977] [Van, 1991] [Davies, 1997] [Zhang and Lu, 2002]. The majority of these signatures are not generative and they are sensitive to noise. None of these can encode the spatial distribution of contours. In addition, for most of these shape signatures, the matching is complicated.

Shape Contexts

This method extracts a global feature, namely, shape context, from a contour for each sampled point [Belongie et al., 2002]. A shape context is obtained between the current sampled point and the other sampled points. Since the sampled points can retain the rough original shape, this method can be regarded as generative. Shape contexts are insensitive to noise. They cannot encode the spatial distribution of contours. However, the computation and matching are simple.

Boundary Moments

Boundary moments were introduced to decrease the dimensionality of the feature vectors extracted using shape signatures. In the case that one contour has been encoded us-

ing a series of shape signatures, a set of moments can be then computed from these signatures directly [Sonka et al., 1993] or be calculated from the histogram accumulated from these signatures [Gonzalez and Woods, 2002]. Boundary moments are not generative and are sensitive to noise. In addition, none of these can encode the spatial distribution of contours. However, the matching of boundary moments is simple.

Stochastic Methods

Contour descriptors can also be computed using time-series models, such as autoregressive (AR) models [Dubois and Glanz, 1986] [Das et al., 1990] [Sekita et al., 1992]. Stochastic methods first compute shape signatures from the points of a contour and then extract some models from these signatures. These methods are not generative and are sensitive to noise. Furthermore, none of those methods can encode the spatial distribution of contours. However, the computation of model coefficients and matching of two sets of model coefficients are simple.

Scale Space

Scale space analysis can also be used to reduce the sensitivity of the representation to noise and contour variations. Low-pass Gaussian filters with changing widths (i.e. scales) are first utilised to smooth a contour. Then, the position of inflection points on the contour is tracked and the scale space representation of the contour is obtained from these points. The smoothing operation will generate an interval tree, i.e. “fingerprint”, including a set of inflection points. Various features can be extracted from the interval tree [Asada and Brandy, 1986] [Mokhtarian et al., 1996] [Abbasi et al., 1999]. Scale space analysis is generative when the scale is small and is insensitive to noise. Nevertheless, it cannot be used to encode the spatial distribution of contours. Although its computation is simple, the matching process is complicated.

Fourier Descriptors

After shape signatures have been transformed using the Fourier transform, Fourier descriptors (FD) [Persoon and Fu, 1977] [Arbter et al., 1990] are then obtained. Since the inverse Fourier transform is known, the method is generative when both the magnitude and phase are retained. Fourier descriptors can be insensitive to noise and the variations of the contour although this naturally depends upon the frequencies of the coefficients

employed. They cannot capture the spatial distribution of contours. However, the computation and matching of FD are simple. Unfortunately, they can only be used on closed contours (boundaries). Thus, Fourier descriptors are not suitable for representing open contours.

Contour Representation Approaches		Criteria			
		Generative	**Noise-Insensitive**	**Encoding Contour Spatial Distribution**	**Simple Computation & Matching**
Structural Method	Chain Code Histogram	✓	✗	✗	✓
	Curve Interpolation	✓	✗	✗	✗
	Curve Approximation	✓	✓	✗	✗
	Polygon Decomposition	✓	✓	✗	✗
	Curve Decomposition	✓	✓	✗	✗
	Syntactic Analysis	✓	✓	✗	✗
	Shape Invariants	✓	✗	✗	✗
Global Method	Simple Shape Descriptors	✗	✓	✗	✓
	Slope Representation	✓	✗	✗	✓
	Shape Signatures	✗	✗	✗	✗
	Shape Context	✓	✓	✗	✓
	Boundary Moments	✗	✗	✗	✓
	Stochastic Methods	✗	✗	✗	✓
	Scale Space	✓	✓	✗	✗
	Fourier Descriptor	✓	✓	✗	✓

Table 9.1: The eligibility of contour representation methods in terms of the four criteria.

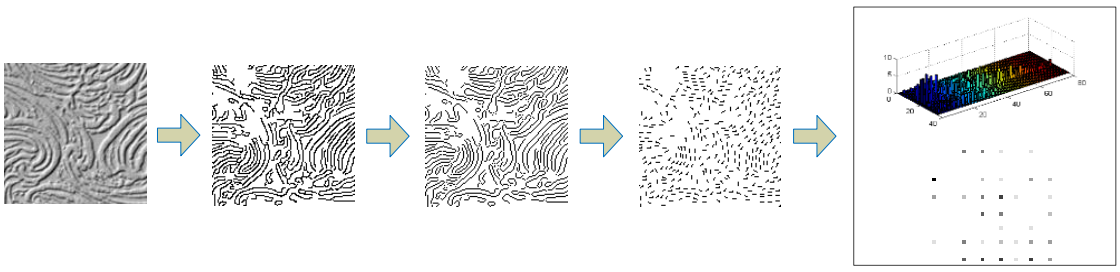
9.2.4 Summary of the Survey

Table 9.1 summarises the eligibility of the approaches surveyed above according to the four criteria that we have chosen. It can be seen that the structural approaches can be regarded as generative because they utilise primitives which retain the primary shape of a contour. However, noise sensitivity affects the discriminatory power of a number of shape representation methods. Since a large number of contours are normally extracted from one texture image, the spatial distribution of these is required in order to efficiently encode multiple contours. Nevertheless, these approaches have been designed to encode an individual contour and none of these consider the spatial distribution of multi-

ple contours. When large numbers of contours have to be encoded the computation or matching is complicated, and the total computational cost can become unacceptable. Consequently, the efficiency of the computation and matching restricts some of the approaches. To summarise, none of the approaches investigated satisfy all the four criteria. In this situation, a new contour representation method is therefore required.

9.3 Spatial Distributions and Orientations of Contour Segments

This section introduces a new contour-based texture feature set: spatial distributions of contour segments (SDoCS). Essentially, each contour is extracted and encoded as a set of segments. We use these data in two ways as outlined in Figure 9.2. In the first we encode the average shape of the contours in a segment joint orientation/distance histogram. This provides data on the long-range higher-order visual interactions that these contours provide. In the second we encode the spatial distributions and orientations of the all of the segments within a local window without regard to which contour they belong. These data naturally provide relatively shorter-range (23×23 or less) HOS.



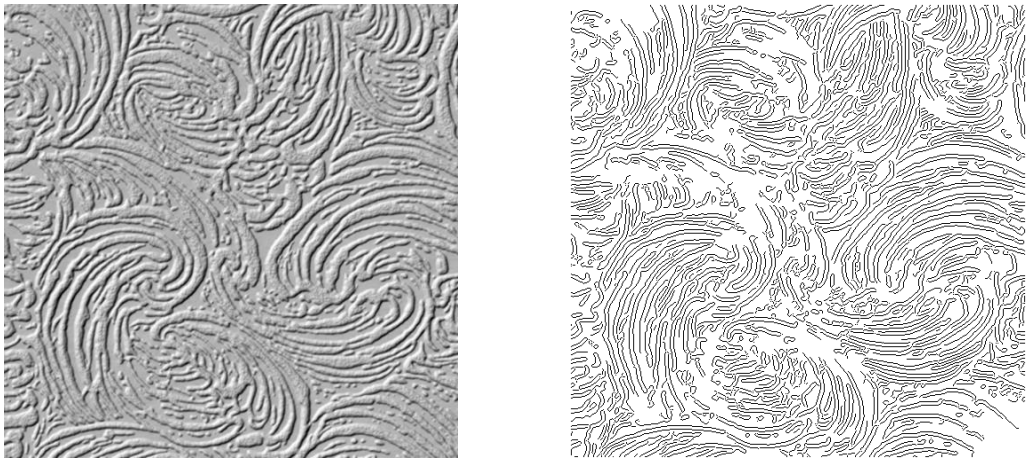
(a) (b) (c) (d) (e)

Figure 9.2: A representation of the basic information flow: (a) original texture image; (b) edge map; (c) skeleton map; (d) segment map. For display purposes, only a part of pixels are shown for each approximate segment; and (e) the joint histogram (upper) and basic aura matrix (lower, only one basic aura matrix is shown here).

9.3.1 Obtaining the Skeleton Maps

We utilised the Canny edge detector [Canny, 1986] to extract contours from a texture image, due to its simplicity and effectiveness. However, the contours extracted normally contain more pixels and are thicker than a single pixel. This increases computational

complexity and impairs the performance of contour representation algorithms. Erosion operations in mathematical morphology [Gonzalez and Woods, 2002] are hence applied on the contour map with a 3×3 neighbourhood in order to remove redundant pixels without allowing contours to break apart. Figures 9.3 (a) and (b) show a texture image ("026" in *Pertex*) and its skeleton map respectively. If not explicitly stated otherwise, the term "contour map" is used to refer to the skeleton map in the rest of this chapter.



(a) (b)

Figure 9.3: A texture image (a) and its skeleton map (b).

9.3.2 Producing the Segment Maps

In this subsection, each contour is first fragmented into a series of equal-length segments.

Tracing a Contour

Prior to conducting our contour representation method, all contours should be traversed from one end to another end in order to obtain a sequence of contour points as the input of the method. It is observed that a number of contours contain branches which make contour representation more difficult. In this case, all branch points are located. The contours involved are then broken into multiple new contours by directly deleting their branch points (see Figure 9.4).

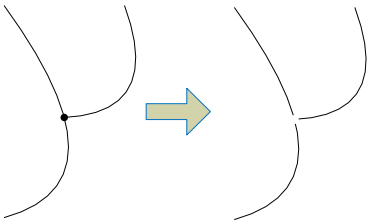


Figure 9.4: (Left): One contour with a branch; (right): three contours obtained from the contour at the left side by removing the branch point.

Connected component labelling [Dillencourt et al., 1992] with 8-connected neighbourhoods (see Figure 9.5) is performed on the skeleton map and a connected component is obtained for each continuous contour. The Moore-Neighbour tracing algorithm with Jacob's stopping criteria [Gonzalez and Woods, 2002] is applied to each component from which a sequence of points is obtained. However, the exterior boundary of one component is derived rather than the component (contour) itself because the tracing algorithm considers each component as a region. Since the Moore-Neighbour tracing algorithm chooses the left-most point as the starting point (see Figure 9.6), the boundary point sequence varies with different contour shapes. Figure 9.6 presents three representative contour shapes. We separately obtain the traversing sequences of the three types of contours as described below:

(1) For the contour displayed in Figure 9.6 (a), the tracing operation starts from the left-most point (the black bold point) and traverses clockwise point by point. After arriving at the end point, the algorithm turns back and traverses all points in a reverse sequence until it reaches to the starting point. As all points are visited twice except the end point, the boundary traversing sequence is symmetric with a centre at the end point. Given that the contour consists of n points, the boundary sequence includes $2n - 1$ points. In this case, the first n points in the boundary sequence are actual sequential contour points.

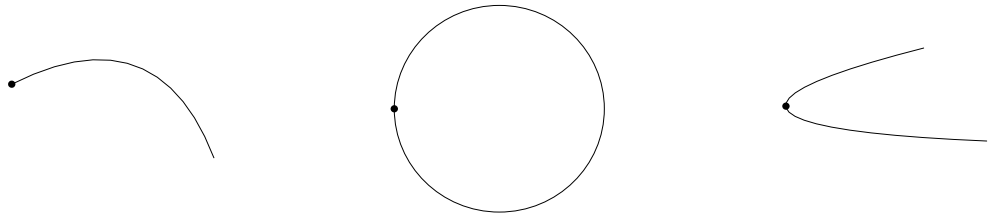
(2) Considering the closed contour presented in Figure 9.6 (b), the tracing algorithm also departs from the left-most point (the black bold point) and traverses clockwise point by point until it returns to the starting point. Thus, only the starting point is visited twice. Given a closed contour with n points, the boundary sequence contains $n + 1$ points. Since our contour representation algorithm does not tackle closed contours, only the $n - 1$ points are used (the starting point is discarded).

(3) Regarding the contour shown in Figure 9.6 (c), there may be several points in the left-most column. In this situation, the tracing algorithm sets off from the top-left point

and traverses the contour clockwise until it arrives at the bottom end point. It then turns back and traverses the points between the starting point and this point in a reverse sequence until it returns to the starting point, after which it continues until it reaches to the top end point where it turns back and traverses all points between this end point and the starting point. Once it arrives at the starting point (the third time), the tracing is complete. Thus, the starting point is visited three times. Given that the positions of the starting point in the sequence are P_1, P_s, and P_t in sequence, the points between P_1 and P_s and the points between P_s and P_t in the sequence can be processed as shown for the contour in Figure 9.6 (a). Finally, the reverse of the first output is merged with the second output excluding its starting point.

g_1	g_2	g_3
g_8	g_0	g_4
g_7	g_6	g_5

Figure 9.5: An 8-connected neighbourhood of pixel g_0, i.e. Moore-Neighbour.



(a) (b) (c)

Figure 9.6: Three types of representative contours: (a) an open contour with the starting point at one end; (b) a closed contour; and (c) an open contour with the starting point between two end points. Here, the black bold point means the starting point of each contour for the contour tracing algorithm.

Fragmenting a Contour into Segments

It was found that humans are able to integrate a continuous contour from a series of discontinuous contour segments [Field et al., 1993] [Kovács et al., 1993] [Pennefather et al., 1999] [Hansen and Hess, 2006]. In addition, objects can also be identified using their discontinuous fragmented contour segments (lines or Gabor elements) [Panis et al., 2008] [Sassi et al., 2010]. Thus, non-overlapping segments can retain structure information. Most importantly, representing a set of non-overlapping contour segments is

more (computationally) efficient and noise-insensitive compared with representing a complete set of contour points.

Although primitives or salient points of contours are commonly utilised for their representation [Groskey et al., 1990] [Dudek and Tsotsos, 1997] [Berretti et al., 2000], the associated computation is complicated, especially, when large numbers of contours have to be processed. As there has been much research reported on representing objects using "fragmented" contour segments [Zhang and Lu, 2004] [Sun and Super, 2005] [Bai et al., 2008] [Bai et al., 2009] [Wang et al., 2014], we were inspired to do likewise. We first fragment a contour into a set of equal-length segments and then encode the spatial distributions and orientations of these segments. Given that a contour contains a sequence of points: $P_1 \ldots P_n$ with coordinates of $(x_1, y_1) \ldots (x_n, y_n)$, the length of the contour (CL) is computed as:

$$CL = \sum_{i=1}^{n-1} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2}. \tag{9.1}$$

If the length of the segment is set as SL, the contour is then divided into $M = \lfloor CL/SL \rfloor$ segments (see Figure 9.7).

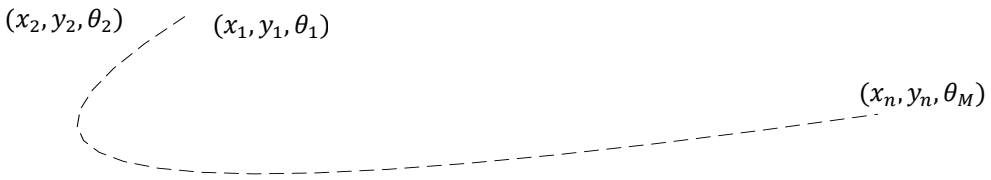
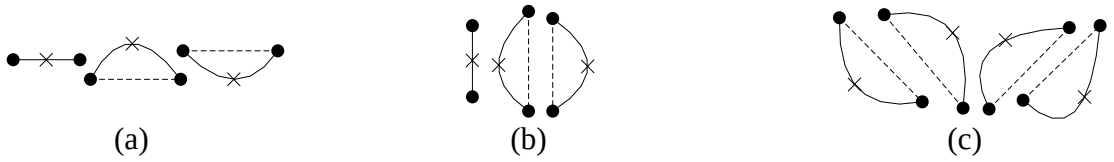


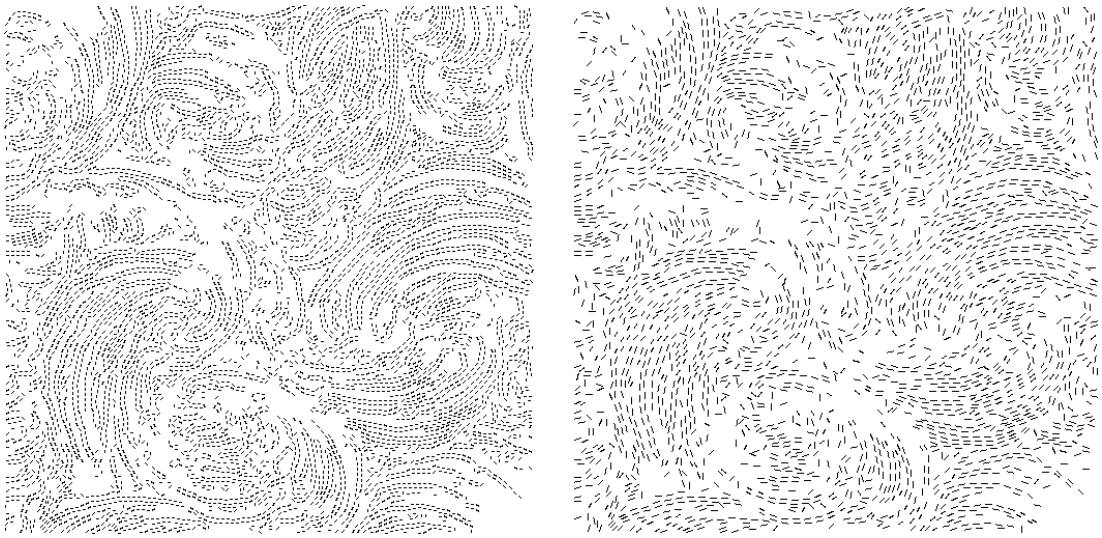
Figure 9.7: One contour is fragmented into a series of equal-length segments. Each segment i is then represented by the position of its mid-point (x_i, y_i) and its chord orientation angle θ_i.

The importance of local orientations to the perception of texture structure has been investigated by Dakin et al. [1997, 1999]. In addition, it was found that objects can also be identified based on the straight-line versions of their outlines [De Winter and Wagemans, 2008B]. Motivated by this research, we represent the segments by their mid-point positions and chord orientation angles θ ($\theta \in (0^o, 180^o]$) (see Figure 9.7). Figure 9.8 presents three sets of typical segment shapes and their approximate chords. The result is a segment map which encodes each contour as a set of labelled segments, i.e. their mid-point positions and chord orientations. However, the shapes of the contours in the segment map will increase in roughness as the length of segments increases (see Figure 9.9). Hence, only short segments with lengths of 3, 5, 7, 9 and 11 pixels were used.



(a) (b) (c)

Figure 9.8: Three sets of typical segment shapes and their approximate chords. The solid lines above represent example contour segments, the solid dots represent segment endpoints, the dotted lines show the chords of the segments, while the crosses show the segment mid-points. The orientations of the chords and the positions of the mid-points are used to represent contours.



(a) (b)

Figure 9.9: Two segment maps obtained from the skeleton map in Figure 9.3 when the length of segments is set at (a) $SL = 5$ *(pixels) and (b)* $SL = 11$ *(pixels) respectively. It is noteworthy that the segments shown are approximated by their chords. Each chord is placed at the middle point of its corresponding segment. In addition, for display purposes, only* $\lfloor 2 \times SL/3 \rfloor$ *central pixels are shown for each chord.*

9.3.3 Encoding Contours' Segment Maps

We use two different approaches to represent the spatial distributions and orientations of contours' segments. In the first we compute an average segment distribution across contours (that is we compute pair-wise segment relationships within contours and then average across all contours in an image). In the second we use the basic aura matrix [Qin and Yang, 2005] to compute segment co-occurrence data with no regard as to

which contour they belong. In the latter case we restrict the pairs to those occurring within a local $L \times L$ neighbourhood.

Encoding the Average Shape of Contours within an Image

Since the orientation difference is regarded as an approximation of the local curvature and can provide better discriminatory power, we use this to encode the change of contour direction. In addition, the distance between the mid-points of M segments (see Figure 9.7) within a contour is also employed to capture their spatial layout. Pair-wise orientation differences and distances are computed for all $(M-1)(M-2)/2$ segment pair combinations.

The contour segment joint histogram (which we refer to as "CSJH", see Figure 9.2 (e) upper) of the orientation differences and distances is accumulated, and is then normalised by the sum of its elements. Note that the angle θ was quantised into $A \in \{18, 36\}$ bins, providing two possible histogram resolutions for texture similarity estimation tasks. It is these histograms that are used to represent individual contours. Also the histograms are averaged across contours to produce a single average contour histogram per image. Of course the final histogram could have been computed without the intermediate step of computing individual contour histograms; however, what is important is that the segment pairs are restricted to those available within single contours.

Representing the Spatial and Angular Distributions of the Segments across Contours

Since the shapes of contours are not regular, the spatial distribution of these is difficult to compute. However, each segment i has been approximated by its mid-point coordinate (x_i, y_i) and its chord orientation angle θ_i. In this feature we compute segment relationships within an image but the mapping of segments to contours is ignored. In this case it is computationally too expensive to compute all pair-wise segment data within an image. Instead we adapt basic aura matrices [Qin and Yang, 2005] to compute segment-to-segment angle and position relationships restricted to a local $L \times L$ neighbourhood.

Given one finite image $f(s)$, $s = (x, y) \in S$, a neighbourhood system $N = \{N_s, s \in S\}$ in which N_s is the $L \times L$ neighbourhood at site s is obtained. Furthermore, a single site neighbourhood system is defined as that only contains a single neighbouring point. In this case, three definitions are given as below.

(i) Aura Measure (AM): [Elfadel and Picard, 1994] Given two subset $A, B \subseteq S$, the AM of A with respect to B, is computed as:

$$m(A, B, N) = \sum_{S \in A} |N_s \cap B|, \tag{9.4}$$

where $|M|$ counts the total number of the elements in M.

(ii) Grey Level Aura Matrix (GLAM): [Elfadel and Picard, 1994] Given that $\{S_i, 0 \leq i \leq G-1\}$ is the grey level sets of $f(s)$, the GLAM of $f(s)$ over N is computed as:

$$A(N) = [a(i,j)] = \left[m(S_i, S_j)\right], \tag{9.5}$$

where G is the number of grey levels in $f(s)$, $S_i = \{s \in S | f(s) = i\}$ is the pixel set whose grey level is i, and $m(S_i, S_j)$ is the AM between S_i and S_j, $0 \leq i, j \leq G-1$.

(iii) Basic Grey Level Aura Matrix (BGLAM): A basic GLAM is a special GLAM and is obtained using a single site neighbourhood system.

Basic aura matrices comprise sets of 2D (co-occurrence) histograms where the axes represent the two grey levels of the pairs of pixels. In our case the axes represent the two angles of the pairs of segments. These angle co-occurrence histograms are generated for different pair sets, where the segment pairs in a pair set are defined by a displacement vector in a similar way to that used for grey level co-occurrence matrices [Haralick et al., 1973]. Thus they represent, for instance, how many pairs of segments exist within an image that are separated by the displacement vector $d = (\Delta x, \Delta y)$ ($|\Delta x|, |\Delta y| \leq \lfloor L/2 \rfloor$, where L is the width of the neighbourhood) and that have angles θ_1 and θ_2. We use the term "basic segment orientation aura matrices" (BSOAMs) to refer to these matrices and their values are used directly in the feature vector. (Note that neighbourhood size was set as $L = 2SL + 1$, where SL is the segment length and $SL \in \{3,5,7,9,11\}$ and therefore the maximum sized neighbourhood considered was 23×23 pixels).

Generating the Contour-Based Feature Vector

The mean of all CSJHs and each BSOAM are concatenated into one feature vector which we refer to as "SDoCS" (spatial distributions of contour segments). We test it at two different segment angle quantisation schemes (using A bins, $A \in \{18, 36\}$) and five different segment lengths ($SL \in \{3,5,7,9,11\}$) plus one multi-scale case ($SL =$ "MS") which concatenates all five feature vectors derived using the five segment lengths.

9.4 Comparison with the Existing Feature Sets

Three hundred and thirty-four textures in the *Pertex* database and the pair-of-pairs based and retrieval based evaluation methods introduced in Chapters 4 and 6 were used to assess the performance of the new contour-based feature set against 52 existing feature sets (51 feature sets as tested in Chapters 5 and 6 and one contour type feature set derived from shape recognition: chain code histogram (CCH) [Iivarinen and Visa, 1996]). It should be noted that the shape context and Fourier descriptor methods were more eligible than the CCH algorithm according to Table 9.1. However, the shape context method yields a large number of features for each contour. The direct concatenation of the features extracted from all contours in a contour map is too long for a feature vector while the higtogram accumulated from these features is dependent on the choice of bins. In addition, Fourier descriptor approaches can only be used on closed contours which makes these approaches unsuitable for representing open contours. Hence, we chose the CCH for comparison due to its popularity.

The performance of the feature sets was assessed by first using these to compute 334×334 texture similarity matrices (see Section 4.3) and then using these matrices in the two tasks. As discussed in Chapters 5 and 6, the multi-resolution scheme can improve the performance of the majority of the 51 feature sets compared with features computed at the original resolution (1024×1024), we therefore only examined the 1024×1024 resolution and the multi-resolution scheme in this section.

9.4.1 Pair-of-Pairs Based Evaluation Experiments

Evaluation Experiment Using $POPJ_{POP}$

When the perceptual pair-of-pairs judgement set: $POPJ_{POP}$ obtained in the "original" pair-of-pairs experiment [Clarke et al., 2012] is used as the ground-truth data, results are obtained and shown for two resolutions in Figure 9.10. The best feature set out of the 51 feature sets tested in Section 5.2.1 is the Multi-resolution Simultaneous Autoregressive Model (MRSAR) [Mao and Jain, 1992]. This is therefore shown separately in Figure 9.10 together with the average performance of the 51 feature sets (as "MeanOf51"). The results of Chain Code Histogram (CCH), are also reported. The remainder of the graph

shows the results for the proposed feature set at two different segment angle quantisation schemes ($A \in \{18, 36\}$) and six different segment lengths ($SL \in \{3,5,7,9,11, MS\}$).

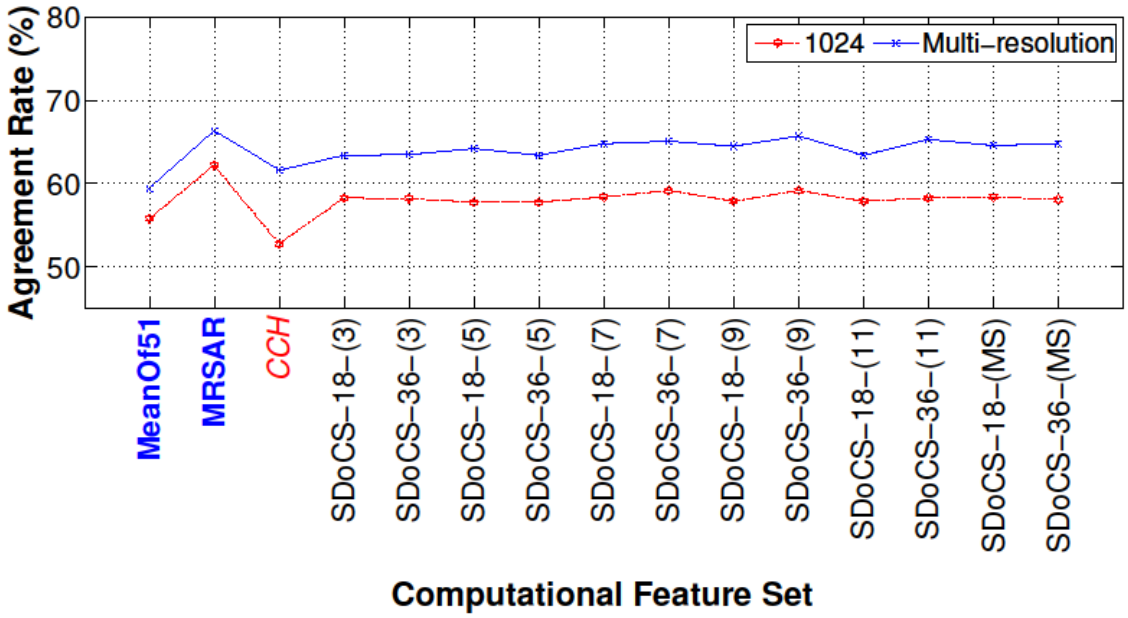


Figure 9.10: Agreement rates of computational features obtained against human pair-of-pairs data: $POPJ_{POP}$ computed at a resolution of 1024×1024 (red trace) and all five resolutions combined (blue trace). The first two columns ("MeanOf51" and "MRSAR") show the mean and best results obtained using the 51 feature sets tested in Section 5.2.1. The next column shows results obtained using Chain Code Histogram (CCH). The remaining results labelled in black "SDoCS-A-SL" are results for our new feature set where the segment angle θ is quantised into A bins ($A \in \{18, 36\}$) and the segment lengths SL are taken from $\{3,5,7,9,11, MS\}$.

It can be observed that (1) the performances of all feature sets are enhanced when the multi-resolution scheme is used; and (2) the proposed feature set normally performs better when segment angle θ is quantised into 36 angle bins than 18. However, it is slightly outperformed by the best conventional feature set: MRSAR.

Evaluation Experiment Using $POPJ_{ISO}$

Figure 9.11 reports results for two resolutions when the perceptual pair-of-pairs judgement set: $POPJ_{ISO}$ constructed from 8D-ISO, is used as the ground-truth data. The best ones out of the 51 feature sets at 1024×1024 resolution and multi-resolution, examined in Section 5.3.1, i.e. Ring and Wedge Filters (RING & WEDGE) [Coggins and Jain, 1985] and Multi-resolution Simultaneous Autoregressive Model (MRSAR) [Mao and

Jain, 1992], are utilised as baselines. Their performances are shown along with the average performance of all of these features (as "MeanOf51"). Furthermore, the results of Chain Code Histogram (CCH) are shown. The rest of the graph displays the results of the proposed feature set at two different segment angle quantisation schemes ($A \in \{18, 36\}$) and six different segment lengths ($SL \in \{3,5,7,9,11, MS\}$).

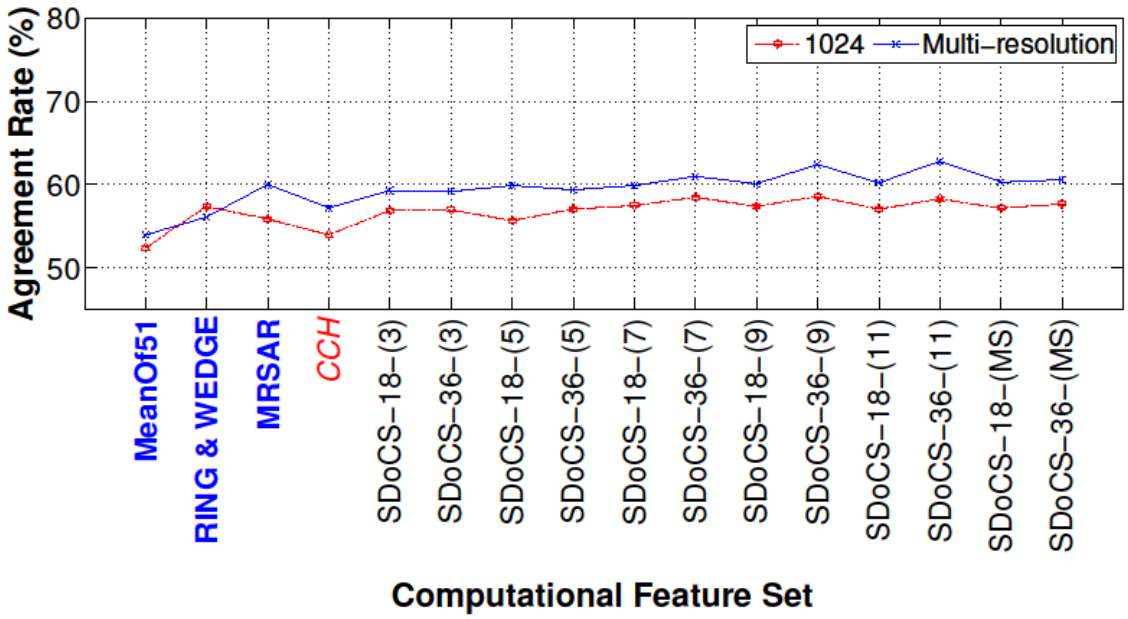


Figure 9.11: Agreement rates of computational features obtained against human pair-of-pairs data: $POPJ_{ISO}$ computed at a resolution of 1024×1024 (red trace) and multi-resolution (blue trace). The first three columns show the mean and two best results obtained using the 51 feature sets tested in Section 5.3.1. The next column shows results of Chain Code Histogram (CCH). The remaining results labelled in black "SDoCS-A-SL" are obtained using our new feature set where the segment angle θ is quantised into A ($A \in \{18, 36\}$) bins and the segment lengths SL are taken from $\{3,5,7,9,11, MS\}$.

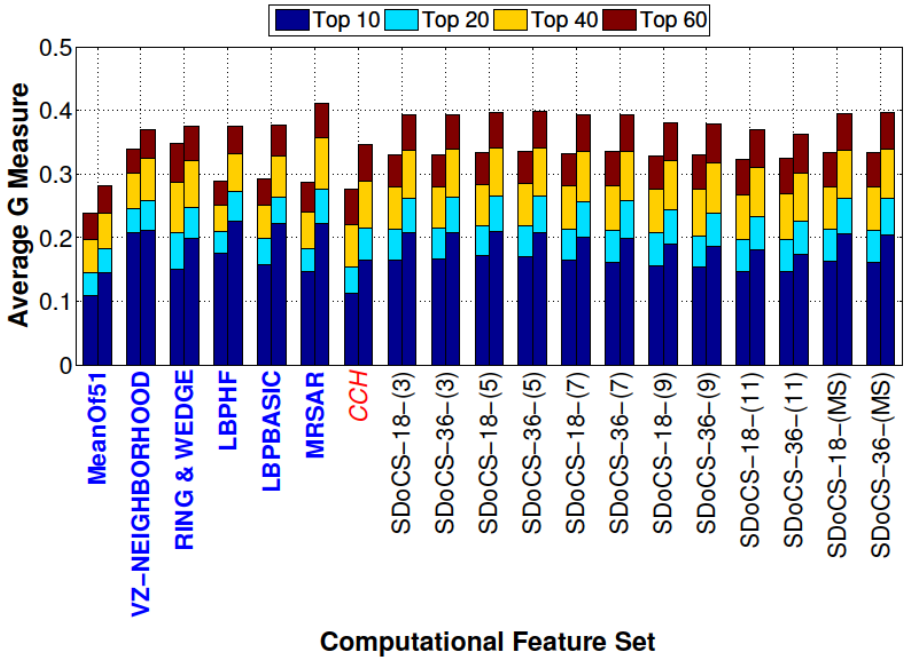
It can be seen that (1) the performances of all feature sets except RING & WEDGE are enhanced by using the multi-resolution scheme; and (2) our feature set performs better when segment angle θ is quantised into 36 bins than 18 and with longer lengths where it outperforms the best conventional feature sets: RING & WEDGE and MRSAR.

9.4.2 Retrieval-Based Evaluation Experiment

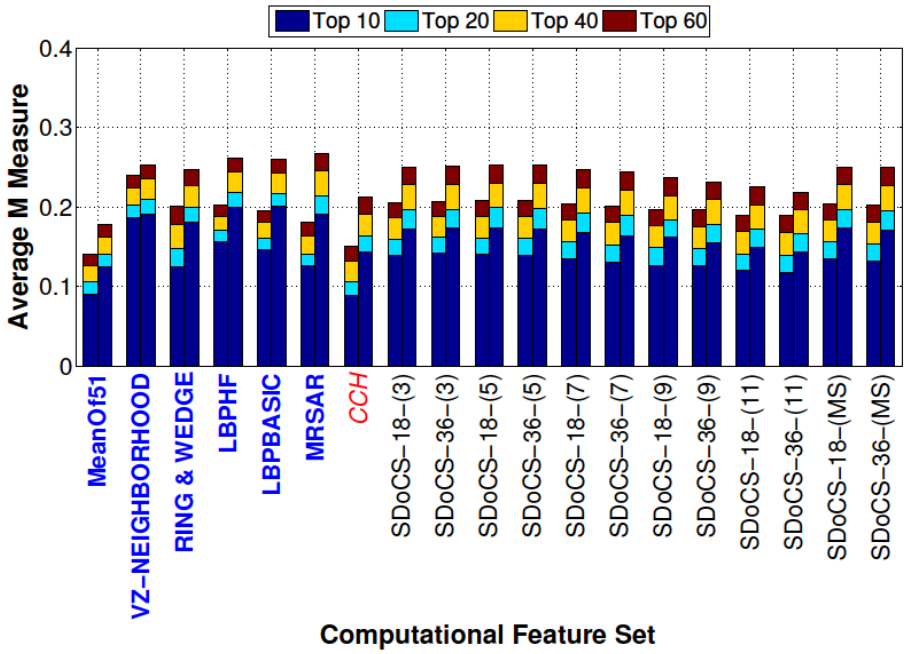
When only 1024×1024 resolution and multi-resolution are used, the five best feature sets examined in Section 6.3, i.e. VZ-NEIGHBORHOOD [Varma and Zisserman,

2009], RING & WEDGE [Coggins and Jain, 1985], MRSAR [Mao and Jain, 1992], LBPBASIC [Ahonen and Pietikäinen, 2009] and LBPHF [Ahonen et al., 2009], are utilised as baselines. Besides, the results of CCH are also reported. The average *G* and average *M* measures are shown in Figures 9.12 (a) and (b), respectively.

It can be observed that: (1) the multi-resolution scheme improves the performance of all these feature sets; (2) when the *G* measure is considered, our feature set outperforms all its counterparts except for the VZ-NEIGHBORHOOD and the RING & WEDGE features at the 1024×1024 resolution, while obtains slightly worse performance than MRSAR at the multi-resolution scheme; and (3) when the *M* measure is examined, our feature set is outperformed by VZ-NEIGHBORHOOD at a resolution of 1024×1024. In addition, it performs slightly worse than VZ-NEIGHBORHOOD, MRSAR, LBPBASIC and LBPHF when the multi-resolution scheme is employed.



(a)



(b)

Figure 9.12: Average G and average M measures of computational features obtained against human ranking data. Each bar shows four different colour-coded results for four values of N ∈{10, 20, 40, 60}. In addition, each bar-group shows two resolutions: 1024×1024 (left), and multi-resolution (right). The first six columns (labelled in blue) show the mean and five best results obtained using the 51 feature sets tested in Section 6.3 at different conditions. The next column shows results obtained using Chain Code Histogram (CCH). The remaining results labelled in black "SDoCS-A-SL" are results for our new feature set where segment angle θ is quantised into A bins (A ∈ {18, 36}) and the segment length SL ∈ {3,5,7,9,11, MS}.

9.5 Conclusions

In this chapter, we developed a new type of texture feature based on representing contours as sets of segments. We refer to this feature set using the title: Spatial Distributions of Contour Segments or "SDoCS" for short. According to the two-stage model that we proposed in Chapter 3, it is notable that the SDoCS exploits both the short-range and long-range HOS (higher order statistics) available from the segments themselves and the segment distributions within contours.

Corresponding to vision science knowledge, segments encode short-range interactions while the spatial distribution of segments within contours captures long-range interactions. This agrees with that stated by Polat [1999] that long-range interactions are formed from a chain of smaller filters connected collinearly. The smaller filters are used to explain local perceptual effect in classical receptive-field concepts [Spillmann and Werner, 1996] while long-range interactions account for global perceptual effects [Spillmann and Werner, 1996] [Nurminen et al., 2010].

Furthermore, the proposed feature set considers the spatial and orientation distributions of segments within a contour as well as across contours, even if it does not encode the spatial relationship of contours directly. Since SDoCS employs approximate segments (i.e. chords) obtained from contours, it can be regarded as generative. In addition, it is also noise-insensitive because it utilises segment chords which are less influenced by noise. The proposed method generates a feature vector and thus avoids the direct matching of two contour maps. Hence, it is more efficient than matching-based methods. As a result, SDoCS satisfies the four criteria introduced in Section 9.2.

We assessed the SDoCS feature set using the two tasks introduced in Chapters 5 and 6. The results showed that SDoCS outperformed or performed comparatively with the other feature sets in the two tasks. We feel that the key point, however, is that we have shown the usefulness of long-range HOS in computing texture similarity and hope that this will inspire other developments of texture features based on such information.

Chapter 10

Conclusions and Future Work

Since texture similarity is context-dependent and a dimensional model [Rao, and Lohse, 1993 & 1996] [Cho et al., 2000] is not appropriate [Heaps and Handel, 1999], the research on perceptual texture similarity based on several texture properties [Tamura et al., 1978] [Amadasun and King, 1989] [Fujii et al., 2003] [Abbadeni, 2011] is considered as limited. In addition, Boolean-valued perceptual texture similarity is normally used for texture segmentation, classification and retrieval tasks while higher resolution texture similarity has received fewer attentions. Inspired by these studies, this thesis investigated the estimation of higher resolution perceptual texture similarity using a large pool of computational features rather than modelling several texture properties. The main contribution of this study is to combine vision science knowledge and computer vision techniques for estimating perceptual texture similarity.

10.1 Contributions

The contributions of this thesis are summarised as follows.

- **Discriminating Higher Resolution Texture Similarity from the Boolean-Valued Texture Similarity**

In contrast with Boolean-valued texture similarity which is normally used by texture classification, segmentation and retrieval tasks, we researched higher resolution texture similarity, including pair-of-pairs judgements and rankings. To the best of our knowledge, previous studies have not made this important distinction.

- **A Review of 51 Computational Feature Sets under a Two-Stage Feature Extraction Model**

We first proposed a two-stage feature extraction model in Chapter 3 and used this to examine the spatial extent and the order of statistics employed by 51 computational feature sets. It was concluded that: (1) the filtering-based features except JSCW [Portilla and Simoncelli, 2000] do not capture aperiodic long-range interactions; and (2) the majority of the other feature sets compute higher order statistics only from small local neighbourhoods and do not exploit aperiodic long-range interactions either. Therefore, we conclude that these 51 feature sets are unable to exploit aperiodic long-range interactions.

- **Introducing Two Different Methods for Evaluating Computational Features on the Estimation of Perceptual Texture Similarity and Conducting Two Series of Evaluation Experiments Correspondingly**

In Chapter 3, we introduced a new pair-of-pairs based evaluation framework for benchmarking computational texture features. The framework possesses four merits: (1) it utilises higher resolution perceptual texture similarity obtained from a texture database of 334 textures as the ground-truth; (2) it extends the spatial extent exploited by the computational features using a multi-pyramid decomposition; (3) it is able to evaluate computational pair-of-pairs similarity judgements against their perceptual counterparts; and (4) it uses the performance measure: agreement rate. In Chapter 5, we evaluated the ability of the computational features to estimate perceptual pair-of-pairs judgements in two experiments.

We also proposed a retrieval-based evaluation method which compares computational and perceptual texture rankings. A retrieval-based evaluation experiment was carried out using the same 51 feature sets.

However, the results of both sets of evaluation experiments showed that the feature sets did not perform consistently with human observers. Since the 51 feature sets cannot exploit aperiodic long-range interactions and humans can utilise these in other tasks [Field et al., 1993] [Spillmann and Werner, 1996], we hypothesised that long-range interactions are important when humans judge texture similarity.

- **Providing Indirect Proofs of the Importance of Long-Range Interactions to Perceptual Texture Similarity**

Non-randomised blocked and randomised blocked images were used in two modified pair-of-pairs experiments in Chapter 7. It was found that the experiment that used randomised blocked images (POP_R) produced significantly less agreement with the original experiment (POP_O) than the non-randomised blocked experiment (POP_N). Furthermore, experimental results showed that when human observers were presented with the original images, they agreed less with the computational results than that they did when these long-range interactions have been removed by block randomisation. As a result, we believe that long-range interactions are important when humans judge the similarity of textures.

- **Confirming the Importance of the Contour to the Perception of Texture and Developing a Set of Contour-Based Texture Features**

In Chapter 8, we asked human observers to compare 334 texture images in *Pertex* with their three different types of property images. Experimental results showed that the contour map is the most important for the perception of the 334 textures. It is well-known that humans are adept at exploiting the long-range visual interactions evident in contour information [Field et al., 1993] [Pettet et al., 1998] [Hansen and Hess, 2006], and this may explain why the contour map is the most representative among the three image properties. In addition, a retrieval-based evaluation experiment was carried out using the 247 textures which can be represented well by their contour maps. However, none of the 51 feature sets used were found to be able to exploit contour information as well as humans did.

Inspired by this, we developed a new set of contour-based texture features which performed well on the ranking data and better with the pair-of-pairs data, compared with the 51 feature sets and a shape recognition type feature set. The new feature set can not only encode the average shape of the contours in one texture but also the spatial and angular distributions of their segments. We attribute this to the fact that the proposed feature set is able to encode long-range interactions.

10.2 Future Work

This research has identified several open problems.

- **Acquisition of a Long-Range Structural Texture Database**

In Chapter 3, it was concluded that the 51 existing feature sets examined in this research cannot exploit aperiodic long-range interactions. However, the *Pertex* texture database used in this study contains both periodic and aperiodic textures. In order to obtain more significant results, a texture database that contains a large number of aperiodic long-range structural textures is required. In addition, higher resolution perceptual texture similarity data should be derived for this database.

- **More Accurate and Efficient Texture Contour Extraction**

In this study, we used the Canny edge detector [Canny, 1986] to extract edges of contours from texture images due to its simplicity and effectiveness. However, contour maps cannot be correctly extracted from the texture images using the Canny edge detector in the case that texture granularity is too small or the contrast between the foreground and background is low. Although some other contour extraction approaches [Malik et al., 2001] [Guo et al., 2007] [Arbelaez et al., 2011] can also be utilised, the computational complexity would be significant for 1024×1024 texture images. Therefore, for future work a more accurate and efficient texture contour extraction algorithm is required. In addition, contours can be broken during the extraction process. As a result, contour grouping and linking are also required.

- **The Effect of the Length of Contour Segments on the Perception of Texture Contour**

We empirically fragmented contours into equal-length segments and employed their mid-points and chord orientations to approximate their shapes in Chapter 9. However, optimal length (i.e. spacing between two mid-points along the contour) of segments is unknown. The importance of the spacing between segments to contour integration for human perception has been addressed by Pennefather et al. [1999] and Kovács et al. [2000]. Similarly, the effect of the spacing between segment mid-points on the percep-

tion of texture contour is also interesting. The impact of these factors on the perception of texture should be investigated.

- **Encoding Higher Order or Sparse but Effective 2nd-Order Spatial Distributions of Contour Segments**

In Chapter 9, we utilised the basic aura matrix (BAM) to capture the spatial and angular distributions of contour segments without regard to the contours that they belonged to. In essence, one BAM is a subset of a dipole histogram. Although a dipole histogram (all possible basic aura matrices) can uniquely determine one finite image [Chubb and Yellott, 2000], the computational complexity for obtaining a complete dipole histogram is unacceptable in practice. Thus, only small, local neighbourhoods were used to compute basic aura matrices in the proposed approach. As an incomplete dipole histogram, the basic aura matrices obtained in this way cannot encode a finite image adequately. In this case, a sparse but more effective 2nd-order statistic is required for more effective encoding of the spatial distributions of contour segments. As an alternative, global higher order statistics could also be extracted from contour segments. In our future work, we will also dedicate to solving this problem.

Appendices

Appendix A

Descriptions of the 14 Clusters of 334 *Pertex* Textures

Clusters	Description
Cluster 1	Regular or nearly-regular uni-directional textures
Cluster 2	Uni-directional textures without an obvious periodicity
Clusters 3 & 4	Regular or nearly-regular bi-directional textures. The texture scale of Cluster 3 is bigger than Cluster 4
Cluster 5	Regular zig-zag-like textures
Cluster 6	Nearly-regular bi-directional textures. The scale (granularity) of this cluster of textures is smaller than those of textures in Clusters 3 and 4
Cluster 7	Irregular textures but with two dominant directions
Cluster 8	Random noise textures. There is normally no obvious structure in these textures
Cluster 9	Blob-like or structural textures (both are also random textures). The biggest difference between this cluster and Cluster 8 is that there are blobs, or blocks, or irregular structures in textures
Cluster 10	Swirly textures (are also random textures)
Cluster 11	Blob-like or random structural textures. These textures are similar with those in Cluster 9 but look like more regular
Cluster 12	three wave-like textures. These textures have characteristics between uni-directional and swirly textures
Cluster 13	Textile-like textures. These textures are regular or nearly-regular but the contrast (or height difference) between the foreground and back-ground is small
Cluster 14	Two random textures

Appendix B

Experimental Results of Chapter 5

Method	Resolution					
	1024	512	256	128	64	Multi
ACF	55.7	56.0	56.4	54.9	56.3	59.6
CVM	56.5	56.8	56.7	58.8	53.7	57.2
DCT	55.8	54.1	54.4	52.1	59.0	56.6
EIGENFILTER	56.9	54.8	55.8	55.1	54.0	57.5
FRACTALDIMENSION	51.2	49.9	51.3	53.0	49.1	51.2
GABORBOVIK	49.7	48.8	50.1	52.0	56.2	54.0
GABORENERGY	55.9	54.6	55.2	65.6	58.7	59.1
GABORJFFD	59.8	60.5	61.6	60.1	56.5	59.8
GABORJFSD	53.9	54.8	56.5	58.3	63.8	57.9
GABORMM	49.7	58.2	61.3	65.3	58.7	55.5
GLADH	56.6	59.0	59.7	60.8	61.9	63.8
GLCM	55.5	60.1	59.0	58.1	52.4	61.0
GLSDH	57.8	58.0	59.6	59.5	62.2	63.7
GLSDSH	53.8	55.0	57.3	58.6	62.6	59.6
GLGLM	56.6	57.2	56.2	55.6	53.2	57.5
GLH	55.3	57.3	56.1	55.6	54.4	60.1
GLRLM	56.4	57.4	61.7	60.5	58.3	57.3
GLSH	53.8	55.0	57.3	58.6	62.6	59.6
GMRF	54.2	56.6	55.3	54.7	54.5	57.4
RING & WEDGE	58.8	57.2	55.8	60.7	65.1	63.0
LAWS	55.9	55.7	58.3	56.8	58.0	61.2
LM	57.5	56.4	63.1	**♣66.5**	57.5	60.9
MR8	56.3	57.8	59.5	63.1	57.1	62.7
MRSAR	**62.2**	60.9	62.2	62.0	**66.3**	**66.3**
MSA	*48.6*	53.6	54.3	57.4	61.0	48.6
RFS	55.8	56.3	57.8	65.3	60.5	60.7
S	55.1	54.0	56.6	60.9	55.9	58.7
JSCW	61.2	60.0	58.1	52.4	50.3	60.7
SRDM	50.6	♥*46.0*	51.3	53.8	54.4	55.3
TT	55.5	56.5	57.0	58.1	58.8	55.5
GDIRCANNY	53.4	54.3	59.6	61.4	60.9	60.4
GDIRSOBEL	52.3	53.8	54.4	59.7	62.5	58.7
GMAGGDIRCANNY	58.2	**61.2**	63.0	63.9	56.4	65.1
GMAGGDIRSOBEL	53.1	58.1	61.1	59.5	59.7	63.4
GMAGCANNY	57.6	60.3	59.3	58.9	54.7	64.2
GMAGSOBEL	56.0	58.4	60.4	58.6	57.7	63.0
LBPRIU2	56.6	55.3	55.0	61.2	61.7	62.2
LBPBASIC	57.2	55.8	56.9	58.9	62.8	60.2
LBPDF	57.4	55.5	56.5	58.9	62.0	60.3
LBPHF	58.1	58.2	57.7	65.3	64.5	64.7
LBPRIU2 & VAR	57.1	57.1	62.4	58.7	57.8	63.0
LDP	57.6	58.6	55.0	54.8	61.3	58.5
LDPSE	57.4	57.9	55.2	54.8	61.3	58.2
RI-LPQ	55.9	54.0	56.4	62.9	63.6	62.1
VZ-MR8	59.3	60.6	61.2	63.3	56.2	63.0
VZ-MRF	58.6	58.3	57.1	56.0	60.8	59.6
VZ-NEIGHBORHOOD	57.5	57.6	56.7	56.0	59.9	59.6
SAC	55.0	54.3	54.3	57.3	60.1	56.4
SRAC	57.0	54.8	55.7	59.7	61.3	57.8
SVR	49.8	46.3	*47.5*	*50.6*	*46.9*	*46.9*
VAR	57.5	58.9	**63.2**	56.6	55.4	62.3

***The blue bold and underlined font means the maximum at each resolution, and ♣ means the highest over six resolutions**

**The red italic font stands for the minimum at each resolution, and ♥ represents the lowest over all resolutions*

Table B.1: Agreement rates (%) between the pair-of-pairs judgements obtained using 51 sets of computational features at five individual resolutions: 1024×1024, 512×512, 256×256, 128×128, 64×64 and multi-resolution ("Multi") and the perceptual pair-of-pairs judgement set $POPJ_{POP}$ obtained using pair-of-pairs experiments directly.

Method	Resolution					
	1024	512	256	128	64	Multi
ACF	52.3	51.3	53.4	53.1	52.3	53.2
CVM	55.8	56.1	57.3	58.3	52.5	54.7
DCT	51.6	51.1	51.7	*49.1*	57.2	52.9
EIGENFILTER	53.3	51.4	51.5	53.8	51.8	52.6
FRACTALDIMENSION	51.2	50.1	51.6	50.8	49.5	51.7
GABORBOVIK	*49.0*	*48.5*	50.4	52.7	52.6	51.7
GABORENERGY	55.3	50.7	55.9	58.7	53.1	53.9
GABORJFFD	53.3	54.5	56.2	56.0	50.0	53.3
GABORJFSD	50.4	49.9	51.9	52.8	57.6	51.3
GABORMM	50.8	53.6	56.6	58.6	52.5	51.8
GLADH	52.3	53.5	55.0	54.5	56.2	56.5
GLCM	51.0	55.7	54.5	54.2	48.7	54.8
GLSDH	54.5	55.9	54.2	55.5	54.7	56.3
GLSDSH	50.8	53.5	53.3	54.8	57.0	55.2
GLGLM	56.2	55.0	55.3	53.3	51.1	56.9
GLH	51.2	52.0	53.4	51.6	49.8	53.4
GLRLM	52.1	54.2	54.1	54.7	52.5	52.8
GLSH	50.8	53.5	53.3	54.8	57.0	55.2
GMRF	50.7	49.7	51.0	51.3	54.7	51.2
RING & WEDGE	**57.4**	53.4	50.7	55.8	58.7	56.1
LAWS	53.5	52.0	51.8	53.5	54.2	54.1
LM	53.6	52.9	58.3	58.8	54.8	56.0
MR8	52.7	53.5	53.1	55.6	53.6	55.7
MRSAR	55.9	**57.8**	58.0	58.0	♣**60.7**	**60.0**
MSA	50.1	50.5	51.1	53.9	54.3	50.1
RFS	52.1	53.6	53.3	58.0	56.5	56.2
S	50.8	49.4	50.4	54.2	51.3	52.2
JSCW	55.5	54.7	54.9	49.7	♥*46.5*	53.3
SRDM	50.3	51.1	**58.4**	54.2	52.5	53.5
TT	54.5	54.4	53.5	54.1	55.1	54.5
GDIRCANNY	52.1	52.1	55.8	57.6	56.7	55.0
GDIRSOBEL	52.1	52.5	51.7	55.9	56.2	53.5
GMAGGDIRCANNY	52.8	54.3	56.7	57.1	56.4	55.3
GMAGGDIRSOBEL	49.9	52.2	54.4	55.7	56.9	55.9
GMAGCANNY	51.4	53.9	55.3	54.0	53.9	55.8
GMAGSOBEL	53.6	51.6	53.8	54.7	54.8	55.3
LBPRIU2	51.2	53.9	50.4	55.7	56.0	53.1
LBPBASIC	52.6	54.1	54.1	53.7	58.2	54.9
LBPDF	51.8	52.9	53.2	53.1	57.8	54.2
LBPHF	54.4	56.4	53.8	**59.0**	57.2	56.3
LBPRIU2 & VAR	51.8	51.4	54.2	55.4	51.1	54.5
LDP	53.9	52.9	53.2	51.8	58.9	53.6
LDPSE	54.4	54.0	53.8	52.1	58.5	53.6
RI-LPQ	50.2	49.4	50.8	58.5	54.0	54.0
VZ-MR8	51.3	51.2	51.5	56.8	53.6	53.5
VZ-MRF	51.1	51.1	52.1	50.5	56.0	52.1
VZ-NEIGHBORHOOD	51.4	50.4	51.9	51.1	55.7	51.7
SAC	53.1	55.8	54.3	51.2	54.5	53.4
SRAC	51.6	55.2	53.3	52.6	54.5	52.6
SVR	52.8	49.8	*49.6*	50.8	51.0	*49.3*
VAR	52.8	50.8	56.3	55.1	50.3	53.7

*__The blue bold and underlined font means the maximum at each resolution, and ♣ means the highest over six resolutions__

**The red italic font stands for the minimum at each resolution, and ♥ represents the lowest over all resolutions*

Table B.2: Agreement rates (%) between the pair-of-pairs judgements obtained using 51 sets of computational features at five individual resolutions: 1024×1024, 512×512, 256×256, 128×128, 64×64 and multi-resolution ("Multi") and the perceptual pair-of-pairs judgement set $POPJ_{ISO}$ constructed from the 8D-ISO similarity matrix.

Appendix C

Is the Power Spectrum More Important to Computational Features than the Phase Spectrum?

C.1 Introduction

As we discussed in Section 3.3, filtering-based features, except quadrature filters based features, normally work in the power spectrum but ignore the phase information, when the premise of Parseval's theorem [Weisstein] is satisfied. However, the aperiodic image structure is mainly retained in the phase spectrum [Oppenheim and Lim, 1991]. Since none of the 51 computational feature sets examined in Chapters 5 and 6 agreed with human observers on estimating texture similarity well, it is possible that the rest of the feature sets aside from the filtering-based features do not exploit the phase information as well. Thus, our hypothesis is that the power spectrum is more important to computational features than the phase spectrum. We test this hypothesis in this appendix.

We managed to remove the original phase spectrum and the original power spectrum of one texture image by replacing these using a white noise matrix and a single value matrix respectively (see Section 2.5.1 for more details). Correspondingly, two sets of property images: phase-randomised (power-only) images and power-uniformised (phase-only) images were obtained. Two sets of pair-of-pairs judgements: $POPJ_{EPR}$ and $POPJ_{EPU}$ were then obtained for each feature set at one of six resolutions (including multi-resolution). Given that the six pair-of-pairs judgement sets derived from the origi-

nal texture images, i.e. $POPJ_E$, were used as the baseline, two agreement rates between $POPJ_{EPR}$ or $POPJ_{EPU}$ and the baseline were computed for each feature set when a resolution was considered.

A factorial repeated-measures ANOVA was conducted on the agreement rates. If the computational pair-of-pairs judgements obtained from the phase-randomised images are significantly more in agreement with the computational pair-of-pairs judgements derived from the original texture images than those obtained from the power-uniformised images, the power spectrum is likely to be more important to computational features than the phase spectrum.

C.2 Experiment

Given a set of property images (phase-randomised or power-uniformised images), the method introduced in Section 4.3 was used to obtain a set of pair-of-pairs judgements for each computational feature set under a resolution (including multi-resolution). In total, two sets of (51 dimensional) pair-of-pairs judgements: $POPJ_{EPR}$ and $POPJ_{EPU}$ were obtained at each resolutions (including multi-resolution).

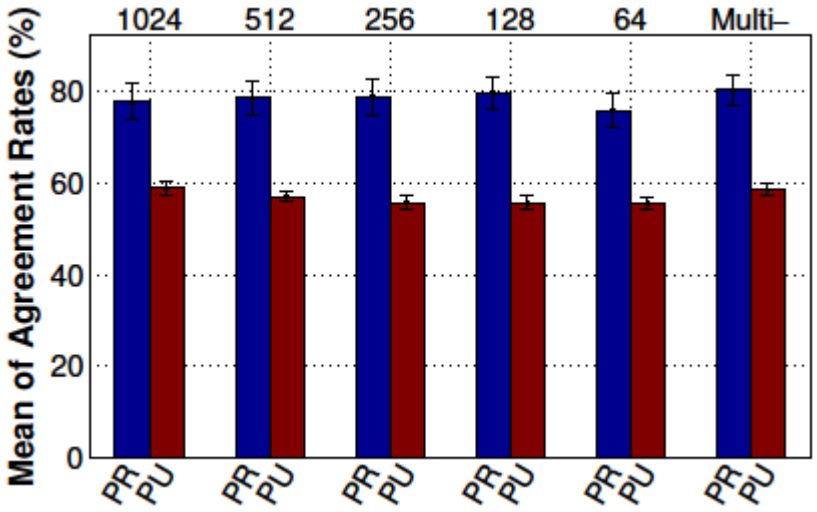


Figure C.1: Means and 95% confidence intervals (error bars) of the agreement rates between the computational pair-of-pairs judgements obtained from the phase-randomised (blue bars) images and power-uniformised images (red bars) and those obtained from the original images over the 51 feature sets at six different resolutions.

The six pair-of-pairs judgement sets derived from the original texture images under six resolutions, i.e. $POPJ_E$, were used as the baseline. Given a resolution, two agreement

rates between $POPJ_{EPR}$ or $POPJ_{EPU}$ and the baseline were computed for each feature set. As a result, 12 (2×6) sets of 51 dimensional agreement rates were derived. The means of the agreement rates over the 51 feature sets at different resolutions are reported in Figure C.1. It can be seen that the computational pair-of-pairs judgements obtained from the phase-randomised images agree with the computational pair-of-pairs judgements derived from the original texture images more than those obtained from the power-uniformised images, at six different resolutions.

Given that the 51 feature sets are considered as a population, the agreement rates obtained using these can be considered as the dependent variable while the resolution and the type of property images are regarded as the independent variables of a factorial repeated-measures ANOVA (Analysis of Variance, $\partial = 0.05$).

According to the results of Mauchly's test [Mauchly, 1940] [Field, 2009], the assumption of sphericity was violated for the main effects of the resolution, $\chi^2(14) = 102.09$, $p < 0.05$, and the interaction between the resolution and the type of property images, $\chi^2(14) = 96.96$, $p < 0.05$, while the assumption of sphericity was satisfied for the main effects of the type of property images, $\chi^2(0) = 0.00$. Degrees of freedom were therefore corrected using Greenhouse-Geisser estimates of sphericity [Greenhouse and Geisser, 1959] ($\varepsilon = 0.53$ for the main effects of the resolution and 0.55 for the interaction effect between the resolution and the type of property images). We, therefore, report the three effects derived from this analysis as below:

(1) The results show a significant main effect of the resolution on the agreement rate, $F(2.63, 131.64) = 7.83$, $p < 0.05$;

(2) The significant main effect of the type of property images on the agreement rate is also found, $F(1, 50) = 162.56$, $p < 0.05$. Contrasts reveal that the agreement rates obtained using phase-randomised images were significantly higher than those obtained using power-uniformised images, $F(1, 50) = 162.56$, $r = 0.87$; and

(3) There was a significant interaction effect between the resolution and the type of property images, $F(2.73, 136.43) = 5.60$, $p < 0.05$. It is indicated that the type of property images generated different effects on agreement rates with the changing of the resolution.

C.3 Conclusions

In this appendix, we investigated whether or not the power spectrum is more important to computational feature sets than the phase spectrum. It was found that the type of property images has a significant main effect on the agreement rates between the computational pair-of-pairs judgements obtained from different types of property images with those obtained from the original texture images. Furthermore, contrasts suggest that the agreement rates obtained using phase-randomised images were significantly higher than those obtained using power-uniformised images. In other words, the computational pair-of-pairs judgements obtained using the 51 feature sets from the power-only (phase-randomzied) images are more in agreement with the computational pair-of-pairs judgements obtained using these feature sets from the original images than those obtained from the phase-only (power-uniformised) images. It indicates that the power spectrum is more important to those feature sets than the phase spectrum. However, the power spectrum does not retain aperiodic image structure [Oppenheim and Lim, 1991]. Therefore, none of the 51 feature sets are able to encode aperiodic texture structures and thus cannot capture aperiodic long-range interactions. This probably explains why none of the 51 computational feature sets agreed well with humans on estimating texture similarity.

Appendix D

Experimental Results of Chapter 6

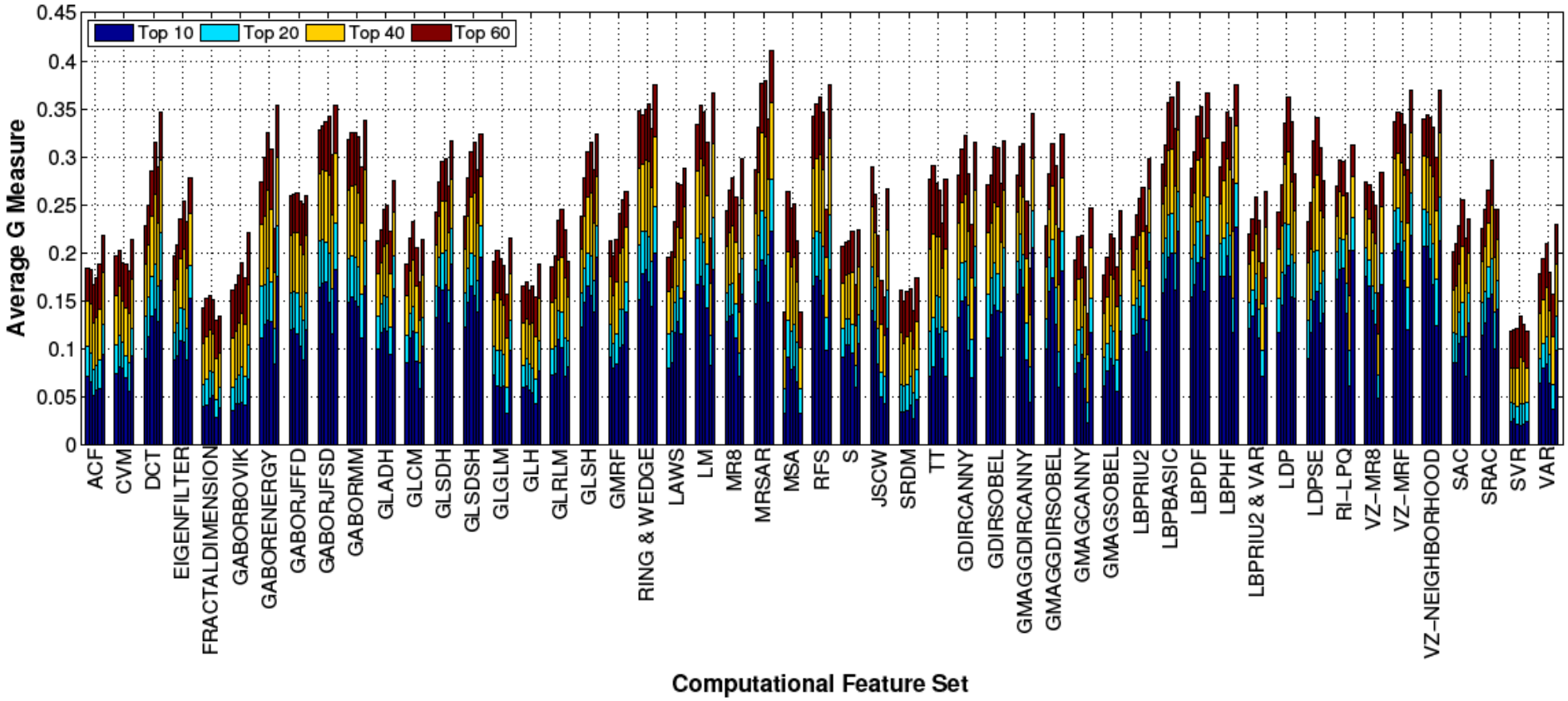


Figure D.1: Average G measures obtained using 51 computational feature sets. Each bar-group shows six resolutions: 1024×1024, 512×512, 256×256, 128×128, 64×64 and multi-resolution (from left to right). Each bar shows four different, colour-coded results for the four values of the retrieval set size $N \in \{10, 20, 40 \text{ and } 60\}$. To be specific, the average G measures mainly lie in 0.26±0.07, when top 60 textures are retrieved.

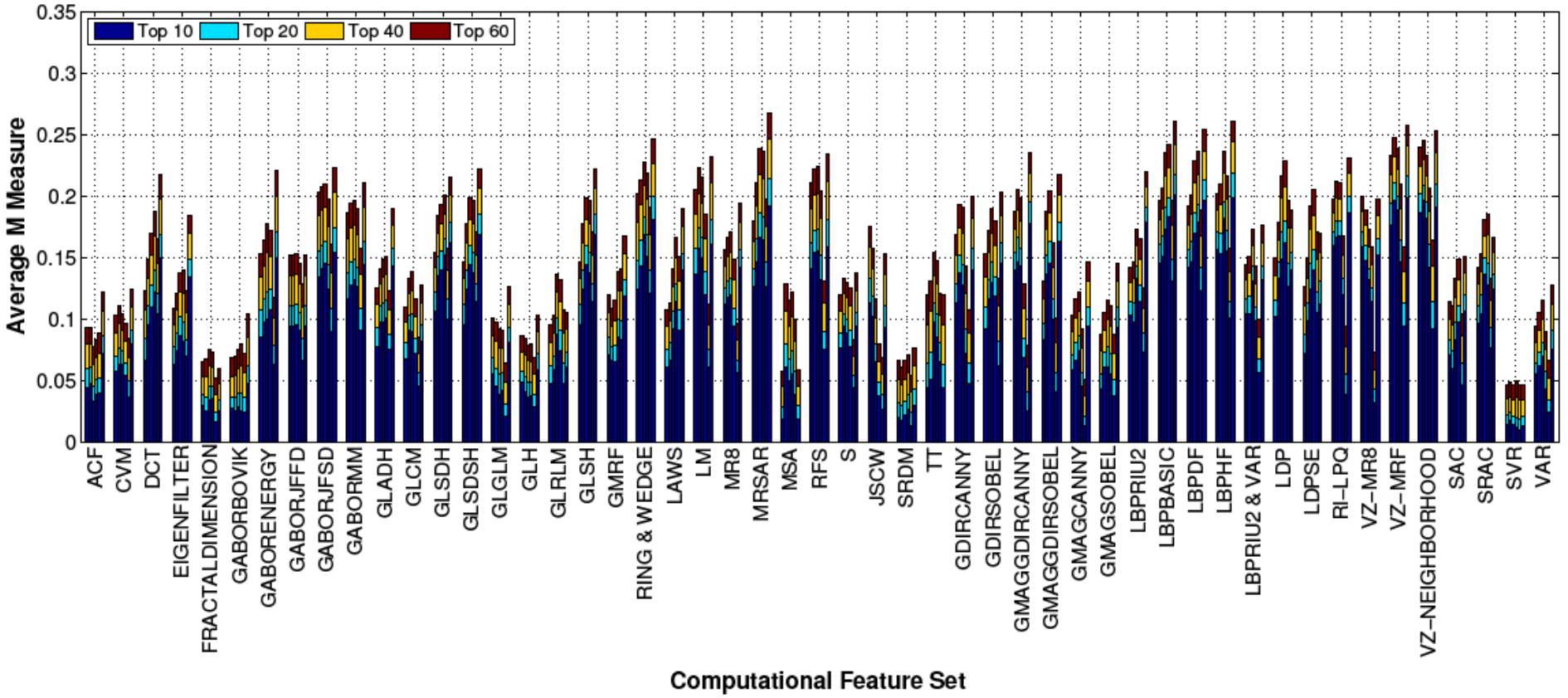


Figure D.2: Average M measures obtained using 51 computational feature sets. Each bar-group shows six resolutions: 1024×1024, 512×512, 256×256, 128×128, 64×64 and multi-resolution (from left to right). Each bar shows four different, colour-coded results for the four values of the retrieval set size $N \in \{10, 20, 40 \text{ and } 60\}$. Specifically, the average M measures mainly distribute in 0.15±0.05, when top 60 textures are retrieved.

Appendix E

Selecting the 80 Most Inconsistent Pairs of Pairs

E.1 Introduction

None of the 51 computational feature sets agreed well with the perceptual pair-of-pairs judgements in both evaluation experiments conducted in Chapter 5. In this appendix, we intend to select the top *N* most inconsistent pairs of pairs, where the disagreements between the majority of the 51 feature sets and the majority of the human observers in the (original) pair-of-pairs experiments [Clarke et al., 2012] reached to the greatest. These pairs of pairs will be used in the experiments reported in Chapter 7.

E.2 Criteria for the Selection

The optimal performance (see Figure 5.3) of each computational feature set against the perceptual pair-of-pairs judgement set $POPJ_{POP}$ was employed to select the most inconsistent pairs of pairs. The reason is that the "failed" cases along with the optimal performance would provide significant insights.

First of all, the disagreement between the $J_E(i)$ (see Equation (4.12)) obtained using one feature set and $J_{POP}(i)$, $i = 1, 2, \ldots, 1000$ (see Equation (4.1)) needs to be large. Let $GAP^i_{POP\&C}$ denote the disagreement between the pair-of-pairs judgements obtained by the observers in the (original) pair-of-pairs experiments [Clarke et al., 2012] and those obtained using a feature set on the i-th pair of pairs. $GAP^i_{POP\&C}$ was computed as:

$$GAP^{i}_{POP\&C} = (J_{POP}(i) - J_E(i)), i = 1, 2, \dots, 1000. \quad \text{(E.1)}$$

Given one computational feature set, the threshold $T_{POP\&C}$ was first applied on the disagreement $GAP^{i}_{POP\&C}$ for each pair of pairs. As a result, all pairs of pairs on which the expression $GAP^{i}_{POP\&C} \leq T_{POP\&C}$, $i = 1, 2, \dots, 1000$ holds true were left out for the current features set and the remaining pairs of pairs were stored temporarily.

Then, the occurrence of the feature sets that disagree with each perceptual pair-of-pairs judgement in $POPJ_{POP}$ was accumulated from the results obtained in the previous step. The accumulated frequencies n_i, $i = 1, 2, \dots, 1000$ are presented in Figure E.1. Since we are only concerned with the most inconsistent pairs of pairs on which the majority (e.g. > 30) of the 51 feature sets disagreed with the observers in the (original) pair-of-pairs experiments [Clarke et al., 2012], a second threshold T_n was applied on the 1000 frequencies, in order to remove the pairs of pairs on which only a small number of feature sets failed to agree with the observers. Only the pairs of pairs whose n_i satisfies $n_i > T_n$, $i = 1, 2, \dots, 1000$ were preserved.

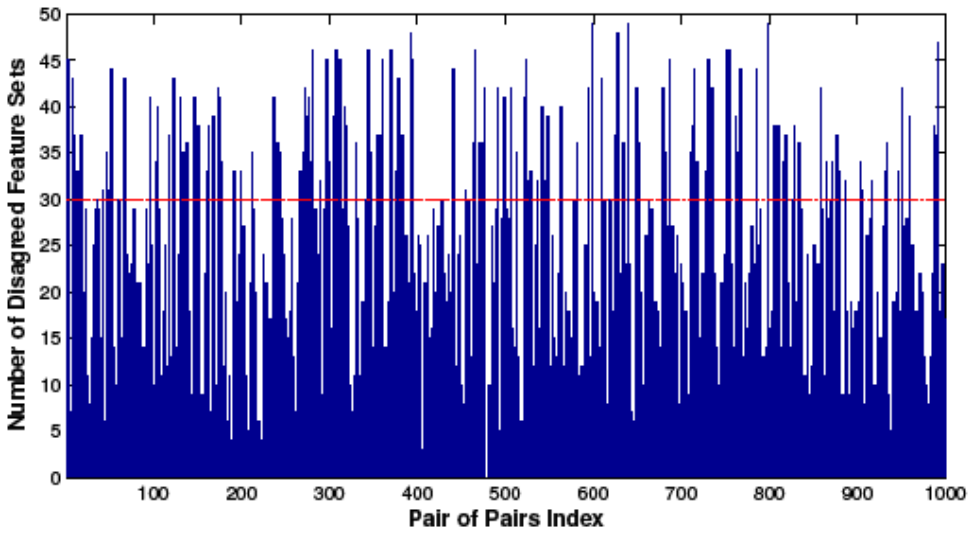


Figure E.1: Numbers of the disagreed feature sets with the 1000 perceptual pair-of-pairs judgements in $POPJ_{POP}$ after the threshold $T_{POP\&C}$ was applied. Another threshold ($T_n = 30$, see the red line) was then utilised on these numbers in order to obtain the most inconsistent pairs of pairs for over 30 computational feature sets.

However, only $T_{POP\&C}$ and T_n are not enough because it is also necessary to guarantee that (1) the same pair is chosen by using $J_{POP}(i)$ (see Equation (4.1)) and $J_{ISO}(i)$, $i = 1, 2, \dots, 1000$ (see Equation (4.3)) and (2) the agreement between the observers in the free-grouping [Halley, 2011B] and the pair-of-pairs experiments is high. In other

words, only the pairs of pairs, on which (1) the majority of the 51 feature sets did not agree with the observers in the pair-of-pairs experiments while (2) the observers in the free-grouping experiments were agreed with the observers in the pair-of-pairs experiments, will be chosen. Therefore, the threshold $T_{POP\&ISO}$ was introduced.

Given that $GAP^i_{POP\&ISO}$, $i = 1, 2, \dots, 1000$ denotes the disagreement between the $J_{POP}(i)$ and $J_{ISO}(i)$ on the i-th pair of pairs. $GAP^i_{POP\&ISO}$ was calculated as:

$$GAP^i_{POP\&ISO} = \left(J_{POP}(i) - J_{ISO}(i)\right), i = 1, 2, \dots, 1000. \quad \text{(E.2)}$$

All pairs of pairs that allow the expression $\left(J_{POP}(i) \times J_{ISO}(i) > 0\right) \&\& (GAP^i_{POP\&ISO} < T_{POP\&ISO})$ hold true were finally chosen as the most inconsistent pairs of pairs for the majority of the 51 sets of computational features.

In addition, considering that these pairs of pairs will be used in the "modified" pair-of-pairs experiments conducted in Chapter 7, the number of these should not be large (which would make these experiments to be time-consuming). However, the number cannot be small either; otherwise, the pairs of pairs that we select would not be representative. We thus chose 80 (8% of the 1000) which is a trade-off between efficiency and accuracy as the number of the most inconsistent pairs of pairs.

E.3 Results

$T_n = 30$, $T_{POP\&C} = 0.3$ and $T_{POP\&ISO} = 0.529$ were used to select 80 out of 1000 pairs of pairs according to the criteria introduced above. It is observed that the value of $T_{POP\&ISO}$ is larger than $T_{POP\&C}$. There exist two possible explanations: (1) the disagreements: $GAP^i_{POP\&C}$ (see Equation (E.1)) and $GAP^i_{POP\&ISO}$ (see Equation (E.2)) were not normalised, thus both disgreements might lie in different scale ranges; and (2) the $J_E(i)$ (see Equation (4.12)) and $J_{ISO}(i)$ (see Equation (4.3)) used for computing the two disagreements were obtained in different ways. Consequently, $GAP^i_{POP\&C}$ and $GAP^i_{POP\&ISO}$ probably lie in different scale ranges. This may result in the fact that $T_{POP\&ISO}$ is larger than $T_{POP\&C}$. Figures E.2-E.5 display the 80 most inconsistent pairs of pairs in the descending sequence of the average $GAP_{POP\&C}$. Obviously, the majority of the 51 computational feature sets "made" completely different choices from the majority of the human observers in the pair-of-pairs and free-grouping on these pairs of pairs.

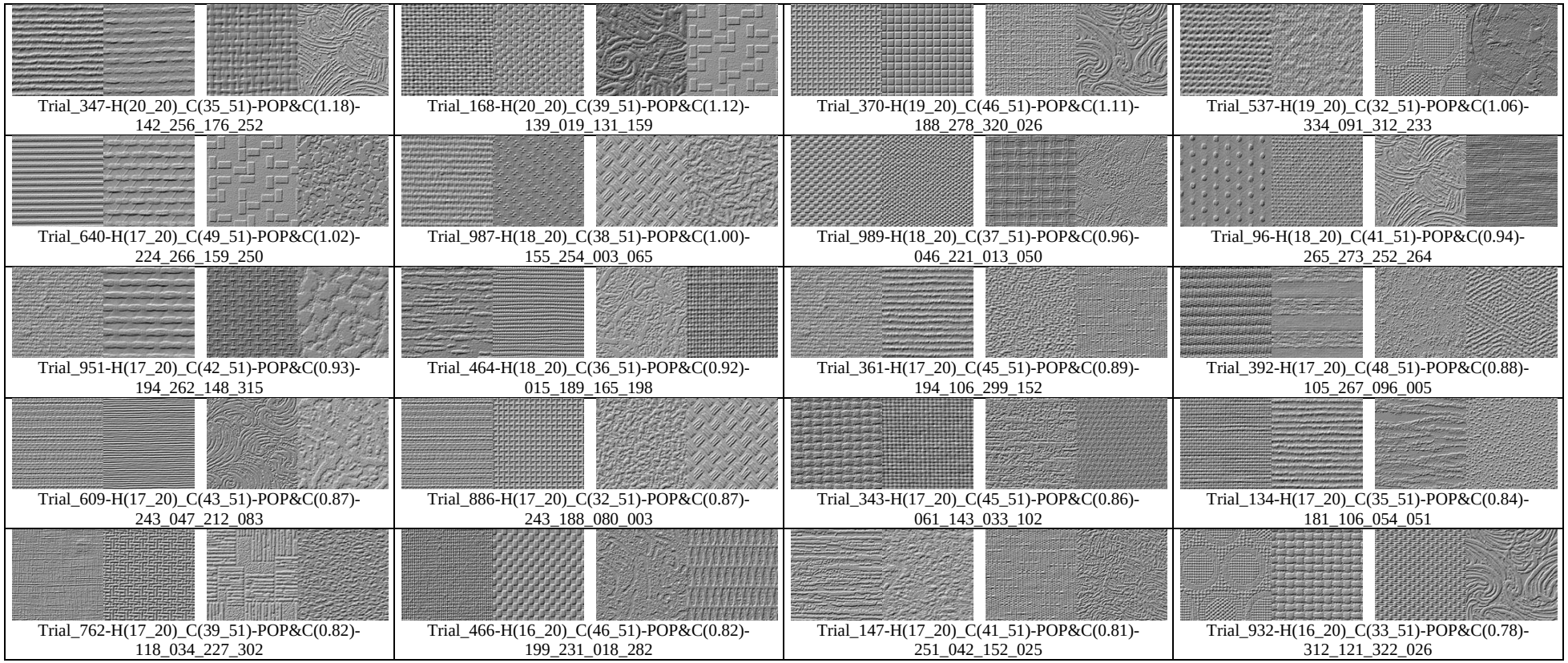


Figure E.2: The details of the first 20 most inconsistent pairs of pairs in the descending sequence (raster-scan order in the figure) of the average $GAP_{POP\&C}$ (i.e. "POP&C(c)", also see Section E.2) between the majority of the observers in the pair-of-pairs and the majority of the 51 feature sets. Here, "Trial_i" means the i-th trial; "H(a_20)"stands for that "a" out of all 20 observers chose the left pair as the more similar one; "C(b_51)" means that "b" out of the 51 feature sets "judged" that the right pair is more similar; and "d_e_f_g" denotes the names of four textures in turn. Note that the left and right pairs in some pairs of pairs have been swapped for display purposes and only the central quarter of each texture is displayed.

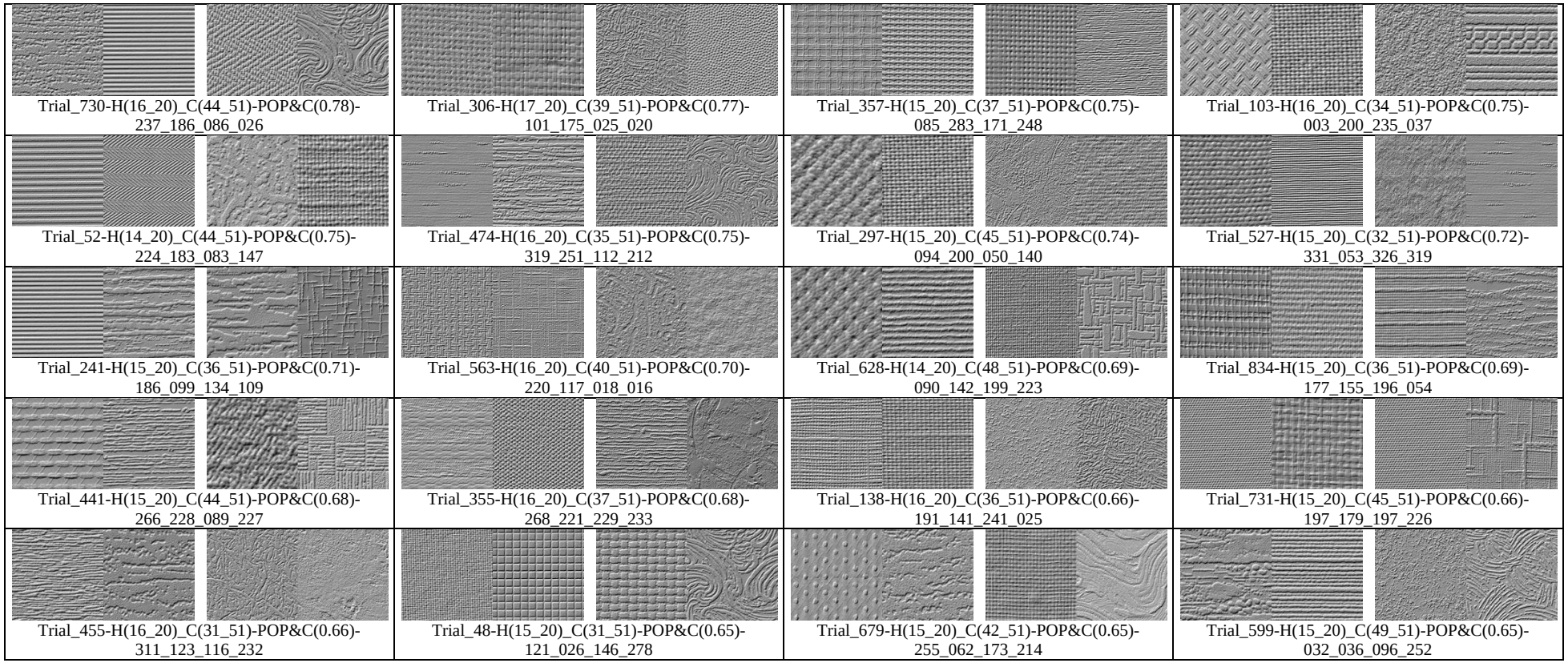


Figure E.3: The details of the second 20 most inconsistent pairs of pairs in the descending sequence (raster-scan order in the figure) of the average $GAP_{POP\&C}$ (i.e. "POP&C(c)", also see Section E.2) between the majority of the observers in the pair-of-pairs and the majority of the 51 feature sets. Here, "Trial_i" means the i-th trial; "H(a_20)"stands for that "a" out of all 20 observers chose the left pair as the more similar one; "C(b_51)" means that "b" out of the 51 feature sets "judged" that the right pair is more similar; and "d_e_f_g" denotes the names of four textures in turn. Note that the left and right pairs in some pairs of pairs have been swapped for display purposes and only the central quarter of each texture is displayed.

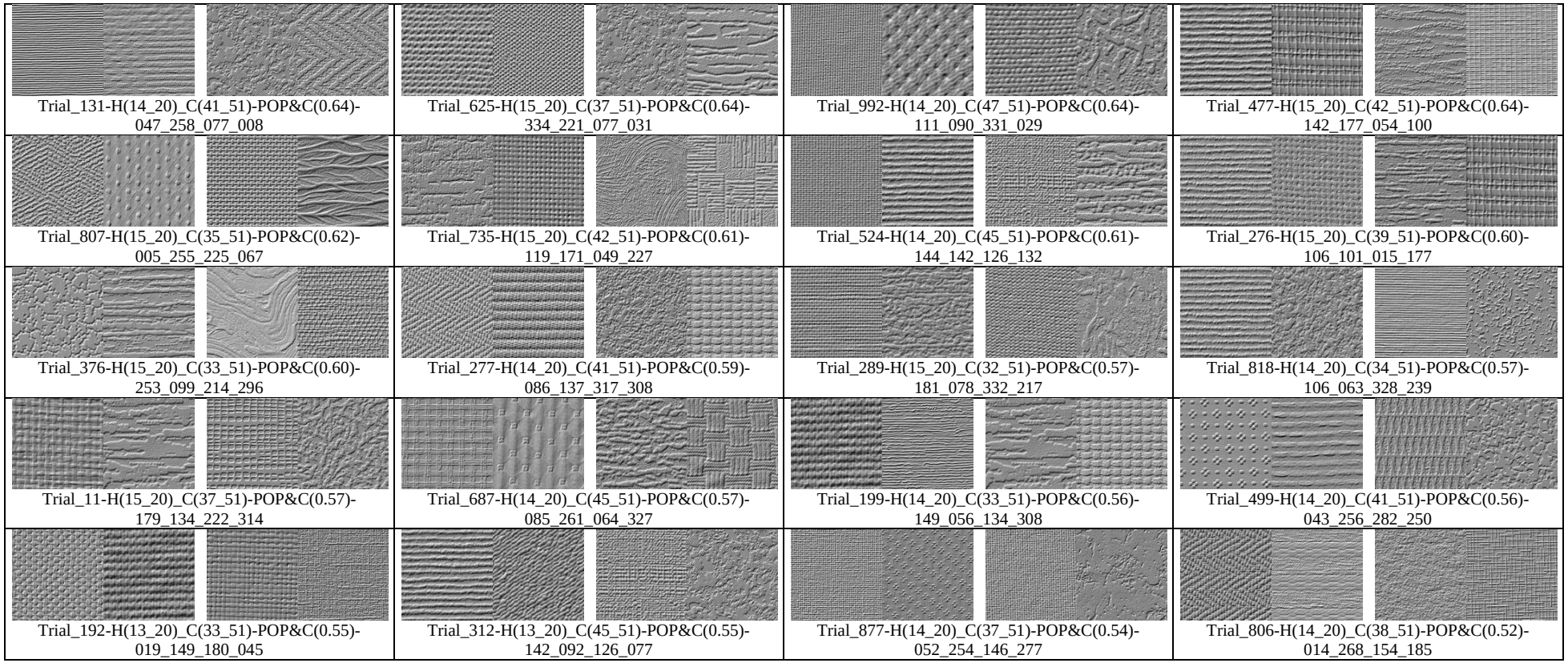


Figure E.4: The details of the third 20 most inconsistent pairs of pairs in the descending sequence (raster-scan order in the figure) of the average $GAP_{POP\&C}$ (i.e. "POP&C(c)", also see Section E.2) between the majority of the observers in the pair-of-pairs and the majority of the 51 feature sets. Here, "Trial_i" means the i-th trial; "H(a_20)"stands for that "a" out of all 20 observers chose the left pair as the more similar one; "C(b_51)" means that "b" out of the 51 feature sets "judged" that the right pair is more similar; and "d_e_f_g" denotes the names of four textures in turn. Note that the left and right pairs in some pairs of pairs have been swapped for display purposes and only the central quarter of each texture is displayed.

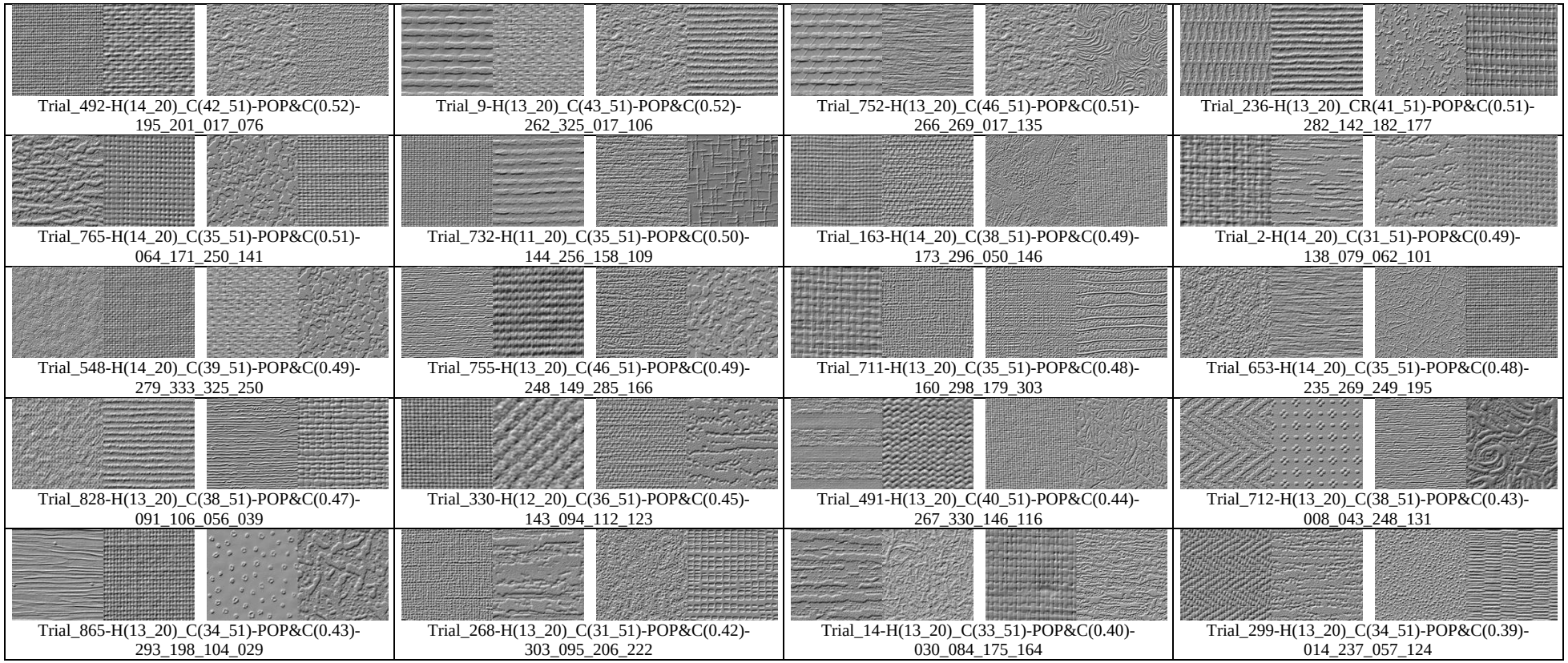


Figure E.5: The details of the fourth 20 most inconsistent pairs of pairs in the descending sequence (raster-scan order in the figure) of the average $GAP_{POP\&C}$ (i.e. "POP&C(c)", also see Section E.2) between the majority of the observers in the pair-of-pairs and the majority of the 51 feature sets. Here, "Trial_i" means the i-th trial; "H(a_20)"stands for that "a" out of all 20 observers chose the left pair as the more similar one; "C(b_51)" means that "b" out of the 51 feature sets "judged" that the right pair is more similar; and "d_e_f_g" denotes the names of four textures in turn. Note that the left and right pairs in some pairs of pairs have been swapped for display purposes and only the central quarter of each texture is displayed.

Publications by the Candidate

Clarke, A. D. F., Dong, X. and Chantler, M. (2011). Can perceptual visual texture similarity be modelled using metrics derived from free grouping experiments? Applied Vision Association (AVA), Christmas Meeting.

Clarke, A. D. F., Dong, X. and Chantler, M. J. (2012). Does Free-sorting Provide a Good Estimate of Visual Similarity. *Predicting Perceptions: The 3rd International Conference on Appearance*, pp. 17-20.

Dong, X. and Chantler, M. J. (2013). The Importance of Long-Range Interactions to Texture Similarity. *Computer Analysis of Images and Patterns, Lecture Notes in Computer Science (CAIP 2013)*, Vol. 8047, pp. 425-432.

Dong, X., Methven, T. and Chantler, M. J. (2014). How Well Do Computational Features Perceptually Rank Textures? A Comparative Evaluation. *Proceedings of the ACM 2014 International Conference on Multimedia Retrieval (ICMR 2014)*, pp. 281-288.

Dong, X. and Chantler, M. J. (2014). Texture Similarity Estimation Using Contours. *Proceedings of the 2014 British Machine Vision Conference (BMVC 2014)*.

Bibliography

Abbadeni, N. (2011). Computational Perceptual Features for Texture Representation and Retrieval. *IEEE Transactions on Image Processing*, Vol. 20(1), pp. 236-246.

Abbasi, S., Mokhtarian, F. and Kittler, J. (1999). Curvature scale space image in shape similarity retrieval. *Multimedia Systems*, Vol. 7, pp. 467-476.

Abdi, H. (2007). Bonferroni and Šidák corrections for multiple comparisons. *Encyclopedia of Measurement and Statistics*. Salkind, N. J. (Ed.). Thousand Oaks, CA: Sage.

Ade, F. (1983). Characterisation of Texture by 'Eigenfilter'. *Signal Processing*, Vol. 5, pp. 451-457.

Agresti, A. (2002). Categorical Data Analysis, 2nd ed., New York: Wiley.

Ahonen, T., Matas, J., He, C., and Pietikainen, M. (2009). Rotation Invariant Image Description with Local Binary Pattern Histogram Fourier Features. In *Proceedings of SCIA 2009*, pp. 61-70.

Ahonen, T. and Pietikäinen, M. (2009). Image description using joint distribution of filter bank responses. *Pattern Recognition Letters*, Vol. 30(4), pp. 368-376.

Amadasun, M. and King, R. (1989). Textural features corresponding to textural properties. *IEEE Transactions on System, Man and Cybernetics*, Vol. 19 (5), pp. 1264-1274.

Arbelaez, P., Maire, M., Fowlkes, C. and Malik, J. (2011). Contour Detection and Hierarchical Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 33(5), pp. 898-916.

Arbter, K., Snyder, W. E., Burkhardt, H. and Hirzinger, G. (1990). Application of affine-invariant Fourier descriptors to recognition of 3-D objects. *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 12(7), pp. 640-647.

Arlinghaus, S. L. (1994). *PHB Practical Handbook of Curve Fitting*. CRC Press.

Asada, H., and Brandy, M. (1986). The curvature primal sketch. *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 8(1), pp. 2-14.

Bai, X., Liu, W. and Tu, Z. (2009). Integrating Contour and Skeleton for Shape Classification. In *Proceedings of International Conference on Computer Vision Workshops (ICCV Workshops)*, pp. 360-367.

Bai, X., Yang, X. and Latecki, L. (2008). Detection and Recognition of Contour Parts Based on Shape Similarity, Pattern Recognition, Vol. 41, pp. 2189-2199.

Bar-Ilan, J., Mat-Hassan, M. and Levene, M. (2006). Methods for Comparing Rankings of Search Engine Results. *Computer Networks*, Vol. 50(10), pp. 1448-1463.

Bar-Ilan, J., Keenoy, K., Yaari, E. and Levene, M. (2007). User Rankings of Search Engine Results. *Journal of the American Society for Information Science and Technology*, Vol. 58(9), pp. 1254-1266.

Belongie, S., Malik, J. and Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24(4), pp. 509-522.

Bennett, J. and Khotanzad, A. (1998). Multispectral random field models for synthesis and analysis of colour images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20(3), pp. 327-332.

Berretti, S., Bimbo, A.D. and Pala, P. (2000). Retrieval by shape similarity with perceptual distance and effective indexing. *IEEE Trans. Multimedia*, Vol. 2 (4), pp. 225-239.

Bogacz, R, Brown, E, Moehlis, J, Holmes, P, and Cohen, JD. (2006). The Physics of Optimal Decision Making: A Formal Analysis of Models of Performance in Two-Alternative Forced-Choice Tasks. *Psychological Review*, Vol. 113 (4), pp. 700-765.

Bovik, A.C., Clark, M., and Geisler, W.S. (1990). Multichannel Texture Analysis Using Localised Spatial Filters. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 12, pp. 55-73.

Braun, J. (1999). On the detection of salient contours. *Spatial Vision*, Vol. 12, pp. 211-225.

Brodatz, P. (1966). *Textures: A Photographic Album for Artists and Designers*. Dover Publications.

Brincat, S. L. and Westheimer, G. (2000). Integration of foveal orientation signals: distinct local and long-range spatial domains. *Journal of Neurophysiology*, Vol. 83, pp. 1900-1911.

Burt, P.J. (1981). Fast filter transforms for image processing. *Computer Graphics and Image Processing*, Vol. 16(1), pp. 20-51.

Canny, J. (1986). A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8(6), pp. 679-698.

Carlucci, L. (1972). A formal system for texture languages. *Pattern Recognition*, Vol. 4(1), pp. 53-72.

Cawley, G. C. and Talbot, N. L. C. (2010). Over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, Vol. 11, pp. 2079-2107.

Chang, K. I., Bowyer, K. W. and Sivagurunath, M. (1999). Evaluation of Texture Segmentation Algorithms. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 294-299.

Charrier, C., Maloney, L. T., Cherifi, H. and Knoblauch, K. (2007). Maximum likelihood difference scaling of image quality in compression-degraded images. J. Opt. Soc. Am. A, Vol. 24(11), pp. 3418-3426.

Chaudhuri, B.B., Sarkar, N. and Kundu, P. (1993). Improved fractal geometry based texture segmentation technique. *IEEE Proceedings of Computers and Digital Techniques*, Vol. 140(5), pp. 233-241.

Chellappa, R. and Chatterjee, S. (1985). Classification of Textures Using Gaussian Markov Random Fields, *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. 33(4), pp. 959-963.

Chen, J. and Kundu, A. (1994). Rotation and Greyscale Transform Invariant Texture Identification using Wavelet Decomposition and HMM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, pp. 208-214.

Cho, R. Y., Yang, V. and Hallett, P. E. (2000). Reliability and dimensionality of judgments of visually textured materials. *Perception & Psychophysics*, Vol. 62(4), pp. 735-752.

Chomsky, N. (1957). *Syntactic Structures*. The Hague/Paris: Mouton.

Chubb, C. and Yellott, J. I. (2000). Every discrete, finite image is uniquely determined by its dipole histogram. *Vision Research*, Vol. 40(5), pp. 485-492.

Clarke, A. D. F., Halley, F., Newell, A. J., Griffin, L. D. and Chantler, M. J. (2011). Perceptual Similarity: A Texture Challenge. In *Proceedings of British Machine Vision Conference*, pp. 120.1-120.10.

Clarke, A.D.F., Dong, X. and Chantler, M. J. (2012). Does Free-sorting Provide a Good Estimate of Visual Similarity. In *Predicting Perceptions: Proceedings of the 3rd International Conference on Appearance*, pp. 17-20.

Coggins, J.M. and Jain, A.K. (1985). A Spatial Filtering Approach to Texture Analysis. *Pattern Recognition Letters*, Vol. 3, pp. 195-203.

Conners, R. W. and Harlow, C. A. (1980). A Theoretical Comparison of Texture Algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 2(3), pp. 204-222.

Contour, available:
http://en.wikipedia.org/wiki/Contour [accessed 8 Sep. 2014].

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, Vol. 20(3), pp. 273-297.

Csurka, G., Dance, C., Fan, L. X., Willamowski, J. and Bray, C. (2004). Visual categorisation with bags of keypoints. *Proceedings of ECCV International Workshop on Statistical Learning in Computer Vision*.

Cula, O. G. and Dana, K. J. (2004). 3D Texture Recognition Using Bidirectional Feature Histograms. *International Journal of Computer Vision*, Vol. 59, pp. 33-60

CVAP. (2004). KTH-TIPS, available:
http://www.nada.kth.se/cvap/databases/kth-tips/ [accessed 11 Nov. 2013].

CVAP. (2005). KTH-TIPS2, available:

http://www.nada.kth.se/cvap/databases/kth-tips/ [accessed 11 Nov. 2013].

Dakin, S. C. and Watt, R. J. (1997). The Computation of Orientation Statistics from Visual Texture. *Vision Research*, Vol. 37(22), pp. 3181-3192.

Dakin, S. C. (1999). Orientation variance as a quantifier of structure in texture. *Spatial Vision*, Vol. 12, pp. 1-30.

Dakin, S. C. and Hess, R. F. (1999). Contour integration and scale combination processes in visual edge detection. *Spatial Vision*, Vol. 12(3), pp. 309-327.

Dana, K.J., van Ginneken, B., Nayar, S.K. and Koenderink, J.J. (1999). Reflectance and Texture of Real World Surfaces. *ACM Transactions on Graphics*, Vol. 18(1), pp. 1-34.

Das, M., Paulik, M. J. and Loh, N. K. (1990). A bivariate autoregressive modeling technique for analysis and classification of planar shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 12(1), pp. 97-103.

David, H. A. (1988). *The Method of Paired Comparisons*. New York: Oxford University Press.

Davies, E.R. (1997). *Machine Vision: Theory, Algorithms, Practicalities*. New York: Academic Press, pp. 171-191.

De Winter, J. and Wagemans, J. (2004). Contour-based object identification and segmentation: Stimuli, norms and data, and software tools. *Behavior Research Methods, Instruments, & Computers*, Vol. 36, pp. 604-624.

De Winter, J. and Wagemans, J. (2008A). Perceptual saliency of points along the contour of everyday objects: A large-scale study. *Perception & Psychophysics*, Vol. 70, pp. 50-64.

De Winter, J. and Wagemans, J. (2008B). The awakening of Attneave's sleeping cat: Identification of everyday objects on the basis of straight-line versions of outlines. *Perception*, Vol. 37, pp. 245-270.

Del Bimbo, A. and Pala, P. (1997). Visual image retrieval by elastic matching of user sketches. *IEEE Trans. Pattern Anal. Mach. Intell*, Vol. 19(2), pp. 121-132.

Deza, E. and Deza, M.M. (2009). *Encyclopedia of Distances*, Springer.

Diaconis, P. and Graham, R. L. (1977). Spearman's footrule as a measure of disarray. *Journal of the Royal Statistical Society, Series B (Methodological)*, Vol. 39, pp. 262-268.

Dillencourt, M. B., Samet, H. and Tamminen, M. (1992). A general approach to connected-component labeling for arbitrary image representations. *Journal of the ACM*, Vol. 39(2), pp. 253-280.

Do, M. N. and Vetterli, M. (2002). Wavelet-Based Texture Retrieval Using Generalised Gaussian Density and Kullback–Leibler Distance. *IEEE TRANSACTIONS ON IMAGE PROCESSING*, Vol. 11(2), pp. 146-158.

Dubois, S. R. and Glanz, F. H. (1986). An model approach to two-dimensional shape classification. *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 8, pp. 627-637.

Dudek, G. and Tsotsos, J.K. (1997). Shape representation and recognition from multiscale curvature. *Comput. Vision Image Understanding*, Vol. 68 (2), pp. 170-189.

Efros, A. A. and Freeman, W. T. (2001). Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 341-346.

Elfadel, I. M. and R. W. Picard. (1994). Gibbs random fields, cooccurrences, and texture modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16(1), pp. 24-37.

Emrith, K. (2008). *Perceptual Dimensions for Surface texture Retrieval*, Ph.D. thesis. Heriot Watt University, Edinburgh.

Emrith, K., Chantler, M. J., Green, P. R., Maloney, L. T. and Clarke, A. D. F. (2010). Measuring Perceived Differences in Surface Texture due to Changes in Higher Order Statistics. *Journal of Optical Society of America*, Vol. 27(5), pp. 1232-1244.

Fagin, R., Kumar, R. and Sivakumar, D. (2003). Comparing Top *K* Lists. In *ACM-SIAM Symposium on Discrete Algorithms - SODA*, Vol. 17(1), pp. 28-36.

Field, D. J., Hayes, A. and Hess, R. F. (1993). Contour integration by the human visual system: evidence for a local "association field". *Vision Research*. Vol. 33, pp. 173-193.

Field, A. (2009). *Discovering Statistics Using SPSS*. SAGE Publications Ltd.

Fogel, I. and Sagi, D. (1989). Gabor filters as texture discriminator. *Biological Cybernetics*, Vol. 61, pp. 103-113.

Fraley, C. and Raferty, A. E. (1998). How many clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis?. *The Computer Journal*, Vol. 41(8), pp. 578-588.

Freeman, H. (1961). On the encoding of arbitrary geometric configurations. *IRE Trans. Electron. Comput*, EC-10, pp. 260-268.

Freeman, H. (1977). Shape Description via the Use of Critical Points. In: *Proceedings of Pattern Recognition of Image Processing*, pp. 168-174.

Freeman, H. and Saghri, A. (1978). Generalised chain codes for planar curves. In: *Proceedings of the Fourth International Joint Conference on Pattern Recognition*, pp. 701-703.

Fujii, K., Sugi, S., and Ando, Y. (2003). Textural properties corresponding to visual perception based on the correlation mechanism in the visual system, *Psychological Research*, Vol. 67(3), pp. 197-208.

Galloway, M. M. (1975). Texture Classification Using Grey Level Run Lengths. *Computer Graphics and Image Processing*, Vol. 4, pp. 172-179.

Gimel'farb, G.L. and Zalesny, A.V. (1993). Markov Random Fields with Short and Long-Range Interaction for Modelling Grey-Scale Texture Images. in *Proceeding of 5th International Conference on Computer Analysis of Images and Patterns*, pp. 275-282.

Glass, J. M. (1966). Smooth-curve interpolation: A generalised spline-fit procedure. *BIT Numerical Mathematics*, Vol. 6(4), pp 277-293.

Gonzalez, R.C. and Woods, R.E. (2002). *Digital Image processing*, Prentice Hall Upper Saddle River. NJ.

Greenhouse, S. W. and Geisser, S. (1959). On methods in the analysis of profile data. *Psychometrika*, 24, 95-112.

Groskey, W.I. and Mehrotra, R. (1990). Index-based object recognition in pictorial data management. *Comput. Vision Graphics Image Process*, Vol. 52, pp. 416-436.

Groskey, W.I., Neo, P. and Mehrotra, R. (1992). A pictorial index mechanism for model-based matching. *Data Knowledge Eng.*, Vol. 8, pp. 309-327.

Guo, C., Zhu, S. and Wu, Y. (2007). Primal sketch: Integrating structure and texture. *Computer Vision and Image Understanding*, Vol. 106(1), pp. 5-19.

Halley, F. (2011A). *Perceptually Relevant Browsing Environments for Large Texture Databases*. PhD thesis. Heriot Watt University, Edinburgh.

Halley, F. (2011B). Pertex v1.0, available: http://www.macs.hw.ac.uk/texturelab/resources/databases/pertex/ [accessed 13 June 2013].

Hansen, B. C. and Hess, R. F. (2006). The role of spatial phase in texture segmentation and contour integration. *Journal of Vision*, Vol. 6, pp. 594-615.

Hansen, B. C. and Hess, R. F. (2007). Structural sparseness and spatial phase alignment in natural scenes. *J. Opt. Soc. Am. A. Opt. Image Sci. Vis.*, Vol. 24(7), pp. 1873-1885.

Haralick, R., Shanmugam, K. and Dinstein, I. (1973). Textural Features for Image Classification. *IEEE Trans. Systems, Man, Cybernetics*, Vol. 3, pp. 610-621.

Haralick, R. M. (1979). Statistical and Structural Approaches to Texture. *Proceedings of the IEEE*, Vol. 67(5), pp. 786-804.

Hariri, N. (2011). Relevance Ranking on Google: Are Top Ranked Results Really Considered More Relevant by the Users?. *Online Information Review*, Vol. 35(4), pp. 598-610.

Harwood, D., Ojala, T., Pietiktiinen, M., Kelman, S., and Davis, L. S. (1995). Texture classification by centre-symmetric auto-correlation, using Kullback discrimination of distributions. *Pattern Recognition Letter*, Vol. 16(1), pp. 1-10.

He, J., Li, M., Zhang, H., Tong, H. and Zhang, C. (2004). Manifold-Ranking Based Image Retrieval. In: *Proceedings of ACM Int'l Conf. Multimedia*, pp. 9-16.

Heaps, C. and Handel, S. (1999). Similarity and features of natural textures. *Journal of Experimental Psycholog.: Human Perception and Performance*, Vol. 25, pp. 299-320.

Hopfield, J.J. (1988). Artificial neural networks, *IEEE Circuits and Devices Magazine*, Vol. 4(5), pp. 3-10.

Huang, C. L. and Huang, D. H. (1998). A content-based image retrieval system, *Image Vision Comput*, Vol. 16, pp. 149-163.

Iivarinen, J. and Visa, A. (1996). Shape recognition of irregular objects. In D.P. Casasent (Ed.), *Intelligent Robots and Computer Vision XV: Algorithms, Techniques, Active Vision, and Materials Handling*, Proc. SPIE 2904, pp. 25-32.

Jafari-Khouzani, K. and Soltanian-Zadeh, H. (2005). Radon transform orientation estimation for rotation invariant texture analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27(6), pp. 1004-1008.

Jain, A.K. and Farrokhnia, F. (1991). Unsupervised Texture Segmentation Using Gabor Filters. *Pattern Recognition*, Vol. 24, pp. 1167-1186.

Jain, A. K. and Karu, K. (1996). Learning texture discrimination masks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18, pp. 195-205.

Jain, R., Kasturi, R. and Schunck, B. G. (1995). *Machine Vision*. McGraw-Hill.

JavaScript, available:
http://en.wikipedia.org/wiki/JavaScript [accessed 11 Dec. 2013].

Joanl, F. B. (1987). Guinness, Gosset, Fisher, and Small Samples. *Statistical Science*. Institute of Mathematical Statistics, Vol. 2(1), pp. 45-52.

Jojic, N., Frey, B. J. and Kannan, A. (2003). Epitomic analysis of appearance and shape. In *Proceedings of IEEE International Conference on Computer Vision*, Vol. 1, pp. 34-43.

Julesz, B. (1981) Textons, the Elements of Texture Perception, and their Interactions. *Nature*, Vol. 290(5802), pp. 91-97.

Kadyrov, A. and Petrou, A. (2001). The Trace Transform and Its Applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23(8), pp. 811-828.

Kadyrov, A., Talepbour, A. and Petrou, M. (2002). Texture classification with thousands of features. In *Proceedings of BMVC 2002*, pp. 656-665.

Kendall, M. and Gibbons, J. D. (1990). *Rank Correlafion Memods*. 5th Ed., London.

Khelifi, F. and Jiang, J. (2011). k-NN Regression to Improve Statistical Feature Extraction for Texture Retrieval. *IEEE TRANSACTIONS ON IMAGE PROCESSING*, Vol. 20(1), pp. 293-298.

Kim, J. K. and Park, H. W. (1999). Statistical textural features for detection of microcalcifications in digitised mammograms. *IEEE Transactions on Medical Imaging*, Vol. 18(3), pp. 231-238.

Koenderink, J. J. (1984). The structure of images. *Biological Cybernetics*, Vol. 50(5), pp. 363-370.

Koenderink, J. J. and Van Doorn, A. J. (1999). The Structure of Locally Orderless Images. *International Journal of Computer Vision*, Vol. 31(2-3), pp. 159-168.

Kolmogorov, A.N. (1933). Sulla determinazione empirica di una legge di distribuzione. *Giornale dell'Istituto Italiano degli Attuari*, Vol. 4, pp. 83-91.

Kovács, I. and Julesz, B. (1993). A closed curve is much more than an incomplete one: effect of closure in figure-ground segmentation. *Proc. Natl. Acad. Sci. USA*, Vol. 90(16), pp. 7495-7497.

Kovács, I., Polat, U., Pennefather, PM., Chandna, A. and Norcia, AM. (2000). A new test of contour integration deficits in patients with a history of disrupted binocular experience during visual development. Vision Res, Vol. 40(13), pp. 775-83.

Kovesi, P. (2000). Phase congruency: a low-level image invariant. *Psychol. Res*, Vol. 64, pp. 136-148.

Kovesi, P. (2003). Phase congruency detects corners and edges. In: *Proceedings of The Australian Pattern Recognition Society Conference*.

Kwitt, R. (2009). MRSAR, available:
http://www.wavelab.at/sources/MRSAR/ [accessed 16 September 2013].

Kwitt, R. and Meerwald, P. STex, available:
http://www.wavelab.at/sources/STex/ [accessed 11 Nov. 2013].

Laws, K.I. (1980). Rapid Texture Identification. In *Proc. SPIE Conf. Image Processing for Missile Guidance*, pp. 376-380.

Lazebnik, L. (2003). Ponce Texture Database, available:

http://www-cvr.ai.uiuc.edu/ponce_grp/data/texture_database/samples/ [accessed 11 Nov. 2013].

Lazebnik, S., Schmid, C. and Ponce, J. (2005). UIUCTex, available: http://staff.neu.edu.tr/~kkilic/prj/lac/uiuc/uiuc.html [accessed 11 Nov. 2013].

Lazebnik, S., Schmid, C. and Ponce, J. (2006). Beyond Bags of Features: Spatial Pyramid Matching for Recognising Natural Scene Categories. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2, pp. 2169-2178.

Leung, T. and Malik, J. (2001). Representing and Recognising the Visual Appearance of Materials using Three-dimensional Textons. *International Journal of Computer Vision*, Vol. 43, pp. 29-44.

Li, S.Z. (1999). Shape matching based on invariants. In O. Omidvar (Ed.), *Shape Recognition, Progress in Neural Networks*, Vol. 6, pp. 203-228.

Li, Y., Zhang, J. and Jiang, P. (2010). Contour Extraction Based on Surround Inhibition and Contour Grouping. In: *Proceedings of ACCV 2009*, Part II, pp. 687–696.

Liang, L., Liu, C., Xu Y., Guo, B., and Shum, H. (2001). Real-time texture synthesis by patch-based sampling. *ACM Transactions on Graphics*, Vol. 20(3), pp. 127-150.

Liu, Z. and Madiraju, S. V. R. (1996). Covariance-based approach to texture processing, *Applied Optics*, Vol. 35(5), pp. 848-853.

Liu, F. and Picard, R.W. (1998). Finding Periodicity in Space and Time. In: Proceedings of International Conference on Computer Vision, pp. 376-383.

Lizorkin, P.I. (2001). Fourier Transform. *Encyclopedia of Mathematics*, available: http://www.encyclopediaofmath.org/index.php?title=Fourier_transform&oldid=12659 [accessed 13 Dec. 2013].

Long, H., Leow, W. K. and Chua, F. K. (2000). Perceptual Texture Space for Content-Based Image Retrieval. In: Proceedings of Int. Conf. on Multimedia Modelling, pp. 167-180.

Long, H., Tan, C. W., and Leow, W. K. (2001). Invariant and perceptually consistent texture mapping for content-based image retrieval. In: *Proceedings of International Conference on Image Processing*, Vol. 2, pp. 117-120.

Long, H. and Leow, W. K. (2002A). A Hybrid Model for Invariant and Perceptual Texture Mapping. In: *Proceedings of 16th International Conference on Pattern Recognition*, Vol. 1, pp. 135-138.

Long, H. and Leow, W. K. (2002B). Perceptual Consistency Improves Image Retrieval Performance. In: *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 434-435.

Long, H. and Leow, W. K. (2002C). Perceptual Texture Space Improves Perceptual Consistency of Computational Features. In: *Proceedings of International Joint Conference on Artificial Intelligence*, pp. 1391-1396.

Lowe, D. G. (1985). *Perceptual Organisation and Visual Recognition*, Kluwer Academic publishers.

Malik, J., Belongie, S., Leung, T. and Shi, J. (2001). Contour and Texture Analysis for Image Segmentation. *International Journal of Computer Vision*, Vol. 43(1), pp. 7-27.

Malik, J. and Perona, P. (1990). Preattentive texture discrimination with early vision mechanisms. *Journal of the Optical Society of America A*, Vol. 7, pp. 923-932.

Mandelbrot, B. B. (1982). *The Fractal Geometry of Nature*, W. H. Freeman & Co Ltd.

Manjunath, B.S. and Ma, W.Y. (1996). Texture features for browsing and retrieval of image data, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18, pp. 837-842.

Mann, H. B. and Whitney, D. R. (1947). On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other. *Annals of Mathematical Statistics*, Vol. 18(1), pp. 50-60.

Mao, J. and Jain, A.K. (1992). Texture classification and segmentation using multiresolution simultaneous autoregressive models. *Pattern Recognition*, Vol. 25(2), pp.173-188.

Marr, D. and Hildreth, E. (1980). Theory of Edge Detection. *Proceedings of the Royal Society of London*, Vol. 207, pp. 187-217.

Marr, D. (1982). *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. W.H. Freeman and Company.

Mauchly, J. W. (1940). Significance Test for Sphericity of a Normal *n*-Variate Distribution. *The Annals of Mathematical Statistics*, Vol. 11(2), pp. 204-209.

Mehrotra, R. and Gary, J.E. (1995). Similar-shape retrieval in shape data management. *IEEE Comput.*, Vol. 28 (9), pp. 57-62.

Metaxas, P. T., Ivanova, L. and Mustafaraj, E. (2009). New Quality Metrics for Web Search Results. In *Proceedings of 4th International Conference on Web Information Systems and Technologies*, Vol. 18, pp. 278-292.

Mirmehdi, M., Xie, X. and Suri, J. (2008). *Handbook of texture analysis*, 1st ed., London: Imperial College Press.

MIT. (1995). VisTex, available:
http://vismod.media.mit.edu/vismod/imagery/VisionTexture/ [accessed 11 Nov. 2013].

Mokhtarian, F., Abbasi, S. and Kittler, J. (1996). Robust and efficient shape indexing through curvature scale space. In: *Proceedings of the British Machine Vision Conference*, pp. 53-62.

Natural Colour System, available:
http://en.wikipedia.org/wiki/Natural_Colour_System [accessed 16 Dec. 2013].

Nealen, A. and Alexa, M. (2003). Hybrid Texture Synthesis. In *Proceedings of the 14th Eurographics workshop on Rendering*, pp. 97-105.

Ng, I., Tan, T., and Kittler, J. (1992). On Local Linear Transform and Gabor Filter Representation of Texture. In *Proceedings of International Conference on Pattern Recognition*, pp. 627-631.

Nurminen, L., Peromaa, T. and Laurinen, P. (2010). Surround suppression and facilitation in the fovea: Very long-range spatial interactions in contrast perception. *Journal of Vision*, Vol. 10, pp. 1-13.

Ojala, T., Mäenpää, T., Pietikäinen, M., Viertola, J., Kyllönen, J. and Huovinen, S. (2002a). Outex-New framework for empirical evaluation of texture analysis algorithms. In *Proceedings of 16th International Conference on Pattern Recognition*, Vol. 1, pp. 701-706.

Ojala, T., Pietikäinen, M., and Harwood, D. (1996). A Comparative Study of Texture Measures with Classification Based on Feature Distributions. *Pattern Recognition*, Vol. 29, pp. 51-59.

Ojala, T., Pietikainen, M., and Maenpaa, T. (2002b). Multiresolution Grey-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, pp. 971-987.

Ojansivu, V. and Heikkila, J. (2008). Blur insensitive texture classification using local phase quantisation. In: *Proceedings of International Conference on Image and Signal*, pp. 236-243.

Ojansivu, V., Rahtu, E. and Heikkila, J. (2008). Rotation invariant local phase quantisation for blur insensitive texture analysis. In: *Proceedings of International Conference on Pattern Recognition 2008*, pp. 1-4.

Oppenheim, A. V. and Lim, J. S. (1991). The Importance of Phase in Signals. *Proceedings of the IEEE*, Vol. 69 (5), pp. 529-541.

Padilla, S. (2008). *Mathematical models for perceived roughness of three-dimensional surface textures*. PhD thesis, Heriot-Watt University.

Panis, S., De Winter, Joachim Vandekerckhove, J., and Wagemans, J. (2008). Identification of everyday objects on the basis of fragmented outline versions. *Perception*, Vol. 37, pp. 271-289.

Papari, G. and Petkov, N. (2011). Edge and line oriented contour detection: State of the art. *Image and Vision Computing*, Vol. 29, pp. 79-103.

Payne, J. S., Hepplewhite, L. and Stonham, T. J. (1999). Perceptually Based Metrics for the Evaluation of Textural Image Retrieval Methods. In *Proceedings of IEEE International Conference on Multimedia Computing and Systems*, Vol.2, pp. 793-797.

Pennefather, P. M., Chandna, A., Kovács, I., Polat, U. and Norcia, A. M. (1999). Contour detection threshold: repeatability and learning with “contour cards”. *Spat Vis*, Vol. 12, pp. 257-266.

Pentland, A. P. (1984). Fractal-Based Description of Natural Scenes, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 6, pp. 661-674.

Persoon, E. and Fu, K. (1977). Shape Discrimination Using Fourier Descriptors. *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 7(3), pp. 170-179.

Pettet, M. W., McKee, S. P. and Grzywacz, N. M. (1998). Constraints on long range interactions mediating contour detection. *Vision Research*, Vol. 38, pp. 865-879.

Peura, M. and Iivarinen, J. (1997). Efficiency of simple shape descriptors. In: Proceedings of the Third International Workshop on Visual Form, pp. 443-451.

PHP, available:
http://en.wikipedia.org/wiki/PHP [accessed 11 Dec. 2013].

Picard, R.W., Kabir, T. and Liu, F. (1993). Real-time recognition with the entire Brodatz texture database. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 638-639.

Polat, U. and Sagi, D. (1994). The Architecture of Perceptual Spatial Interactions. *Vision Research*, Vol. 34, pp. 73-78.

Polat, U. (1999). Functional architecture of long-range perceptual interactions. *Spatial Vision*, Vol. 12, pp. 143-162.

Pont, S. C. and Koenderink, J. J. (2003). Illuminance Flow. In: *N. Petkov and m.A. Wetsenberg (eds): Computer Analysis of Images and Patterns, Springer-Verlag, Berlin Heidelberg, LNCS 2756*, pp. 90-97.

Portilla, J. and Simoncelli, E. P. (2000). A Parametric Texture Model Based on Joint Statistics of Complex Wavelet Coefficients. *International Journal of Computer Vision*, Vol. 40(1), pp. 49-71.

Press, W., Teukolsky, S., Vetterling, W. and Flannery, B. (1992). *Numerical Recipes in C*, 2nd ed., Cambridge University Press.

Prewitt, J. (1970). Object Enhancement and Extraction. In Lipkin, B. and Rosenfeld, A. (Ed.), *Picture Processing and Psychopictorics*, NY, Academic Press.

Qin, X. and Yang, Y. (2005). Basic Grey level aura matrices: theory and its application to texture synthesis. In: *Proceedings of the Tenth IEEE International Conference on Computer Vision*, Vol. 1, pp. 128-135.

Rahtu, E., Salo, M., and Heikkila, J. (2005). Affine invariant pattern recognition using multiscale autoconvolution, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, pp. 908-918.

Randen, T. and Husøy, J.H. (1999). Filtering for Texture Classification: A Comparative Study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 21, pp. 291-310.

Rao, A. R. and Lohse, G. L. (1993). Identifying high level features of texture perception. *Graphical Models and Image Processing*, Vol. 55 (3), pp. 218-233.

Rao, A. R. and Lohse, G. L. (1996). Towards a texture naming system: Identifying relevant dimensions of Texture. *Vision Research*, Vol. 36, pp. 1649-1669.

Reed, T. and Buf, J. (1993). A review of recent texture segmentation and feature extraction techniques. *Computer Vision, Image Processing and Graphics*, Vol. 57(3), pp. 359-372.

Roberts, L. G. (1965). Machine Perception of Three-Dimensional Solids. In Tippett, J. (Ed.), *Optical and Electro-Optical Information Processing*, MIT Press, pp. 159-197.

Rocchio, Jr. J.J. (1971). Performance Indices for Document Retrieval. In Salton, G. (Ed.), *The SMART Retrieval System-Experiments in Automatic Document Processing*. Englewood Cliffs, NJ: Prentice-Hall, pp. 57-67.

Rui, Y., Huang, T. S. and Mehrotra, S. (1998). Relevance Feedback Techniques in Interactive Content-Based Image Retrieval. In *Proceedings of the SPIE 3312, Storage and Retrieval for Image and Video Databases VI*, pp. 25-36.

Santini, S. and Jain, R. (1999). Similarity Measures. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, Vol. 21(9), pp. 871-883.

Sassi, M., Vancleef, K., Machilsen, B., Panis, S. and Wagemans, J. (2010). Identification of everyday objects on the basis of Gaborised outline versions. *i-Perception*, Vol. 1, pp. 121-142.

Schmid, C. (2001). Constructing models for content-based image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 39-45.

Sekita, I., Kurita, T. and Otsu, N. (1992). Complex autoregressive model for shape recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 14, pp. 489-496.

Shen, J. and Castan, S. (1992). An optimal linear operator for step edge detection. *CVGIP: Graphical Models and Image Processing*, Vol. 54, pp. 112-133.

Simoncelli, E. (2009). MatlabPyrTools-v1.4, available: http://www.cns.nyu.edu/~lcv/software.php [accessed 13 June 2013].

Sivic, J. and Zisserman, A. (2003). Video Google: A Text Retrieval Approach to Object Matching in Videos. *Proceedings of the Ninth IEEE International Conference on Computer Vision*, Vol. 2, pp. 1470-1477.

Smirnov, N. (1948). Tables for estimating the goodness of fit of empirical distributions. *Annals of Mathematical Statistics*, Vol. 19, pp. 279-281.

Smith, G. and Burns, I. (1997). Measuring texture classification algorithms. *Pattern Recognition Letters*, Vol. 18(14), pp. 1495-1501.

Smith, G., Burns, I. (1997). Meastex V1.1, available: http://www.texturesynthesis.com/meastex/meastex.html [accessed 11 Nov. 2013].

Sobel, I. (1990). An Isotropic 3×3 Gradient Operator. In Freeman, H. (Ed.), *Machine Vision for Three-Dimensional Scenes*, Academic Press, pp. 376-379.

Sokal, R. R. and Rohlf, F.J. (1969). *Biometry*. W.H. Freeman.

Sonka, M., Hlavac, V. and Boyle, R. (1993). *Image Processing, Analysis and Machine Vision*. London, UK, NJ: Chapman & Hall, pp. 193-242.

Speer, T., Kuppe, M. and Hoschek, J. (1998). Global reparametrisation for curve approximation. *Computer Aided Geometric Design*, Vol. 15(9), pp. 869-877.

Spillmann, L. and Werner, J. S. (1996). Long-Range Interactions in Visual Perception. *Trends in Neurosciences*, Vol. 19, pp. 428-434.

Squire, D.M. and Caelli, T.M. (2000). Invariance signature: characterising contours by their departures from invariance. *Comput. Vision Image Understanding*, Vol. 77, pp. 284-316.

Sun, K. and Super, B. (2005). Classification of Contour Shapes Using Class Segmentsets, In *Proceedings of IEEE Conference on Computer Visionand Pattern Recognition*, Vol. 2, pp.727-733.

Swain, M. J. and Ballard, D. H. (1991). Colour Indexing. *International Journal of Computer Vision*, Vol. 7(1), pp. 11-32.

Tadmor, Y. and Tolhurst, D. J. (1993). Both the phase and amplitude spectrum may determine the appearance of natural images. *Vision Res.*, Vol. 33, pp. 141-145.

Tamura, H., Mori, S. and Yamawaki, T. (1978). Textural Features Corresponding to Visual Perception. *IEEE Trans. Systems, Man, and Cybernetics*, Vol. 8, pp. 460-473.

Tenenbaum, J. B., Silva, V. de and Langford, J. C. (2000). A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, Vol. 290(5500), pp. 2319-2323.

Texture Lab, Heriot-Watt University. (2003). PhoTex, available: http://www.macs.hw.ac.uk/texturelab/resources/databases/photex/ [accessed 11 Nov. 2013].

Thacker, N. A., Aherne, F. J. and Rockett, P. I. (1997). The Bhattacharyya Metric as An Absolute Similarity Measure for Frequency Coded Data. *Kybernetika*, Vol. 34(4), pp. 363-368.

Todorovic, D. (2008). Gestalt principles, *Scholarpedia*, 3(12):5345, available: http://www.scholarpedia.org/article/Gestalt_principles [accessed 22 Dec 2013].

Toft, P. (1996). *The Radon Transform-Theory and Implementation*, Ph.D. thesis. Technical University of Denmark.

Tuceryan, M. and Jain, A.K. (1993). Texture Analysis. In *Handbook Pattern Recognition and Computer Vision*, pp. 235-276.

Tzvetanov, T. and Dresp, B. (2002). Short- and long-range effects in line contrast integration. *Vision Research*, Vol. 42, pp. 2493-2498.

Tzvetanov, T. and Simon, L. (2006). Short- and long-range spatial interactions: A redefinition. Vision Research, Vol. 46 pp. 1302-1306.

Unser, M. (1986). Sum and Difference Histograms for Texture Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8(1), pp. 118-125.

Van Gool, L., Dewaele, P. and Oosterlinck, A. (1985). Texture analysis. *Computer Vision, Graphics and Image Processing*, Vol. 29, pp. 336-357.

Varma, M. and Zisserman, A. (2005). A Statistical Approach to Texture Classification from Single Images. *International Journal of Computer Vision*, Vol. 62, pp. 61-81.

Varma, M. and Zisserman, A. (2009). A Statistical Approach to Material Classification Using Image Patch Exemplars. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 31, pp. 2032-2047.

Vilnrotter, F. M., Nevatia, R., and Price, K. E. (1986). Structural Analysis of Natural Textures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8, pp. 76-89.

Wagemans, J., De Winter, J., Op de Beeck, H., Ploeger, A., Beckers, T. and Vanroose, P. (2008). Identification of everyday objects on the basis of silhouette and outline versions. *Perception*, Vol. 37, pp. 207-244.

Wang, X., Albregtsen, F., and Foyn, B. (1994). Texture Features from Grey Level Gap Length Matrix, In *Proceedings of MVA94: IAPR Workshop on Machine Vision Applications*, pp. 375-378.

Wang, X., Feng, B., Bai, X., Liu, W. and Latecki, L. (2014). Bag of Contour Fragments for Robust Shape Classification. *Pattern Recognition*. Vol. 47, pp. 2116-2125.

Wang, L. and He, D. (1990). Texture classification using texture spectrum. *Pattern Recognition*, Vol. 23(8), pp. 905-910.

Weisstein, E. W.. Parseval's Theorem, available: http://mathworld.wolfram.com/ParsevalsTheorem.html [accessed 7 June 2013].

Wenger, R. (1997). Visual Art, Archaeology and Gestalt. *Leonardo*, Vol. 30, pp. 35-46.

Weszka, J.S., Dyer, C.R., and Rosenfeld, A. (1976). A Comparative Study of Texture Measures for Terrain Classification. *IEEE Trans. Systems, Man, Cybernetics*, Vol. 6, pp. 269-285.

Wilk, M. B. and Gnanadesikan, R. (1968). Probability plotting methods for the analysis of data. *Biometrika*, Vol. 55 (1), pp. 1-17.

Willamowski, J., Arregui, D., Csurka, G., Dance, C. R. and Fan, L. (2004). Categorising nine visual classes using local appearance descriptors. *Proceedings of ICPR Workshop on Learning for Adaptable Visual Systems*.

Wu, M. and Schölkopf, B. (2007). Transductive Classification via Local Learning Regularisation. In: *Proceedings of Int'l Conf. Artificial Intelligence and Statistics*.

Xu, B., Bu, J., Chen, C., Cai, D., He, X., Liu, W. and Luo, J. (2011). Efficient Manifold Ranking for Image Retrieval. *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, pp. 525-534.*

Xu, B., Bu, J., Chen, C., Wang, C., Cai, D. and He, X. (2013). EMR: A Scalable Graph-based Ranking Model for Content-based Image Retrieval. *IEEE Transactions on* Knowledge and Data Engineering.

Yang, Y., Nie, F., Xu, D., Luo, J., Zhuang, Y. and Pan, Y. (2012). A Multimedia Retrieval Framework based on Semi-Supervised Ranking and Relevance Feedback. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, Vol. 34(4), pp. 723-742.

Ying, L. (2006). Phase unwrapping. *Wiley Encyclopedia of Biomedical Engineering*, Vol. 6, pp. 1-11.

Yong, I., Walker, J. and Bowie, J. (1974). An analysis technique for biological shape. *Comput. Graphics Image Process*, Vol. 25, pp. 357-370.

Zhang, B., Gao, Y., Zhao, S., and Liu, J. (2010). Local Derivative Pattern Versus Local Binary Pattern: Face Recognition With High-Order Local Pattern Descriptor. *IEEE Transactions on Image Processing*, Vol. 19(2), pp. 533-544.

Zhang, D. and Lu, G. (2004). Review of shape representation and description techniques. *Pattern Recognition*, Vol. 37, pp. 1-19.

Zhang, J., Marszalek, M., Lazebnik, S. and Schmid, C. (2007). Local Features and Kernels for Classification of Texture and Object, Categories: A Comprehensive Study. *International Journal of Computer Vision*, Vol. 73(2), pp. 213-238.

Zhong, S., Liu, Y. and Liu, Y. (2011). Bilinear deep learning for image classification. Proceedings of the 19th ACM international conference on Multimedia, pp. 343-352.

Zhou, D., Weston, J., Gretton, A., Bousquet, O. and Schölkopf, B. (2003). Ranking on Data Manifolds. *Advances in Neural Information Processing Systems*.

Zhu, S., Guo, C., Wang, Y., and Xu, Z. (2005). What are Textons? *International Journal of Computer Vision*, Vol. 62(1-2), pp. 121-143.

Zujovic, J. (2011). *Perceptual Texture Similarity Metrics*. PhD thesis, NORTHWESTERN UNIVERSITY.