

Learning Coupled Conditional Random Field for Image Decomposition: Theory and Application in Object Categorization

by

Xiaoxu Ma

Bachelor of Engineering, Tsinghua University, P.R.China (1997)

Master of Engineering, Tsinghua University, P.R.China (2000)

Submitted to the Department of Electrical Engineering and Computer Science

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2008

© Massachusetts Institute of Technology 2008. All rights reserved.

Author

Department of Electrical Engineering and Computer Science

March 13, 2008

Certified by

W. Eric L. Grimson

Bernard Gordon Professor of Medical Engineering

Thesis Supervisor

Accepted by

Terry P. Orlando

Chairman, Department Committee on Graduate Students

Learning Coupled Conditional Random Field for Image Decomposition: Theory and Application in Object Categorization

by

Xiaoxu Ma

Submitted to the Department of Electrical Engineering and Computer Science
on March 13, 2008, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Computer Science

Abstract

The goal of this thesis is to build a computational system that is able to identify object categories within images. To this end, this thesis proposes a computational model of “recognition-through-decomposition-and-fusion” based on the psychophysical theories of information dissociation and integration in human visual perception. At the lowest level, contour and texture processes are defined and measured. In the mid-level, a novel coupled Conditional Random Field model is proposed to model and decompose the contour and texture processes in natural images. Various matching schemes are introduced to match the decomposed contour and texture channels in a dissociative manner. As a counterpart to the integrative process in the human visual system, adaptive combination is applied to fuse the perception in the decomposed contour and texture channels.

The proposed coupled Conditional Random Field model is shown to be an important extension of popular single-layer Random Field models for modeling image processes, by dedicating a separate layer of random field grid to each individual image process and capturing the distinct properties of multiple visual processes. The decomposition enables the system to fully leverage each decomposed visual stimulus to its full potential in discriminating different object classes. Adaptive combination of multiple visual cues well mirrors the fact that different visual cues play different roles in distinguishing various object classes. Experimental results demonstrate that the proposed computational model of “recognition-through-decomposition-and-fusion” achieves better performance than most of the state-of-the-art methods in recognizing the objects in Caltech-101, especially when only a limited number of training samples are available, which conforms with the capability of learning to recognize a class of objects from a few sample images in the human visual system.

Thesis Supervisor: W. Eric L. Grimson

Title: Bernard Gordon Professor of Medical Engineering

Acknowledgments

I would like to convey my utmost gratitude to my advisor, Professor Eric Grimson, for his guidance, support, and encouragement throughout my study and research. I appreciate and really enjoy the motivating and free atmosphere he creates in the group. His truly scientist intuition has made him a constant oasis of ideas and passions in research, which greatly inspires and enriches my growth. I learned from him that research is not only inspiring and exciting, but also enjoyable and satisfying. My discussions with him turned the most difficult obstacles into the most interesting challenges, and eventually became the most joyful accomplishment. The unflinching encouragement and support from him are invaluable for me as a foreign student. As an advisor, a friend, and a spiritual mentor, he gave me more than a professor can give to a student, a friend can offer to another friend. I am indebted to him more than he knows.

I am very grateful to Professor William Freeman and Professor Tomas Lozano-Perez for their very generously spending time to review my thesis. I am really lucky to have these incredibly knowledgeable and kind committee members.

No words can describe the sincere appreciation I have for Professor William Freeman as my academic advisor, especially for his advice and support during my hardest time.

I am blessed with a group of wonderful colleagues. A special thank-you goes to Xiaogang Wang, who introduced fascinating models into automatic visual surveillance and stood side-by-side with me during the developmental stage of the project, also with whom I shared lots of moments of rejoicing and frustration. I appreciate the great help from Gerald Dalley, whom I bored with numerous questions regarding scientific research and not-so-much scientific English words and grammar. I want to thank Biswajit Bose and Chaowei Niu for the patience and friendship they generously offered to me. Thanks also goes to Ce Liu for sharing his experience on how to do research and how to live a meaningful life.

Last but definitely not least, I would like to thank my fiancée Zixiao Pan for her

unconditional love and incredible tolerance throughout this entire journey. Without her precious love and support, and uncountable deliciously cooked meals, I wouldn't have made it to where I am now.

To my parents

Contents

1	Introduction	23
1.1	Dissociation and Integration Nature of Human Perception	24
1.2	Computational Model of Visual Information Decomposition and Fusion	26
1.3	Thesis Outline and Results Preview	28
1.4	Contributions	31
2	Object Recognition Review	33
2.1	Early Years	33
2.1.1	Object-centered Model-based Recognition	34
2.1.2	View-centered Model-based Recognition	36
2.2	Global Appearance Methods	38
2.3	Local Appearance Methods	41
2.3.1	Geometry-based Methods	43
2.3.2	Geometry-free Methods	46
2.4	Summary	48
3	Low-level Image Measurements	49
3.1	Contour Process and Texture Process	49
3.2	Measuring Contourness and Textureness	51
3.2.1	Contourness Measurement and Edge Extraction	51
3.2.2	Textureness Measurement	56
3.3	Summary	60

4	Learning Coupled Conditional Random Field for Image Decomposition	61
4.1	Motivation	61
4.2	Importance of Learning Coupled Conditional Random Field Model	63
4.3	Parametrization of Coupled Conditional Random Field	66
4.4	Learning and Inference of Coupled Conditional Random Field	68
4.4.1	Model Learning with Maximum Pseudolikelihood	68
4.4.2	Derivation of Maximum Pseudolikelihood Learning of Coupled Conditional Random Field	71
4.4.3	Model Learning with Tempered Maximum Pseudolikelihood	75
4.4.4	Parameter Initialization	76
4.4.5	Model Inference	76
4.5	Model Learning and Evaluation	77
4.5.1	Model Learning and Analysis	77
4.5.2	Model Evaluation	86
4.6	Summary	89
5	Matching Decomposed Visual Cues	103
5.1	Choices of Matching Schemes	103
5.2	Local Features	107
5.2.1	Feature Point Extraction	107
5.2.2	Local Appearance Descriptor	109
5.2.3	Color Feature	115
5.3	Matching Individual Decomposed Channels	117
5.3.1	Spatial Pyramid Matching of Local Features	117
5.3.2	Shape Matching with Robust Oriented Chamfer Distance	120
5.4	Experiments	123
5.4.1	Spatial Pyramid Matching of Local Appearance Features	124
5.4.2	Spatial Pyramid Matching of Local Color Features	129
5.4.3	Robust Chamfer Matching on Contour Channels	131

5.5	Summary	133
6	Adaptive Multiple Visual Information Combination	135
6.1	Types of Information Fusion Schemes	136
6.2	Adaptive Information Fusion by Kernel Alignment	137
6.3	Experiments	141
6.3.1	Combining Multiple Scales	141
6.3.2	Complementarity of Weak and Strong Shape Matching in Con- tour Channel	145
6.3.3	Combining Contour, Texture and Color	147
6.3.4	Comparison with Other Methods	157
6.4	Summary	163
7	Conclusion and Discussion	165
7.1	Recent Developments	165
7.2	Comparison to “Learning Probabilities of Boundary”	167
7.3	Summary and Conclusion	167

List of Figures

1-1	A schematic illustration of the concept that various visual cues should play different roles and have different weights in discriminating different class pairs. Three types of visual cue are used: shape contours, texture and color. The line width depicts the relative weights of a visual cue in distinguishing a pair of object classes.	25
1-2	System overview of “recognition-through-decomposition-and-fusion” . .	27
1-3	Contour and texture decomposition examples. The first and fourth rows are object images. Their corresponding contours, decomposed by the proposed coupled Conditional Random Field, are shown in the second and fifth rows. Decomposed textures are show in the third and sixth rows.	29
1-4	Comparison of the proposed scheme to state-of-the-art methods where multiple visual cues were not decomposed and adaptively combined. Dataset used is Caltech-101.	30
2-1	Example scene from the ‘Blocks World’ by Roberts [95]. Objects are made of polyhedra in a uniform background.	35
2-2	Example from the ACRONYM system by Brooks [20]. Objects are modeled by generalized cones and their spatial relationships.	35
2-3	Example of object recognition and localization with the local feature focus method [12].	36

2-4	Example from <i>Localizing Overlapping Parts by Searching the Interpretation Tree</i> by Grimson and Lozano-Perez [52]. Figures show located objects superimposed on images.	36
2-5	The SCERPO system by Lowe [75]. Straight line segments are grouped by perceptual organization. Model primitives are matched to the grouped structures in images. The model is projected onto the image for verification.	37
2-6	Example of <i>Geometric Hashing</i> by Lamdan and Wolfson [68]. A gray scale image of a crane and car is observed. Features such as points and lines are extracted. Combinations of features are matched to model features in a hash table. Transformed model edges and scene features are matched for verification.	38
2-7	Example of <i>Recognizing Solid Objects by Alignment with an Image</i> by Huttenlocher and Ullman [61]. Three solid 3D objects are matched to images. Corners and inflection points in extracted edge segments are used to compute possible alignments. Alignments are verified by matching edge contours.	38
2-8	(a) Three model pictures of a pyramid in [109]. The new images of the pyramid can be generated by linear combinations of the three models. (b) The linear object class model in [112]. New views of faces can be synthesized by linearly combining prototype face images.	40
2-9	Three dimensional color histograms of images of a cereal box (with the black background subtracted) [106]. Color histograms are shown to be invariant to translation and rotation.	40
2-10	(a) Seven eigenfaces used in [108]. (b) Three dimensional manifold defined by the three most prominent dimensions of the eigenspace is used to determine the identity and pose of an object [87].	40
2-11	Two dimensional histograms of two objects [100]. Histograms are the joint statistics of local appearance filter outputs.	41

2-12	In the image retrieval application in [101], corner features are detected. A vector describing local characteristics is formed for each region around a corner point. The collection of vectors is used for matching.	42
2-13	An example of object recognition from local scale-invariant features [77]. Model images of planar box faces are matched in a cluttered scene containing occluded objects. The SIFT approach successfully discovers these objects.	42
2-14	The constellation model of faces learned by [38]. Appearance of parts and examples of detections are shown in the second and third columns respectively.	44
2-15	A sample car image used in the vocabulary construction in [2]. Some examples of learned parts such as wheels, hood, windows, and trunk, are also shown.	44
2-16	The recognition and localization procedure of [71]. Image patches are extracted around interest points. Matching patches cast votes for the object center. Refined hypotheses are used for segmentation.	45
2-17	An example of low distortion correspondences [9]. Feature points in the left-most image are matched to a model image (left center). The entire set of matched features are shown in the right center image. Correspondences after the thin plate spline transform are shown in the right-most image.	45
2-18	A schematic illustration of learning the Potemkin model of a chair [28]. In each view, parts are locally planar. The centroids of parts are estimated, which are used for estimating the skeleton locations.	45
2-19	Bag-of-features model [31]. Affine invariant features are detected from the motorbike image. Spatial relations of features are discarded. The image is represented and classified with a global histogram of feature occurrence.	47

2-20	An example of a face image as a mixture of visual topics [62]. The face topic is shown in yellow, and background topics are shown in blue and cyan.	47
2-21	Random multi-scale subwindows [79] are extracted from three classes of objects. An ensemble of extremely randomized decision trees is learned and used for classification.	47
2-22	The pyramid matching kernel [48] intersects histogram pyramids formed over local features, approximating the optimal correspondences between features of the two images.	48
3-1	Illustration of definition of the contour and texture processes.	50
3-2	Quadrature pair used as base filters.	52
3-3	Quadrature filter bank.	54
3-4	An example of computed orientation energy.	55
3-5	An example of edge extraction.	56
3-6	Texton filter bank.	57
3-7	Illustration of computation of texture gradient.	59
3-8	An example of computed texture gradient.	59
4-1	A simple single-layer Conditional Random Field model for both contour and texture processes.	64
4-2	Coupled Conditional Random Field for modeling contour and texture processes.	65
4-3	Training set for learning coupled Conditional Random Field and single-layer Conditional Random Field. Edge points are first extracted, then a majority of the extracted edge points are labeled as either contour or texture. Contour edges are shown in red color, while texture as yellow. White edges are left unlabeled.	78
4-4	Comparisons of different models (better view in color).	83
4-5	Comparisons of compatibility functions of different models.	85

4-6	Comparisons of per-image precision and recall rates of coupled Conditional Random Field and single-layer Conditional Random Field. . . .	91
4-7	Comparison of contour and texture decomposition by the coupled Conditional Random Field and the single-layer Conditional Random Field models.	100
4-8	Some examples where coupled Conditional Random Field's decomposition performance degrades.	101
5-1	Examples of global features, semi-local features and local features of images.	105
5-2	A schematic illustration of definition of the SIFT descriptor. See text for details.	110
5-3	A schematic illustration of the SIFT descriptors on decomposed channels of contour and texture. See text for details.	112
5-4	Illustration of the definition of color features. (1): The HSV color space is discretized. Each quantized color is regarded as a color word. All color words form the color dictionary. (2): For an image, a small local patch is extracted. (3) and (4): The RGB colors in the local patch are transformed to HSV space. (5) Average Hue, Saturation and Value are computed respectively on the local patch. (6) The color dictionary is referenced to assign the corresponding color word index to the average HSV of the patch.	116
5-5	Illustration of bag-of-features and spatial-pyramid representation. (a) In the bag-of-features representation, global statistics, such as occurrence frequencies, are derived from the ensemble of features in the entire image. (b) In the spatial-pyramid representation, an image is divided into three levels of resolution. For each resolution, statistics such as occurrence frequencies in each spatial cell can be calculated. Statistics from all levels collectively form the representation. Different levels of resolution can have different weights in the representation.	118

5-6	Illustration of computing the robust chamfer distance in one direction. Two laptop images to be matched are shown in (a). Robust chamfer distance is computed between the contour channels X and Y of the two images. To compute the robust chamfer distance from Y to X , Y is transformed by a series of translation and rotation, and the distances $d(X, Y)$ of these transformed images to X are computed with Equation 5.3. The best match of these distances $d(X, Y)$ is kept as the robust chamfer distance from Y to X	122
5-7	Box plots of performance comparisons of different vocabulary sizes on appearance matching in contour and texture channels. In each case, the box draws the first quartile, median and third quartile of recognition rates, the whiskers show the extent of the non-outlier recognition rates, and the outliers (if any) are marked with red cross. Vocabulary size doesn't have a great impact on the recognition performance, although medium-sized vocabularies are slightly better.	125
5-8	Box plots of performance comparisons of different patch sizes on appearance matching in contour and texture channels. In each case, the box draws the first quartile, median and third quartile of recognition rates, the whiskers show the extent of the non-outlier recognition rates, and the outliers (if any) are marked with red cross. It is observed that medium-sized patches give better recognition performance.	128
5-9	Box plots of performance comparisons of different vocabulary sizes on color matching in contour and texture channels. In each case, the box draws the first quartile, median and third quartile of recognition rates, the whiskers show the extent of the non-outlier recognition rates, and the outliers (if any) are marked with red cross. Vocabulary size, or equivalently, the granularity of HSV quantization, doesn't have a great impact on the recognition performance.	130
5-10	Comparison of average per-class recognition rates of different kernels.	132

6-1	Comparison of average per-class recognition rates of the combined SIFT-spatial-pyramid-matching kernel $K_{SIFTcomb}$, the robust chamfer matching kernel $K_{rchamfer}$ and the combined contour kernel $K_{contour}$ which adaptively integrates $K_{cSIFT25}$, and $K_{cSIFT50}$ and $K_{rchamfer}$	146
6-2	Comparison of average per-class recognition rates of the combined contour kernel $K_{contour}$, and the adaptive kernel K_{all_comb} and the average kernel K_{all_avg} that combine matching schemes of contour, texture and color.	148
6-3	Example images of Bonsai and Joshua Tree.	151
6-4	Multi-dimensional Scaling (MDS) embedding of different kernels: contour kernel, texture kernel, color kernel and adaptively combined kernel, for the class of Bonsai versus the class of Joshua Tree.	151
6-5	Example images of Pizza and Soccer Ball.	152
6-6	Multi-dimensional Scaling (MDS) embedding of different kernels: contour kernel, texture kernel, color kernel and adaptively combined kernel, for the class of Pizza versus the class of Soccer Ball.	152
6-7	Learned adaptive weights for contour, texture and color when combining visual cues for classifying Caltech-101 (better view in color).	156
6-8	Comparison of the proposed scheme to state-of-the-art methods where multiple visual cues were not adaptively combined.	159
6-9	The confusion matrix for classification of Caltech-101 by the proposed method in this thesis.	160
6-10	Comparison of the proposed scheme to state-of-the-art methods where multiple distance functions or kernels were adaptively combined.	162
7-1	Comparisons of the probabilities of boundary obtained by the learned model by Martin <i>et al.</i> [80] and the decomposed contour channels by the coupled Conditional Random Field model.	168

List of Tables

4.1	Definition of the coupled Conditional Random Field for contour and texture processes of an image.	65
4.2	Evidence and compatibility functions of the proposed coupled Conditional Random Field. See text for details.	67
4.3	Evidence and compatibility functions of a single-layer Conditional Random Field model of contour and texture processes. See text for detailed explanation.	79
4.4	Different compatibility functions of a single-layer Conditional Random Field.	79
4.5	Learned parameters for different models of coupled Conditional Random Field and single-layer Conditional Random Field.	81
4.6	Performance evaluation of different models.	88
5.1	Comparison of average per-class recognition rates of appearance matching on Caltech-101 in contour and texture channels respectively, with different visual vocabulary sizes. Numbers in parenthesis are standard deviation.	124
5.2	Average per-class recognition rates on Caltech-101 with SIFT descriptors in contour and texture channels respectively. Different patch sizes are tested. Numbers in parenthesis are standard deviation.	127
5.3	Comparison of average per-class recognition rates of appearance matching on Caltech-101 of some current best methods. Numbers in parenthesis are standard deviation.	127

5.4	Comparison of average per-class recognition rates of color matching on Caltech-101 in contour and texture channels respectively, with different visual vocabulary sizes. Numbers in parenthesis are standard deviation.	129
5.5	Comparison of average per-class recognition rates of different kernels. Numbers are in percentile.	132
6.1	Comparison of recognition performance on Caltech-101 with kernels $K_{cSIFT25}$ and $K_{cSIFT50}$ (spatial pyramid matching kernels for patch size 25 and 50 respectively), their average combination $K_{SIFT_{avg}}$ and adaptive combination K_{comb} , for 30 training samples per class. The second row are average per-class recognition rates on the contour channel. The third are average per-class recognition rates on the texture channel. Numbers in parenthesis are standard deviation.	142
6.2	The left three columns are the 10 classes where small patches are of more importance in recognition. The first column is the class name, the second is the weight of the kernel $K_{cSIFT25}$ with patches of size 25, and the third column is exemplar images of corresponding classes. The right three columns show the 10 classes where large patches of size 50 play more important roles in recognition. See text for details.	144
6.3	Comparison of average per-class recognition rates of the combined SIFT-spatial-pyramid-matching kernel $K_{SIFT_{comb}}$, the robust chamfer matching kernel $K_{rchamfer}$ and the combined contour kernel $K_{contour}$ which adaptively integrates $K_{cSIFT25}$, and $K_{cSIFT50}$ and $K_{rchamfer}$. Numbers are in percentile.	146
6.4	Average per-class recognition rates of the combined contour kernel $K_{contour}$, and the adaptive kernel $K_{all_{comb}}$ and the average kernel $K_{all_{avg}}$ that combine matching schemes of contour, texture and color. Numbers are in percentile.	148
6.5	Classes where the visual cue of contour plays a larger role for recognition compared with other classes.	153

6.6	Classes where the visual cue of texture plays a larger role for recognition compared with other classes.	154
6.7	Classes where the visual cue of color plays a larger role for recognition compared with other classes.	155
6.8	Comparison of the proposed scheme to state-of-the-art methods where multiple visual cues are not adaptively combined.	158
6.9	Comparison of the proposed scheme to state-of-the-art methods where multiple distance functions or kernels were adaptively combined. . . .	161

Chapter 1

Introduction

The goal of this thesis is to build a computational system that is able to identify object categories within images. The physical visual world is a rich and complex source of information. Natural images of objects generally contain a great amount of rich visual information about the objects of interest and their backgrounds. Yet, in spite of the complexities of many visual tasks, the human visual system can effortlessly and efficiently perceive and form meaningful interpretations of image contents. For decades, computer vision researchers have pursued computational models that emulate the performance of the human visual system. This thesis proposes a computational model based on the associationism theories of information dissociation and integration in human visual perception. Specifically, a computational model of visual information decomposition is developed to simulate the dissociative nature of human perception; a system built on visual information decomposition and adaptive fusion is introduced as a counterpart to the integration process in the human visual system; and the various properties of the proposed computational system are studied.

1.1 Dissociation and Integration Nature of Human Perception

It has been shown in cognitive science and psychophysical science that a visual scene is analyzed at an early stage by specialized populations of receptors that respond selectively to different properties such as orientation, color, spatial frequency, and that map these properties into different areas of the brain [116]. Moreover, visual form recognition requires the analysis of both local object features and global shape contours [59]. These findings demonstrated that visual stimuli are first processed in a dissociative manner and then integrated in later processing stages in human perception.

For example, a case study with a patient, who had very grave difficulty in recognizing common objects although he could recognize their texture, provided evidence of a dissociation between texture processing and shape integration [1]. Another case study revealed that a patient was impaired at recognizing objects on the basis of texture information, whereas shape recognition on the basis of contours was comparatively preserved [6]. The results suggest that contour-based and texture-based processing are separable operations in object perception. Furthermore, both psychophysical [26] and physiological studies [36] suggest that features such as texture are extracted and analyzed in separate channels, whereas shape contours are perceived in a specialized region, and these separate channels are later combined into a common representation of the visual scene. Evidence of this theory was also found in a patient who was observed to be able to distinguish local features of complex patterns but was unable to integrate them into a whole configuration [93]. Thus, an anatomy-based integrated system of visual information processing is proposed by Essen *et al.* where ordinary visual tasks involve the coordinated use of multiple types of information, the different types of information are represented explicitly in separated processing streams, and multi-channel information converges at the highest level of processing [35].

As an emulation of human perception, it is desirable to design object recognition systems such that multiple visual cues such as contour and texture in images are de-

composed into explicitly different channels, perceptual potential of each decomposed visual stimuli is fully leveraged, and, in the highest level, the decomposed visual information is selectively combined for recognizing different classes of objects. This aspect of visual information fusion in object recognition has been relatively lacking in many popular approaches. For example, many of the approaches based on local appearance features mix all pixels in a local region and describe the region as an integral entity, essentially giving uniform or fixed weights to all the information contained in the region. However, based on dissociation and integration models of human perception, it is more sensible to assume that various visual cues should play different roles in discriminating different class pairs. Figure 1.1 gives a schematic illustration of this postulation. For instance, to classify *beaver* versus *emu*, the shape information may be more important since both classes have similar texture and color; while for *laptop* versus *gerenuk*, all visual cues such as shape, texture and color could be important for discriminating the two classes.

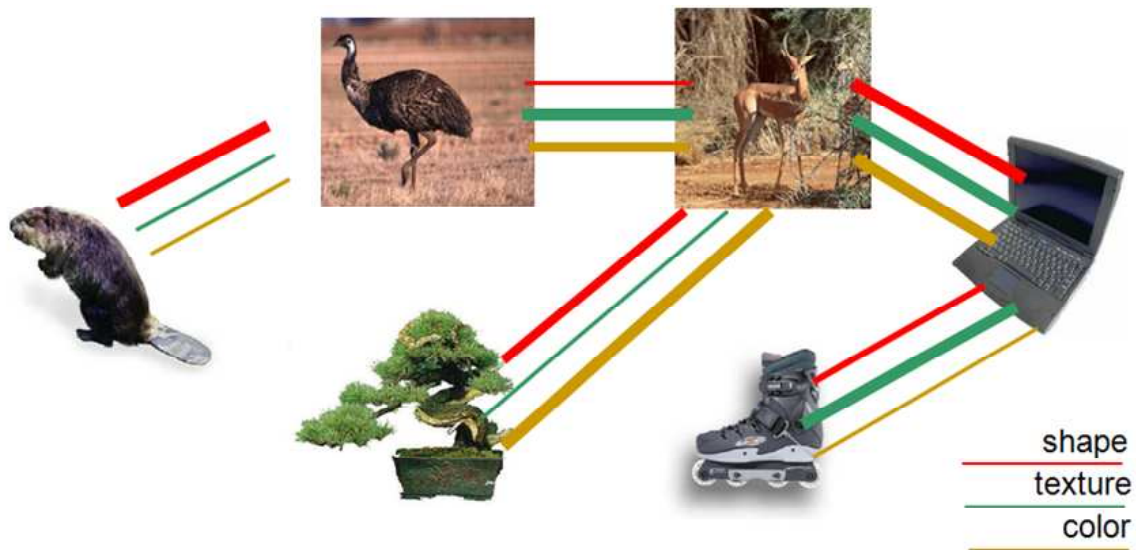


Figure 1-1: A schematic illustration of the concept that various visual cues should play different roles and have different weights in discriminating different class pairs. Three types of visual cue are used: shape contours, texture and color. The line width depicts the relative weights of a visual cue in distinguishing a pair of object classes.

1.2 Computational Model of Visual Information Decomposition and Fusion

In this thesis, the proposed computational system of visual information decomposition and fusion has four key components, as shown in Figure 1-2. For each input image, the proposed system has the following processing streams:

1. **Low-level image measurements:** Filter banks are used to focus attention on a set of pixels of interest such as edge points, and to give compact measurements for pixels of interest.
2. **Mid-level modeling:** A novel coupled Conditional Random Field is proposed to model the interactive processes of various visual cues. The coupled Conditional Random Field is shown to be superior to a single-layer Conditional Random Field for the task of contour and texture decomposition.
3. **High-level matching of individual visual cues:** Various matching schemes, such as appearance matching, color matching, and shape matching, can be employed to match each individual cue. The decomposition of contour and texture naturally enables methods of matching different visual stimuli separately to fully leverage each perceptual cue.
4. **High-level adaptive combination of multiple visual cues:** As one implementation, a principled method of adaptively combining the decomposed contour and texture channels is incorporated to integrate various visual cues into a complex whole. Different visual cues play different roles in discriminating different classes in the integrated system.

The above scheme is referred as “recognition-through-decomposition-and-fusion” in this thesis. The key concept of this computational model is to achieve better recognition performance by decomposing and recombining multiple disparate visual cues in object images.

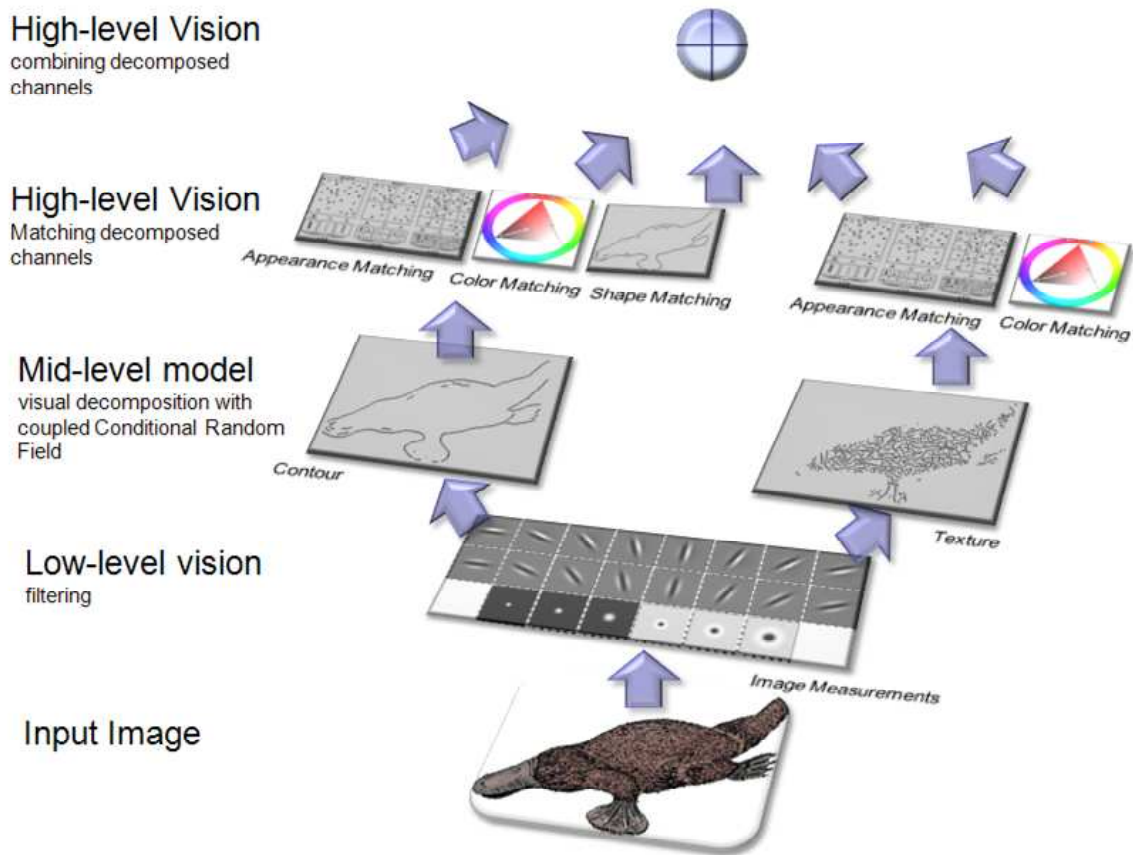


Figure 1-2: System overview of “recognition-through-decomposition-and-fusion”.

1.3 Thesis Outline and Results Preview

In this thesis, Chapter 2 reviews previous and current state-of-the-art object recognition systems. Chapter 3 introduces the low level image measurements used in the model. Chapter 4 develops the mathematical forms of the coupled Conditional Random Field and its learning and inference for contour and texture decomposition. The learned coupled Conditional Random Field model is shown to be able to achieve good decomposition of contour and texture in natural images. Some results are shown Figure 1-3. With the decomposed visual information, suitable matching schemes are introduced in Chapter 5 for each visual cue to address their different characteristics. Chapter 6 employs the kernel alignment theory to adaptively combine multiple visual cues, and the effectiveness of “recognition-through-decomposition-and-fusion” is demonstrated with recognition experiments on the challenging dataset of Caltech-101.

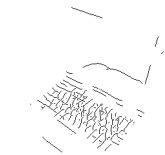
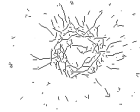
Figure 1-4 shows the recognition performance of the proposed scheme on Caltech-101, as compared with some of the state-of-the-art methods in which visual information is treated in an integral manner without visual decomposition. Compared with one of those top methods [51], the proposed scheme in this thesis achieves recognition improvement of about {7.24%, 5.75%, 4.31%, 2.02%, 1.98%, 2.24%,} for {5,10,15,20,25,30} training samples per class respectively. These comparison experiments demonstrate the effectiveness of the proposed visual decomposition and recombination scheme for object recognition. The performance improvements are more significant when only a few training samples, *e.g.*, 5, 10 or 15, are available for each class. This suggests that when there are not enough training samples, it is more important to decompose various visual cues, leverage each of them to their full potential and recombine them for a better understanding of image contents. This conforms with the capability of learning to recognize a class of objects from a few sample images in the human visual system.



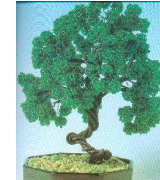
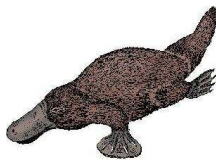
Images



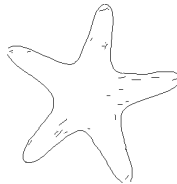
Decomposed contour by coupled Conditional Random Field



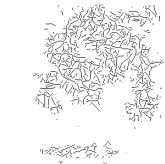
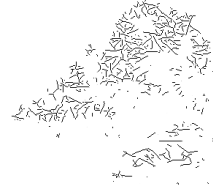
Decomposed texture by coupled Conditional Random Field



Images



Decomposed contour by coupled Conditional Random Field



Decomposed texture by coupled Conditional Random Field

Figure 1-3: Contour and texture decomposition examples. The first and fourth rows are object images. Their corresponding contours, decomposed by the proposed coupled Conditional Random Field, are shown in the second and fifth rows. Decomposed textures are show in the third and sixth rows.

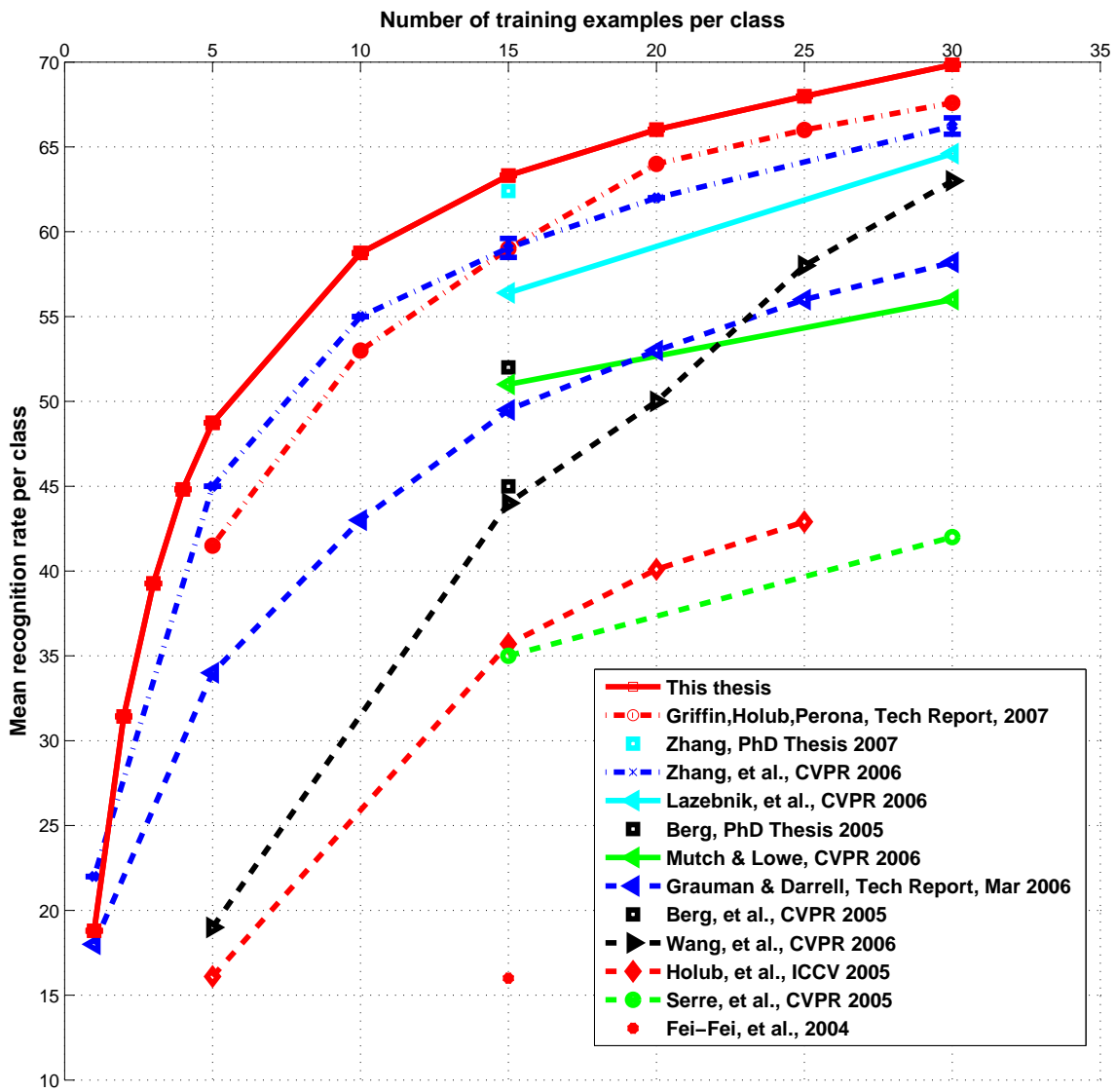


Figure 1-4: Comparison of the proposed scheme to state-of-the-art methods where multiple visual cues were not decomposed and adaptively combined. Dataset used is Caltech-101.

1.4 Contributions

The main contribution of this thesis can be summarized as follows:

1. A computational model is developed based on the associationism theories of information dissociation and integration in human visual perception.
2. A novel coupled Conditional Random Field model is proposed as an image model of interactive contour and texture processes. Model learning and inference are formulated. A set of training images and test images is created to evaluate the proposed model. The learned model is demonstrated to be able to capture many distinct characteristics of contour and texture channels, where a single-layer Conditional Random Field model has to make compromises. Empirical evaluation shows the superiority of the coupled Conditional Random Field model.
3. The coupled Conditional Random Field model is an important extension for modeling image processes. It is expected to be a fundamental and general model of images. Additional perceptual properties such as regions and class-specific shapes can be incorporated in this framework. The proposed model is potentially extendable to versatile applications such as image statistics, image rendering and computer graphics.
4. In Chapter 5, various matching methods are studied for the decomposed contour and texture channels. It is shown that recognition based on the decomposed contour alone achieves relatively comparable performance to many previous best results. This demonstrates that salient contours play the most important role and are the dominant visual information in recognizing objects in Caltech-101.
5. Chapter 6 shows that adaptive selection of discriminative visual cues for different classes helps to improve object recognition performance. This corroborates the validity of dissociation and integration nature of human visual perception, which inspires the proposed model in the first place. Especially when there are

only a limited number of training samples, it is more important to decompose visual stimuli, fully leverage them and adaptively recombine them.

6. Various aspects of recognizing objects in Caltech-101 are studied. It is observed that combining multiple scales helps to boost recognition performance, which indicates that a certain degree of inter-class scale variability exists in Caltech-101. Weak geometric matching and strong geometric matching are shown to be complementary. When combining contour, texture and color, contour is observed to be the most prominent visual cue for recognizing objects in Caltech-101; texture and color information play comparable roles, with texture information slightly more important.

Chapter 2

Object Recognition Review

Object recognition has been an active research area of computer vision for more than forty years. Over the decades, this research frontier has been rapidly advancing with persevering efforts of diligent researchers in related scientific and engineering fields. Advancements in many aspects, such as more in-depth understanding of the human visual system, more sophisticated mathematical tools, more and more challenging data collections, and better computational power, have brought object recognition to where it is today. This chapter summarizes many of the popular object recognition methods and systems from the early years to the current state of the art. It is by no means an attempt for a complete review of all historical and contemporary efforts on the subject of object recognition. The review in this chapter aims at rendering a general picture of the evolution of object recognition, and helps the readers to put the work in this thesis into the context of object recognition research.

2.1 Early Years

The early years of computational object recognition started with model-based object recognition, where the knowledge of an object was provided by an explicit model of its shape and appearance. Model-based object recognition systems can be roughly divided into two categories based on the representation that was used: object-centered representation and view-centered representation. The object-centered representation affixes a single coordinate system to the object and uses this coordinate system to lo-

cate and match various constituent parts. The view-centered representation describes objects using a set of 2D characteristic views or aspects, with each characteristic view describing how the object appears from a single viewpoint.

Many of the early approaches used edges or boundaries as features, and applied geometric, relational and/or topological constraints for recognizing objects in scenes. These methods achieved their strength in insensitiveness to illumination changes and capability of recovering 2D and 3D poses.

2.1.1 Object-centered Model-based Recognition

Blocks World(1963, 1975) Early approaches to object recognition made strong simplifying assumption about the real visual world. The ‘Blocks World’ model by Roberts [95], as shown in Figure 2-1, assumed that objects of interest such as blocks, wedges, and prisms were made of combinations of polyhedra and appeared in a uniform background. Edges and lines were detected as features. Recognition was done with matching the polygon structures to the models by topological constraints.

ACRONYM(1981) Brooks [20] built a rule-based system named ACRONYM to interpret 2D images based on 3D models. Three-dimensional geometric objects were modeled as *generalized cones* and their spatial relationships. The system was based on the prediction-hypothesis-verification paradigm. Initially, edges were combined into features such as ribbons and ellipses. The interpreter then looked for matches between the model as a set of generalized cones and the observed features based on prediction of the ways the generalized cones could appear in the image. Interpretation proceeded by combining local matches of shapes into more global matches, requiring consistency among matches. An example of the generalized cones representation of a Boeing-747 is shown in Figure 2-2.

Local Feature Focus Method(1982) Bolles and Cain [12] developed a system for object recognition called the local-feature-focus method. A list of distinct local features was first specified with feature types, their positions and orientations relative to the center of the object. Each feature was augmented to a feature-centered subgraph where a sufficient set of nearby secondary features were included. Object

recognition and localization were done with matching clusters of consistent secondary features and focus features, hypothesizing the location and orientation of potential occurrences of an object in an image, and verifying potential detections by checking the boundaries of the hypothesized objects. Figure 2-3 shows an example of detecting hinge objects in an image.

Interpretation Tree(1987) The *interpretation tree* [52] approach identified and located objects in a scene by matching positions and normals of surfaces to those in 3D models. The objects were modeled as polyhedra and sets of planar faces. The method proceeded by examining all hypotheses about correspondences between sensed data and object surfaces. Heuristics and decoupled and coupled constraints were applied to significantly prune the interpretation tree to achieve an efficient solution. Figure 2-4 shows the recognition and detection results by the interpretation tree approach in a scene with overlapping objects.

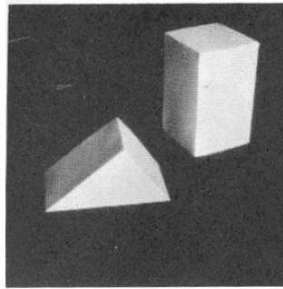


Figure 2-1: Example scene from the ‘Blocks World’ by Roberts [95]. Objects are made of polyhedra in a uniform background.

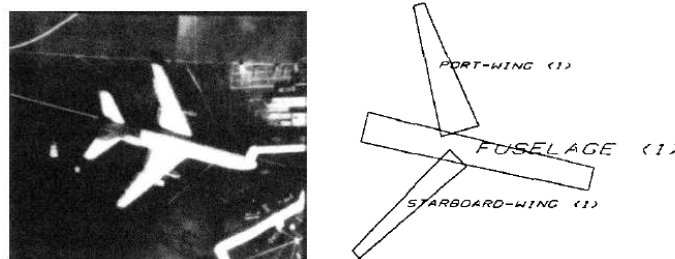


Figure 2-2: Example from the ACRONYM system by Brooks [20]. Objects are modeled by generalized cones and their spatial relationships.

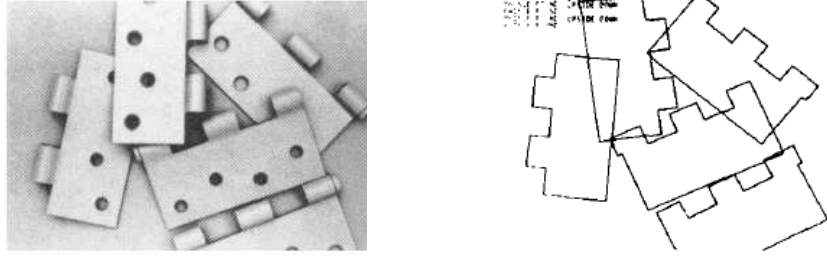


Figure 2-3: Example of object recognition and localization with the local feature focus method [12].

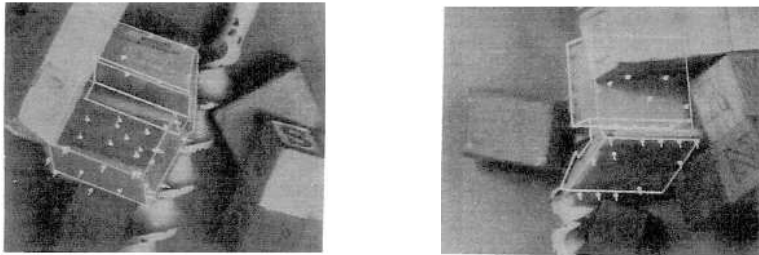


Figure 2-4: Example from *Localizing Overlapping Parts by Searching the Interpretation Tree* by Grimson and Lozano-Perez [52]. Figures show located objects superimposed on images.

2.1.2 View-centered Model-based Recognition

SCERPO(1987) In the SCERPO (Spatial Correspondence, Evidential Reasoning, and Perceptual Organization) system developed by Lowe [75], the goal was to recognize and locate rigid 3D objects in a single gray-scale image. First, a process of perceptual organization was used to form groupings in the image. Pairs of straight lines were combined into perceptual structures, that is, instances of collinearity, proximity, and parallelism. Then these primitive structures were combined into larger, more complex structures such as trapezoid shapes. These structural patterns were used to limit the search space during model matching. Unknown viewpoint and model parameters were solved. Finally, hypotheses were verified by spatial correspondences of the back-projection of 3D models and observed image edges. An example of object detection in a cluttered scene is shown in Figure 2-5.

Geometric Hashing(1988) Geometric hashing was proposed by Lamdan and Wolfson [68] as a general method for model-based object recognition, especially the recognition of 3D objects in occluded scenes from 2D gray scale images. The underlying idea of geometric hashing was to extract geometric features from a set of model object images and the model information was encoded and stored in an indexing data structure such as a hash table. During the recognition phase, a set of features was extracted from the scene and the method accessed the previously constructed hash table for matching the scene features to model features. Figure 2-6 shows that the Geometric Hashing approach correctly recognizes the crane and the car in a scene although the objects occlude each other.

Recognizing Solid Objects by Alignment(1990) Huttenlocher and Ullman [61] presented a model-based method for recognizing solid objects with unknown 3D position and orientation from a single 2D image. In the first stage, possible alignments were computed to generate transformations from the model to the image. Local features derived from corners and inflection points were used for the computation of possible alignments. In the second stage, each of these hypothesized matches was verified by comparing the complete edge contours of the aligned objects with the observed image edges. In Figure 2-7, 3D object models are matched to the scene by the alignment method.

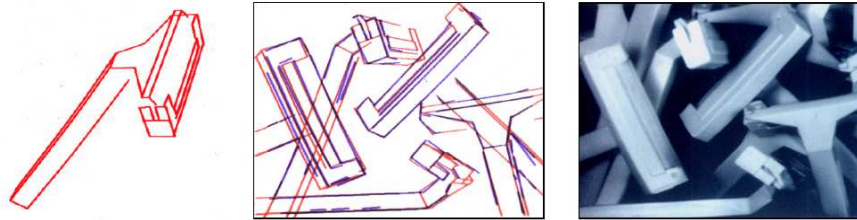


Figure 2-5: The SCERPO system by Lowe [75]. Straight line segments are grouped by perceptual organization. Model primitives are matched to the grouped structures in images. The model is projected onto the image for verification.

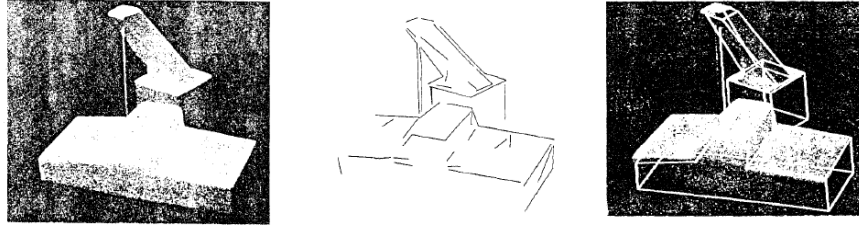


Figure 2-6: Example of *Geometric Hashing* by Lamdan and Wolfson [68]. A gray scale image of a crane and car is observed. Features such as points and lines are extracted. Combinations of features are matched to model features in a hash table. Transformed model edges and scene features are matched for verification.

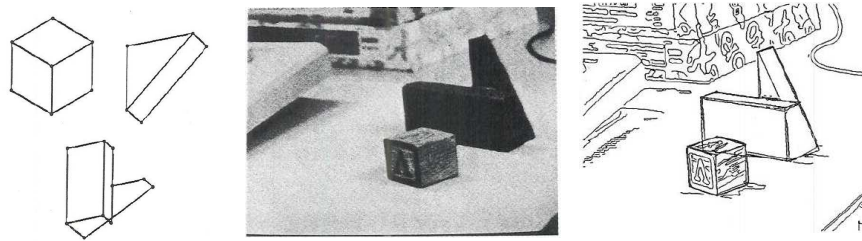


Figure 2-7: Example of *Recognizing Solid Objects by Alignment with an Image* by Huttenlocher and Ullman [61]. Three solid 3D objects are matched to images. Corners and inflection points in extracted edge segments are used to compute possible alignments. Alignments are verified by matching edge contours.

2.2 Global Appearance Methods

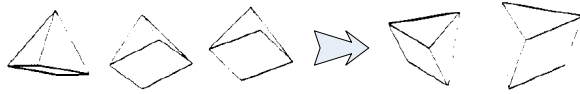
During late 1980's and early 1990's, the object recognition field has seen a gradual transition from 3D model based representation to 2D multi-view based representation of objects. Much psychological research work [21, 22, 34] has provided support for using a set of 2D views to describe and recognize 3D objects. The early methods in this direction represented objects by storing their global appearance information. Viewpoint and lighting invariance were achieved by capturing many images from various viewpoints and under various illumination. Object recognition in a new image was carried out by finding the most similar image in the stored database.

Linear Combination of Views(1991, 1995) Ullman and Basri [109] followed the theory that visual object recognition requires the matching of an image with a set of models stored in memory. They represented a 3D object with the linear combination of 2D images of the object. Vetter and Poggio [112] proposed a method to generate virtual new views given one view of an object by exploiting prior knowledge. An example-based approach was used as an alternative to 3D model-based approach. Linear combination of views were used to synthesize new views. Examples of linear combination representation of objects are shown in Figure 2-8.

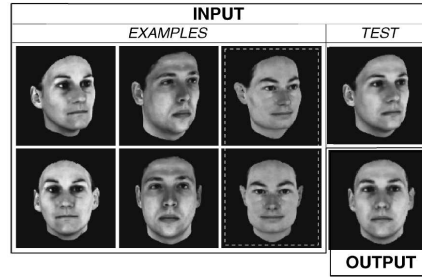
Color Histogram Matching(1991, 1995) Swain and Ballard [106] demonstrated that color histograms of multicolored objects provide a robust and efficient cue for indexing into a large database of object models. Object recognition was done by using *histogram intersection*, which matches the model and image color histograms. Color indexing was shown to be invariant to translation and rotation, and was insensitive to deformations and occlusions. Fun and Finlayson [45] extended color indexing to deal with changing lighting conditions by using ratios of color RGB triples. Three dimensional color histograms of images of a cereal box are shown in Figure 2-9.

Eigenspace Representation(1991, 1995) Turk and Pentland [108] presented an approach to the detection and identification of human faces by using a compact set of characteristic faces in the eigenspace, projecting face images onto the eigenspace, and checking if the images are sufficiently close to the “face space”. Murase and Nayar [87] extended the eigenspace method to recognize 3D objects. Images of objects captured over a wide range of viewpoints were represented in a 3D eigenspace. A novel view of an object was projected onto the eigenspace, and projection coefficients determined the identity and pose of the object. The ‘eigenface’ for face images and 3D eigenspace of a 3D object are shown in Figure 2-10.

Local Appearance Histogram(1996) Schiele and Crowley [100] presented a technique where appearances of objects were represented by the joint statistics of outputs of local filters such as Gaussian derivatives or Gabor filters, as in Figure 2-11. Probabilistic recognition based on the holistic representation without correspondences was developed.



(a)



(b)

Figure 2-8: (a) Three model pictures of a pyramid in [109]. The new images of the pyramid can be generated by linear combinations of the three models. (b) The linear object class model in [112]. New views of faces can be synthesized by linearly combining prototype face images.

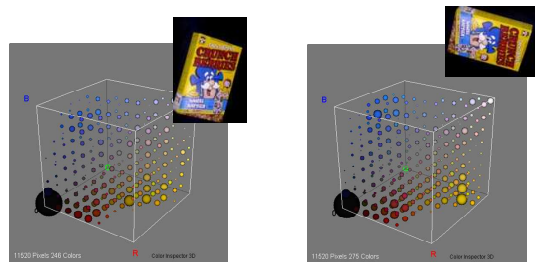


Figure 2-9: Three dimensional color histograms of images of a cereal box (with the black background subtracted) [106]. Color histograms are shown to be invariant to translation and rotation.

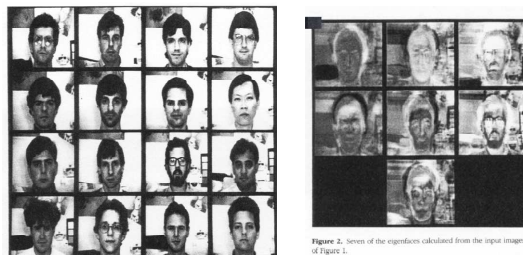
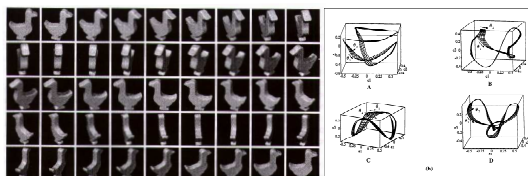


Figure 2. Seven of the eigenfaces calculated from the input images of Figure 1.

(a)



(b)

Figure 2-10: (a) Seven eigenfaces used in [108]. (b) Three dimensional manifold defined by the three most prominent dimensions of the eigenspace is used to determine the identity and pose of an object [87].

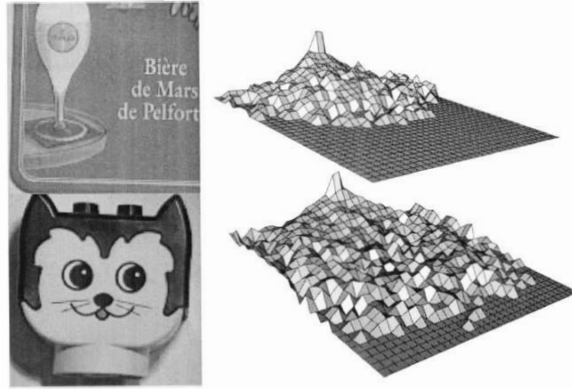


Figure 2-11: Two dimensional histograms of two objects [100]. Histograms are the joint statistics of local appearance filter outputs.

2.3 Local Appearance Methods

The appeal of global appearance methods lies in its simplicity and computational efficiency. However, global appearance methods are sensitive to background clutter and occlusion. Another drawback is the large amount of training data usually required by many global appearance methods. In the late 1990's, the field of object recognition was gradually shifted to utilize local appearance information to recognize objects.

Methods using local regions for object recognition made their debut in the seminal work of Schmid and Mohr [101]. The general idea of using local regions for object recognition is to represent objects by the appearance of a set of, often hundreds of, local regions or patches extracted from the object images. Recognition typically proceeds by matching the local regions in new images to the local regions of model objects in the database. In many implementations, geometric modeling can be incorporated to achieve additional discriminative power.

Schmid and Mohr(1997) Schmid and Mohr [101] proposed using a collection of automatically detected local regions to represent objects. Interest points were extracted using a Harris corner detector [53]. Each local region around an interest point was described by a vector of rotationally invariant gray-scale measures, as illustrated in Figure 2-12. Object image retrieval was carried out with a voting algorithm and semilocal constraints.

Scale Invariant Feature Transform(1999) Another influential paper is the object recognition from local scale-invariant features by Lowe [77]. The idea was to represent objects by a set of circular regions detected by a scale-invariant Difference of Gaussian operator. Each local region was described with a SIFT (Scale Invariant Feature Transform) descriptor. During recognition, extracted SIFT descriptors from a novel image were matched by a nearest neighbor scheme to the SIFT descriptors of model objects. A Hough-base scheme was used to help a geometric verification process to determine the affine pose transformation of the objects. Examples of object recognition and localization are shown in Figure 2-13.

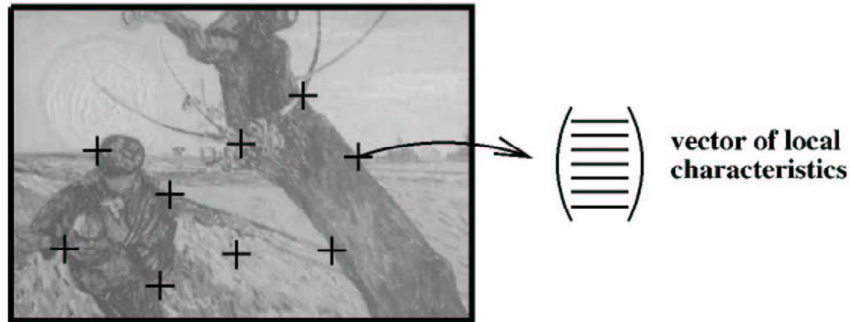


Figure 2-12: In the image retrieval application in [101], corner features are detected. A vector describing local characteristics is formed for each region around a corner point. The collection of vectors is used for matching.



Figure 2-13: An example of object recognition from local scale-invariant features [77]. Model images of planar box faces are matched in a cluttered scene containing occluded objects. The SIFT approach successfully discovers these objects.

Since the inception of the idea of recognition with local regions, local appearance based methods have received immense attention in the field of object recognition, and have acted as the fundamental building blocks of many state-of-the-art approaches. Many of these approaches can be roughly categorized into two types: methods with explicit geometric modeling, and methods without geometric modeling.

2.3.1 Geometry-based Methods

Constellation Model(2000, 2003) The constellation model developed by Burl, Weber, Welling and Perona [24, 114, 115] extended the idea of “pictorial structure” by Fischler and Elschlager [40] by building a face model with manually labeled landmark points [24] or automatically detected interest points [114, 115]. The model was completed with the learned appearance of landmark regions and the joint distribution of their relative locations. Fergus *et al.* [38] further improved the constellation model to incorporate appearance variability and scale-invariance. A Bayesian extension to tackle the problem with a limited number of training images was proposed by Fei-Fei *et al.* [37]. Figure 2-14 illustrates the learned constellation model for faces in [38].

Sparse, Part-Based Representation(2004) Agarwal *et al.* [2] studied car classification by learning a sparse, part-based representation. A vocabulary of parts was automatically constructed from a training set, as shown in Figure 2-15. The object model was formed by the learned parts and their spatial relations. The recognition proceeded with a sliding window searching exhaustively in a query image. Each window was classified with a sparse-network-of-winnows classifier [97] to detect cars.

Implicit Shape Model(2004) Leibe *et al.* [71] devised an approach to simultaneously segmenting and recognizing objects in images. Strong supervision was used in the learning stage with each training image manually segmented. First, a codebook of local appearance was learned from automatically detected interest points. Then an *implicit shape model* was built by capturing the locations of parts relative to the object center. During recognition, extracted image patches in test images were first matched to codebook entries, and these matches voted for the position of the object center in the test images. Refined hypotheses were used for segmentation. This

recognition and localization procedure is illustrated in Figure 2-16.

Low Distortion Correspondences(2005) Berg *et al.* [9] formulated object recognition as a problem of deformable shape matching. The method computed correspondences between randomly sampled feature points based on the similarity of geometric blur descriptors [10] as well as the geometric distortion. Given the correspondences, a regularized thin plate spline transformation [13] was estimated to compute a dense correspondence between the test image and the model image, which was in turn used in a nearest neighbor framework for recognition. An example is shown in Figure 2-17.

Potemkin model(2007) Chiu *et al.* [28] proposed a Potemkin model for describing 3D objects. A Potemkin model represents a 3D object with a collection of nearly planar parts, with a skeleton defined by the arrangement of the part centroids, as illustrated in Figure 2-18. Arbitrary virtual views can be generated from a set of observed views of a 3D object, with the knowledge learned from some simple objects. This augmented set of views can then be fed into any view-dependent 2D part-based recognition system for object recognition and localization.

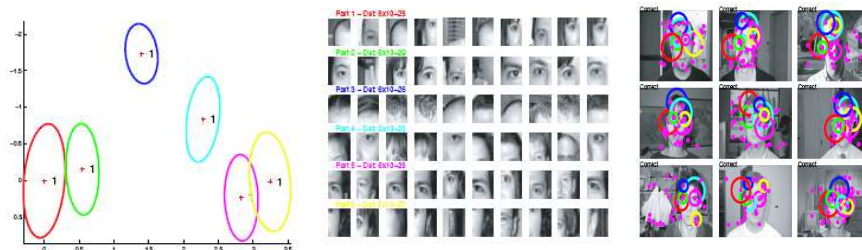


Figure 2-14: The constellation model of faces learned by [38]. Appearance of parts and examples of detections are shown in the second and third columns respectively.

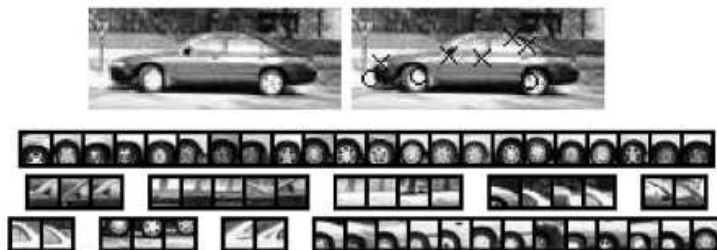


Figure 2-15: A sample car image used in the vocabulary construction in [2]. Some examples of learned parts such as wheels, hood, windows, and trunk, are also shown.

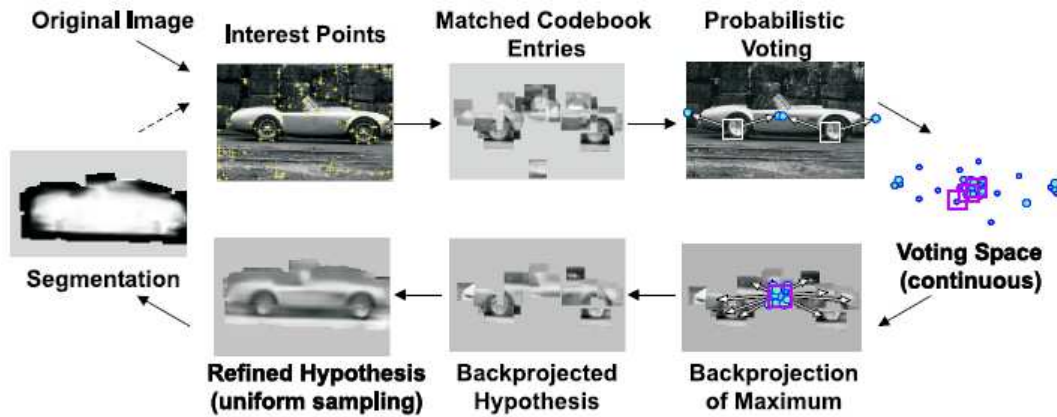


Figure 2-16: The recognition and localization procedure of [71]. Image patches are extracted around interest points. Matching patches cast votes for the object center. Refined hypotheses are used for segmentation.

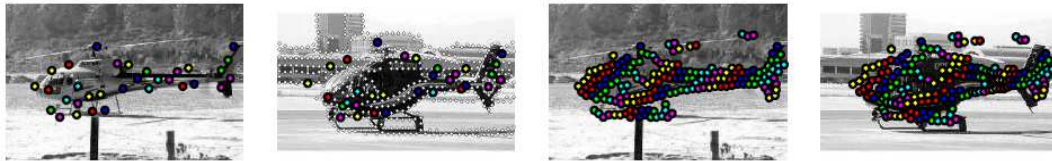


Figure 2-17: An example of low distortion correspondences [9]. Feature points in the left-most image are matched to a model image (left center). The entire set of matched features are shown in the right center image. Correspondences after the thin plate spline transform are shown in the right-most image.

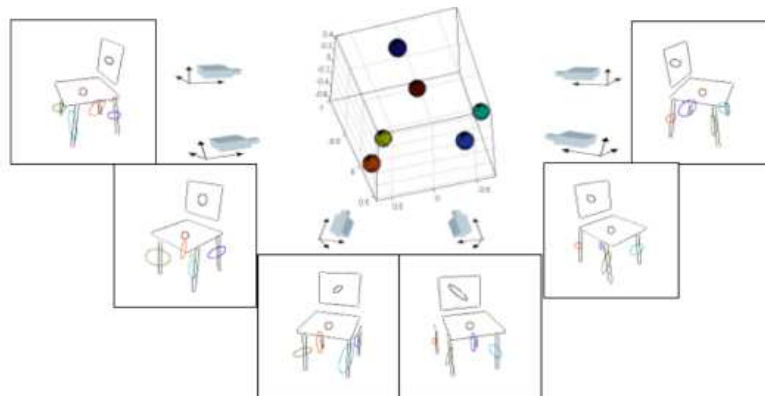


Figure 2-18: A schematic illustration of learning the Potemkin model of a chair [28]. In each view, parts are locally planar. The centroids of parts are estimated, which are used for estimating the skeleton locations.

2.3.2 Geometry-free Methods

Bag of Keypoints(2004) The bag-of-keypoints approach by Csurka *et al.* [31], as illustrated in Figure 2-19, regarded the orderless collection of local patches as the representation of objects. The method was based on vector quantization of SIFT descriptors of affine invariant feature points. Each image was represented by a histogram of the number of occurrences of each quantized visual word. An SVM was trained and used for detecting the presence of objects in images.

Natural Language Models(2005, 2006) In [62], Sivic *et al.* introduced natural language models such as probabilistic Latent Semantic Analysis (pLSA) [55] into the bag-of-words representation of object classes. In its original applications, pLSA represented textual documents with a structure of words-topics-documents, with the capability of unsupervisedly discovering the hidden variable of topics. Sivic *et al.* successfully translated pLSA into the domain of visual object recognition by treating quantized appearance descriptors as words, object classes as topics, and images as documents. Visual words and topics in a face image are shown in Figure 2-20. A non-parametric version of natural language models, Hierarchical Dirichlet Processes, was applied by Sudderth *et al.* [105] and Wang *et al.* [113].

Random Subwindows(2005) Maree *et al.* [79] represented images with a set of randomly extracted subwindows at random locations and scales. Then subwindows were normalized in size and described by a feature vector of 768 numerical values in the HSV color space. Ensembles of extremely randomized decision trees were employed for recognizing objects. Figure 2-21 gives a schematic illustration of object representation and recognition based on random subwindows.

Pyramid Matching Kernel(2005) In [48], Grauman and Darrell designed a new fast kernel function which maps unordered feature sets using multi-resolution histograms. The approach didn't rely on quantization of local patches into visual words. Instead, the algorithm operated directly on the high dimensional feature space which was discretized in multiple resolutions. Histograms of features were formed in this multi-resolutions space. Object recognition was done by matching with weighted his-

togram intersection. Figure 2-22 demonstrates this matching scheme for two panda images. Lazebnik later extended pyramid matching to incorporate rough geometric correspondences in the work of spatial pyramid matching [70].

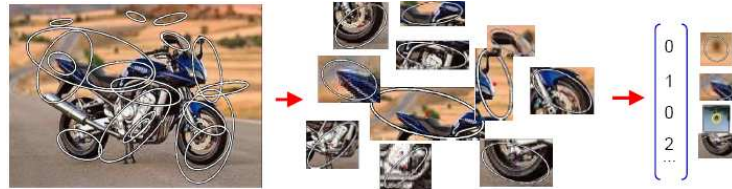


Figure 2-19: Bag-of-features model [31]. Affine invariant features are detected from the motorbike image. Spatial relations of features are discarded. The image is represented and classified with a global histogram of feature occurrence.



Figure 2-20: An example of a face image as a mixture of visual topics [62]. The face topic is shown in yellow, and background topics are shown in blue and cyan.

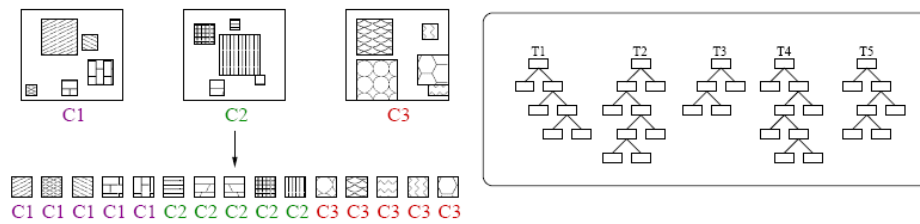


Figure 2-21: Random multi-scale subwindows [79] are extracted from three classes of objects. An ensemble of extremely randomized decision trees is learned and used for classification.

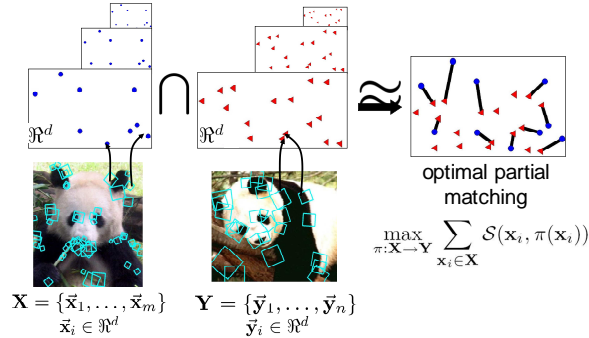


Figure 2-22: The pyramid matching kernel [48] intersects histogram pyramids formed over local features, approximating the optimal correspondences between features of the two images.

2.4 Summary

Generally speaking, most of the early work on object recognition focused on simplified scenes such as a composition of 3D objects in a uniform background. Methods based on representations of 3D models and multiple 2D views were shown to achieve good performance, mainly in applications where a single or several specific object instances were to be detected and recognized in a scene. The field of object recognition then gradually progressed into studying the more general problem of categorical recognition of object classes. Early work of categorical recognition targeted for distinguishing a limited set of object classes such as digits, cars, and faces. Recent developments in object categorization started to tackle the categorical recognition problem on much larger scales, *e.g.*, the 101 classes of natural objects in the dataset of Caltech-101 [37], where many appearance-based and geometry-based methods are demonstrated to steadily improve the state-of-the-art recognition performance.

This thesis works on the problem of object categorization in natural images. Many components in the proposed “recognition-through-decomposition-and-fusion” model, especially the components for matching objects by appearance, color, and shape in decomposed visual channels, are built upon the success of many prior art. The integration of multiple matching schemes is naturally enabled by the perceptual decomposition and recombination framework proposed in this thesis, and is demonstrated to be effective in building a better object categorization system.

Chapter 3

Low-level Image Measurements

The goal of this thesis is to develop a generic object categorization system that will automatically decompose the rich visual information in natural images, fully leverage each individual decomposed channel with suitable matching schemes, and adaptively combine the decomposed information to achieve maximal discriminability. The lowest level of this system determines how visual information contained in images should be measured and represented. This chapter first introduces a representation of contour and texture processes, which will act as a compact summary of salient visual information in images and will form the basis of visual information decomposition. Image features for the contour and texture processes, such as measurements of contourness and textureness, are then described.

3.1 Contour Process and Texture Process

This section gives formal definitions of contour process and texture process to represent and decompose visual information in images. We assume that images are defined over a finite lattice $\mathbf{P} = \{p_1, p_2 \dots p_N\}$ where $p = (i, j)$ denotes image sites or pixels on the lattice. It is further assumed that, with certain operators, a set of pixels of interest can be extracted, denoted as $\mathbf{POI} = \{p_1, p_2 \dots p_M\} \subseteq \mathbf{P}$. For each pixel $p \in \mathbf{POI}$ in the set of pixels of interest, a label c_p , that indicates whether the pixel is a contour pixel or not, can be assigned. Each label c_p is modeled as a discrete

random variable taking a value in $\{1, -1\}$, with 1 signaling that the pixel p is a contour pixel and -1 for non-contour. The set of labeled variables for all pixels of interest, $\mathbf{C} = \{c_p : p \in \mathbf{POI}\}$, is called the *contour label process*, or *contour process* for short. A *texture process* can be defined in the same way. For each pixel $p \in \mathbf{POI}$, a label t_p is assigned to indicate whether pixel p is a texture pixel or not. Each label t_p is also a discrete random variable taking a value in $\{1, -1\}$, with 1 for texture and -1 for non-texture, and the set of labeled variables $\mathbf{T} = \{t_p : p \in \mathbf{POI}\}$ is called the *texture process*. Figure 3-1 illustrates the definition of the contour and texture processes. The importance of using two individual processes instead of one single process to represent contour and texture will be demonstrated in Chapter 4. Given an image \mathbf{I} , the optimal labelings (\mathbf{C}, \mathbf{T}) of pixels in \mathbf{POI} , which maximizes the posteriori probability $P(\mathbf{C}, \mathbf{T}|\mathbf{I})$, can be estimated with probabilistic inferences. The set of optimal labelings for pixels of interest given an image \mathbf{I} naturally defines a decomposition of visual information in the \mathbf{POI} , with pixels with $c_p = 1$ in the contour channel and pixels with $t_p = 1$ in the texture channel. Once we have learned the decomposition, we can use the decomposed visual cues as a complementary basis for classification of objects, including learning which cues are more salient for distinguishing classes.

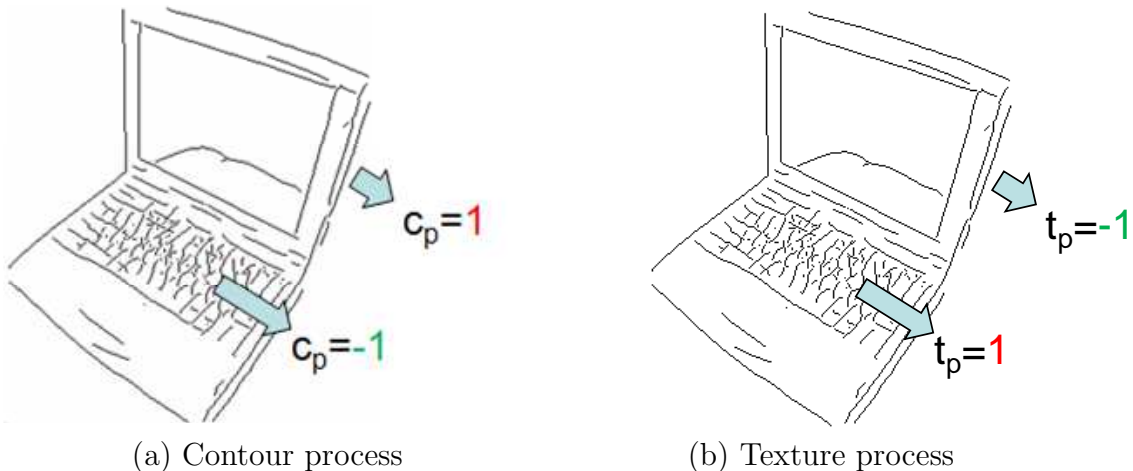


Figure 3-1: Illustration of definition of the contour and texture processes.

There are at least two alternative ways to define pixels of interest \mathbf{POI} for these

image processes. One way is to label every pixel in an image to be either contour or texture, i.e. $POI = P$. A second way is to first extract salient visual cues from an image and then focus on labeling the extracted salient pixels. In the latter case, edge pixels can be extracted first as visually salient information and the contour process and the texture process are defined only on the extracted edge pixels. Edges often receive special attention in early stages of computer vision, because sharp changes in image properties usually reflect important cues for perception. Focusing on edge pixels will also significantly reduce the amount of data to be processed, while preserving most of the relevant information. This thesis uses edges as the pixels of interest POI for decomposition.

Some flexibility also lies in how contour pixels are defined. One possibility is that only occluding contours are regarded as contour pixels. An alternative definition regards both occluding and internal contours as contour pixels. In the latter definition, depending on the scale, some internal edge pixels may be considered as texture flow or an internal contour. For instance, at a large scale, zebra stripes or soccer-ball patches have a repeating pattern and appear to be texture; at a smaller scale, edge pixels from stripes or patches are well-aligned and appear to be internal contours. This thesis adopts the latter definition.

3.2 Measuring Contourness and Textureness

To instantiate the posteriori probability $P(\mathbf{C}, \mathbf{T} | \mathbf{I})$ for contour and texture decomposition, observed image features need to be defined. The image features used in this thesis are measurements of contourness and textureness of edge pixels in POI .

3.2.1 Contourness Measurement and Edge Extraction

One choice of how to measure contourness of edge pixels is to adopt the widely used gradient magnitude. Gradient magnitude is typically computed from outputs of some linear differential filters which approximate continuous gradient operators in the discrete space of 2D lattices of images. As shown by Perona and Malik [91], these linear filters often only perform well on simple edges such as step edges or

steep ramps. However, it is well known that the projection of depth or orientation discontinuities in a physical scene results in image intensity edges which are not only step edges but are more typically a combination of steps, ramps, peaks and roofs [58]. Gradient magnitude measurement with linear filters ignores the composite nature of these edges, and in many cases results in systematic errors in detection, localization and magnitude computation [42, 86, 91]. As an improvement, many researchers, *e.g.*, Marrone and Owens [86], Perona and Malik [91], and Freeman and Adelson [42], have shown that the measurement and detection of edge points based on local energy are adequate to deal with most of the composite edges in images.

Local energy is usually computed with a quadrature filter bank. Two functions are said to be in quadrature when they are each other's Hilbert transform. The reasons for using quadrature filters are two-fold. First, the filter bank for computing orientation energy is typically designed to stem from two base filters: an even filter and an odd filter, and the entire filter bank is composed of a number of rotated versions of the base even/odd pair. One property of the Hilbert transform is that the Hilbert transform of an even function is an odd function and the Hilbert transform of an odd function is an even function. Hence a quadrature pair, created with an even or odd function, and its Hilbert transform are a handy way to generate a pair of base even and odd filters for the computation of orientation energy. Figure 3-2 shows 2D plots of one implementation of a base quadrature filter pair. The even filter in Figure 3-2(a) typically has extrema at even signal structures, such as delta peaks, roofs or lines. And the odd filter in Figure 3-2(b) generally assumes extrema at odd signal structures, such as step edge features.

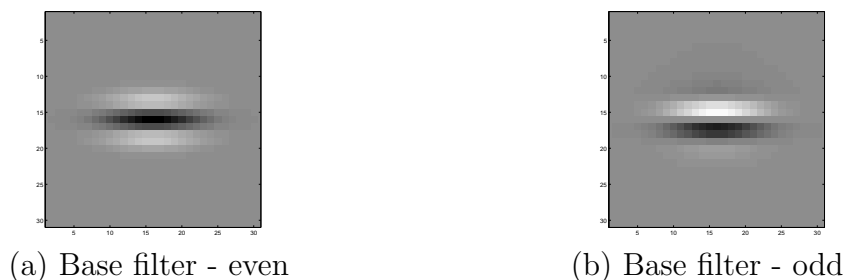


Figure 3-2: Quadrature pair used as base filters.

The second reason is that mathematically the filter outputs of a quadrature pair effectively define an *analytical signal*, with one output as the real part and the other output as the imaginary part of the corresponding analytic signal. The magnitude of the corresponding analytical signal is equivalent to the *amplitude envelope* of the original underlying signal, and the square of the magnitude corresponds to the local energy of the amplitude envelope [86]. Morrone and Owens [86] and Perona and Malik [91] showed that this local energy generally assumes maxima at many composite features such as even features, odd features and features in between. Hence the local energy derived from the outputs of a quadrature filter pair can be used as a strong cue and appropriate measurement of composite edge features, *e.g.*, step edges, peaks and roofs. In [42, 80], this local energy is termed “*orientation energy*” and is computed as the sum of squares of a quadrature filter’s responses:

$$OE_\theta = (I * f_\theta^e)^2 + (I * f_\theta^o)^2 \quad (3.1)$$

where OE_θ is the orientation energy along direction θ , f_0^e and f_0^o are the base quadrature pair and f_θ^e is the rotated version of f_0^e in the direction of θ . Similarly for f_θ^o . I is the image.

The quadrature filter bank used in this thesis are the even and odd pairs as in [78, 80]. That is, the base symmetric filter is the second derivative of an elongated Gaussian, with $\sigma = 1.5$ in the x direction and 3σ in the y direction, and the base odd-symmetric filter is its Hilbert transform. Their 2D plots are shown in Figure 3-2. The entire filter bank consists of 8 rotated versions of the base even/odd pair, as in Figure 3-3. This filter bank is fixed in terms of spatial scale. It could be easily extended to be scale invariant, by using a series of multiple scales and the selection scheme as in [111], or by using the detected spatial scales of features as in [76]. However, in many object recognition datasets such as Caltech-101 [37], objects are mostly imaged under canonical poses and don’t have large scale variations. In these cases, the filter bank can be fixed in spatial scale.

For color images, color channels are added in addition to brightness for computing

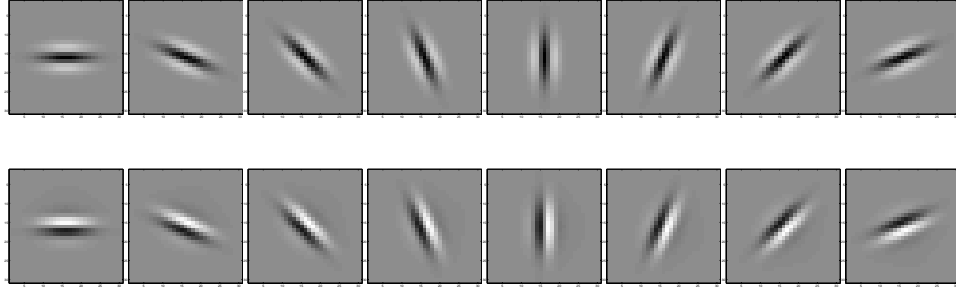


Figure 3-3: Quadrature filter bank.

orientation energy. Color information helps to disambiguate contours where gray-values are similar but colors are different. RGB images are first converted to CIELAB space. For each of the L^* , a^* and b^* channels, orientation energy is computed as in Equation 3.1. One way to integrate these orientation energy into the low-level measurements is to use the orientation energy of the L^* , a^* and b^* channels separately. As an alternative, in practice, adding the orientation energies of the three channels is observed to be able to effectively capture most of the salient contours. Moreover, adding the three channels gives an integrated orientation energy representation, which reduces the complexity of the proposed coupled Conditional Random Field model in Chapter 4. In this thesis, orientation energies of the L^* , a^* and b^* channels are added together as the final orientation energy. For gray-scale images, only the original brightness is used.

For each pixel p , the largest of the 8 energy terms of OE_θ 's is kept as the orientation energy of pixel p , and the orientation of pixel p is determined by the corresponding θ . As an example, the computed orientation energies for a laptop image is shown in Figure 3-4.

Postprocessing for Computing Orientation Energy

Due to large illumination variations in natural images, many contours of commensurate visual saliency always have quite different orientation energies within one image. Reducing their discrepancies is helpful to facilitate subsequent stages such as edge extraction and computation of contour probability based on orientation energy.

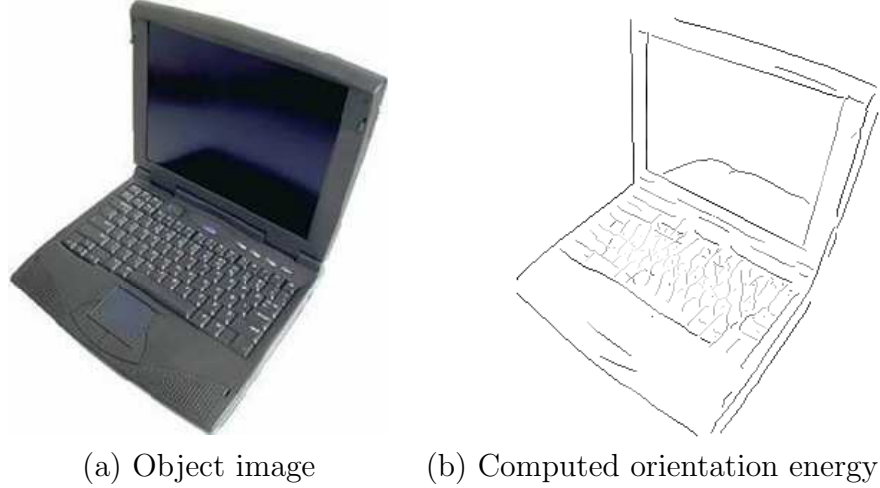


Figure 3-4: An example of computed orientation energy.

One possible way is to use the local contrast normalization procedure proposed by Freeman *et al.* [41]. In this thesis a power-law on the orientation energy is used:

$$rOE_{\theta} = \left\{ \sum_c [(I * f_{\theta}^c)^2 + (I * f_{\theta}^o)^2] \right\}^{\alpha} \quad (3.2)$$

where rOE_{θ} is the rectified orientation energy in direction θ . α is 0.5 in this thesis. c stands for the channels used for computing orientation energy. For color images, $c = \{L^*, a^*, b^*\}$; for gray-scale images, c is a singleton channel of original grayvalue.

Edge Points Extraction

As discussed in Section 3.1, the pixels of interest **POI** used in this thesis are edge pixels of images. To robustly extract edge pixels in images, a Canny's hysteresis thresholding [25] is applied to the orientation energy image to extract edge points, as in [42]. Both the lower and higher thresholds of the hysteresis thresholding are set to be relatively small in order to minimize misses at true edges with low contrast. Further rectification will be postponed until inferences on the coupled Conditional Random Field in Chapter 4. Figure 3-5 shows an example of the extracted edges of a laptop image.

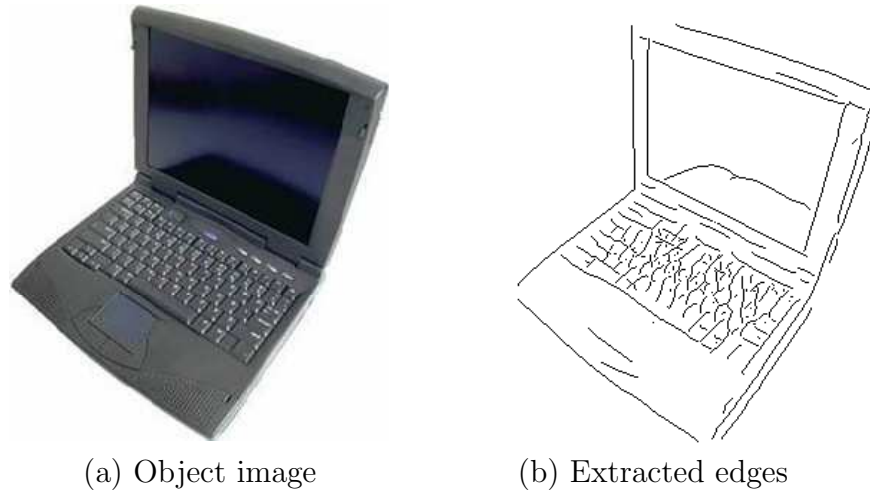


Figure 3-5: An example of edge extraction.

3.2.2 Textureness Measurement

Texture is often referred as a perceptually coherent and distinctive physical composition or structure of certain basic elements, especially with respect to the size, shape, and arrangement of its parts. As an important aspect of human perception, texture has been studied for decades. Properties such as coarseness, anisotropy, homogeneity and entropy have been examined as perceptual measurements for texture. In [63], Julesz proposed the concept of *texton*, which is an atomic element for texture perception. Since the seminal work of Julesz [63] and Leung and Malik [72], *texton* has become the standard tool for texture analysis. This thesis uses a textureness measurement derived from distributions of textons, which was first proposed by Martin *et al.* and named as “*texture gradient*” in [80]. Martin *et al.* originally developed the concept of *texture gradient* to learn isolated mapping functions to compute probabilities of boundaries. In this thesis, texture gradient is computed for measuring textureness of points, which will be used together with contourness measurements in the coupled Conditional Random Field model in Chapter 4 to decompose contour and texture in images.

The concept of *texture gradient* is a measurement of how different the distribution of texture on one side of a pixel is relative to that of the other side of the pixel. When

the appearance of the texture on one side of a pixel is perceptually different than the texture's appearance on the other side, the *texture gradient* will be large and the pixel has a large probability to lie on some perceptually salient contour delineated by different textures. The texture gradient is a complementary measurement to, and in some cases better measurement than, the contourness measurement of orientation energy.

To compare the texture appearance of the two 'sides' of a pixel, the local orientation of the pixel should be first known. In this thesis, the local orientation of a pixel is straightforwardly set as the pixel's orientation computed by orientation energy in Section 3.2.1. The next step to compute texture gradient is to define the distribution of textures on both sides, which are described as histograms of textons. To this end, texture features, which are extracted by a filter bank, are first computed for pixels of interest. The filter bank used in this thesis consists of the 8 rotated quadrature pairs used in measuring orientation energy in Section 3.2.1, plus 3 Gaussians and 3 center-surround filters of Difference of Gaussians at 3 different scales of $\sigma = \{1.5, 2, 3\}$. Figure 3-6 shows these filters in the filter bank. A 22-dimensional vector is formed as the extracted feature for each pixel of interest by concatenating the filter bank's responses at the pixel. With the extracted texture features, vocabularies of textons can be built. There are two alternative ways to build textons - globally or locally. A global texton vocabulary can be built by clustering all texture features from a set of training images and this global texton vocabulary is used for all images. Local texton

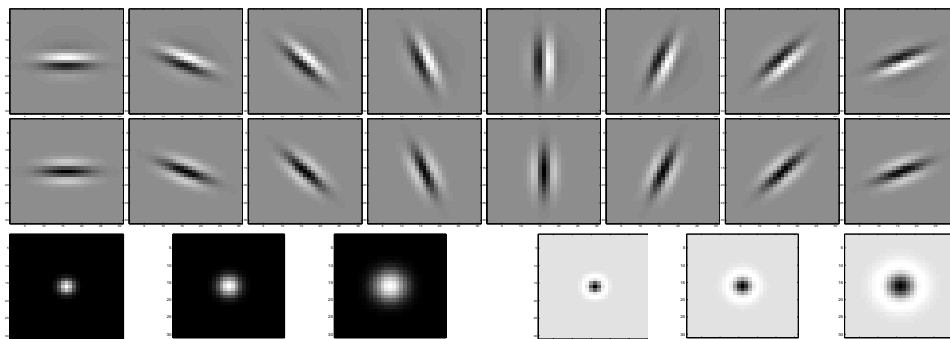


Figure 3-6: Texton filter bank.

vocabularies are image-specific, which means the texture features from one image can be clustered and form a texton vocabulary for this image only. For one specific image, both the global texton vocabulary and the local texton vocabulary of this image can be used to measure the texture gradient. In [80] and also in our experiments, the two approaches achieve practically the same measurements of texture. This thesis uses local texton vocabularies for computing texture gradient. Texture features from images are clustered into 50 textons for each of the images.

With the local orientation of a pixel and the texton vocabulary defined, texture gradient can now be introduced. Similar to [78], for each edge pixel, a 20-pixel wide circular region around the pixel is extracted and cut in three parts: a 10-pixel wide center strip D_0 along the orientation of the edge pixel of interest, and D_+ and D_- which are the pixels to the left and right of D_0 respectively, as illustrated by Figure 3-7. Next, D_0 is first merged with D_- and $D_0 \cup D_-$ is compared with D_+ . A χ^2 -distance is computed between the histograms of textons in D_+ and $D_0 \cup D_-$. Similarly a χ^2 -distance is computed between the histograms of textons in D_- and $D_0 \cup D_+$. The larger of the two distances is kept as the texture gradient for the edge pixel. Figure 3-8 shows a laptop image and its corresponding texture gradient. Texture gradient is used as the texture measurement in the proposed coupled Conditional Random Field. The smaller the texture gradient, the greater the texture. In this thesis, we only model textures on edge pixels. Homogeneous regions will be added in separate color channels.



Figure 3-7: Illustration of computation of texture gradient.

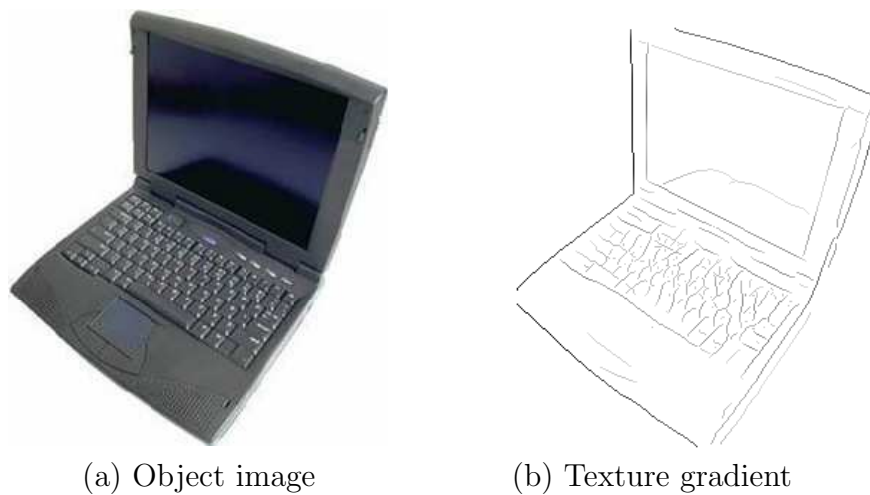


Figure 3-8: An example of computed texture gradient.

3.3 Summary

As the lowest level of the proposed computational model, the contour process and texture process in images are introduced in this chapter. Typically, contour and texture processes are defined on a set of pixels of interest such as edge points, in order to focus attention on salient structures and improve computational efficiency while preserving most of the relevant information. To extract edge points, orientation energy computed by a quadrature filter bank is used. The computed orientation energy also determines the orientation of edge points and acts as the contourness measurement of extracted edge points. Textureness of edge points are measured by texture gradient.

As can be seen in the computed orientation energy in Figure 3-4, and the texture gradient in Figure 3-8, some pixels from the keyboard texture of the laptop have large orientation energy and some texture gradients on object boundaries are small. Simple thresholding on the contourness and textureness measurements will not separate contour and texture pixels. Moreover, in many natural images, contour and texture pixels are often close to and intertwined with each other, which makes decomposition of contour and texture even harder. In the next chapter, a coupled Conditional Random Field model is introduced to model and decompose the contour and texture processes.

Chapter 4

Learning Coupled Conditional Random Field for Image Decomposition

4.1 Motivation

Understanding low-level visual cues such as contour and texture in natural images is of great importance. Behavioral and physiological evidence suggests that human observers perceive the visual information of contour and texture in functionally separable dimensions and recombine them in an integrative stage to recognize objects [1, 6, 26, 36, 59, 116]. It is desirable to design object categorization systems to simulate this perceptual behavior of human observers. Decomposition of different visual stimuli would form better low-level and mid-level image models, and enable better high-level modeling which can fully leverage decomposed perceptual cues.

Generally speaking, contours are more salient in high contrast regions of brightness, colors and/or texture and typically form continuous curves along occluding boundaries and internal patterns; textures are perceived as certain compositions of elements, with perceptually notable consistency of appearance and/or geometric layout. While these observations are true in general, in most natural images, the percep-

tion of contours and textures is more complicated. Point-wise measurements are not always readily distinct enough for discriminating contours and textures, for instance, some contours might have low contrast whereas elements of textures could have quite high local contrast. What makes the problem even harder is that contours and textures are often intertwined together in images. Some contours might be adjacent to or even buried in textures. Apparently a simple thresholding (regular or hysteresis) or function mapping scheme won't suffice. For example, the widely adopted Canny edge detector uses hysteresis thresholding and non-maxima suppression to extract salient edge points. However, it won't discriminate edge points from contour and texture processes which have different characteristics, and won't work well to decompose the two processes where they mingle. This leads to an important observation about the perception of contour and texture processes: contour and texture are not independent of each other in images, and a model of contour and texture decomposition will have to address this dependency.

In this thesis, a coupled Conditional Random Field model is proposed for modeling contour and texture processes in images. The following sections first give a general description of the importance of the coupled Conditional Random Field model, then introduce the mathematical form of the proposed model and its learning and inference components. The model is trained with a set of labeled images. Analysis of the learned model shows that the proposed coupled Conditional Random Field is able to capture many distinct characteristics of contour and texture channels, where a single-layer Conditional Random Field model has to make unacceptable compromises. Evaluation on another set of images shows that, for contour and texture decomposition, the proposed coupled Conditional Random Field model outperforms a single-layer Conditional Random Field.

4.2 Importance of Learning Coupled Conditional Random Field Model

Decomposition of contour and texture processes entails a labeling of edge points. Markovian models are widely adopted for a variety of labeling tasks. Just to name a few, Geman and Graffigne [47] used a Markov Random Field model for texture segmentation; Freeman *et al.* [41] developed Markov Network models for super-resolution, shading and reflectance estimation and motion estimation; He *et al.* [54] and Kumar and Hebert [66] used Conditional Random Field for image site labeling. Compared with generative models such as Markov Random Field models, a Conditional Random Field [67] focuses resources on modeling posterior distributions as a Gibbs field. Without the strong assumption of conditional independence of observations, Conditional Random Field allows arbitrary dependent structures between observations. Considering the complex interactive nature of contour and texture processes, a Conditional Random Field model for the joint posterior of contours and textures given an image is suitable.

A popular way of labeling image processes is to use a single layer of a random field grid. Such a model for labeling edge process, with one node for each edge point, is shown in Figure 4-1. Each node e_i in Figure 4-1 represents an edge point. Each edge point e_i is assigned a label of +1 or -1, where +1 means the edge point is a contour point whereas -1 signals a texture point. The underlying idea is that labels for each edge point are influenced by nearby labels as well as local measurements, and thus local context helps propagate labels throughout a region.

However, this single-layer random field is limited in modeling power. The two processes, contour and texture, could have disparate characteristics and dynamics in their respective inter-point interactions. One distinction lies in the angular alignment of points. In the contour process, compatible contour points are mostly aligned in local neighborhoods. For the texture process, edge points are seldomly well-aligned and often random (recall that locally well-aligned patterns such as zebra stripes are defined as internal contours, part of the contour process.) Hence, it is logical to

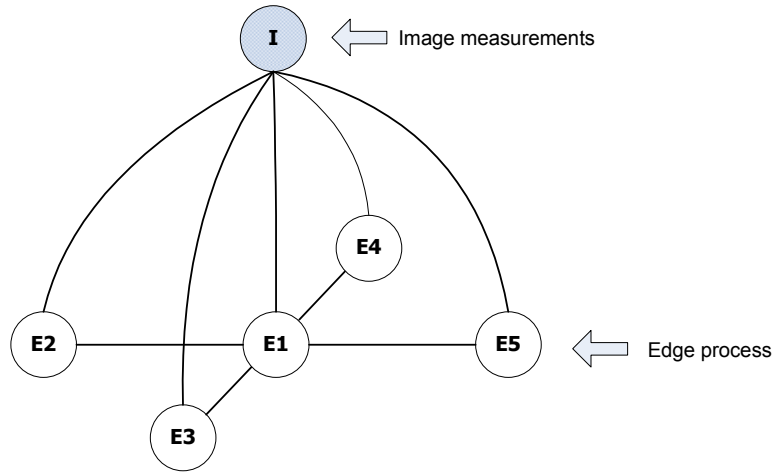


Figure 4-1: A simple single-layer Conditional Random Field model for both contour and texture processes.

postulate that contour points are compatible only when they are locally continuous and aligned, while the compatibility of texture points could allow a random layout. This means the compatibility functions of the two processes will exhibit disparate dependencies on an angular alignment parameter. Other different dynamics may also exist in the measurements of coarseness, anisotropy, homogeneity and entropy.

Under these situations, using single-layer random field models inevitably has to introduce a trade-off between distinct dynamics of different processes. To accommodate different characteristics of interactions, the compatibility function in a one-layer model will be forced to compromise between the two otherwise distinct compatibility functions of different processes. A better model is to explicitly capture different dynamics of processes, with more than one layer of random field grids. In the proposed model, one grid layer is used for a contour process and a separate grid is used for a texture process. The dependency between the two processes is modeled with coupling links between the layers. To reduce the complexity, each node in one layer is only coupled with the same node in the other layer. This leads to the proposed coupled Conditional Random Field (cCRF) model, shown in Figure 4-2. A formal definition

of the coupled Conditional Random Field is given in Table 4.1. The importance of using coupled rather than single layer Conditional Random Fields to address different image processes will become more evident in the experimental results (Section 4.5.2).

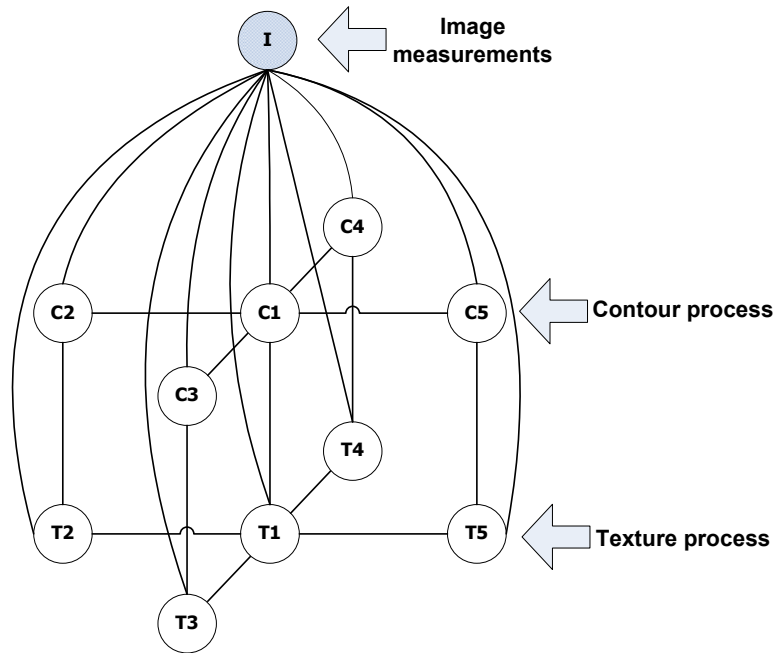


Figure 4-2: Coupled Conditional Random Field for modeling contour and texture processes.

Coupled Conditional Random Field for contour and texture processes of an image: $P(\mathbf{C}, \mathbf{T} | \mathbf{I}, \Theta)$, where \mathbf{C} represents the contour process, \mathbf{T} represents the texture process and \mathbf{I} is an image. Θ is the set of parameters of coupled Conditional Random Field. The coupled Conditional Random Field is only defined on extracted edge pixels.

Table 4.1: Definition of the coupled Conditional Random Field for contour and texture processes of an image.

4.3 Parametrization of Coupled Conditional Random Field

The proposed functional forms of the coupled Conditional Random Field model in Figure 4-2 are shown in Table 4.2. The pixels of interest, which are edge pixels in this thesis, are indexed by the variable i . c_i represents the labeling variable in the contour layer, or equivalently, the contour process. t_i represents the labeling variable in the texture layer. Five image measurements are used in the current work. Contourness cm_i and textureness tm_i are used for local evidence functions, which have a form of logistic regression. This discriminative form of local evidence was originally proposed in Discriminative Random Field [66]. Unlike the log-linear compatibility in [66], the proposed coupled Conditional Random Field uses a form of logistic regression for compatibility functions. The corresponding measurements used are: (1) $\delta\theta_{ij}$, angular difference between the orientation of i and the line joining i with a neighboring pixel j , where the orientation of i is given by the low-level measurement as described in Section 3.2.1; (2) $\delta cm_{ij} = |cm_i - cm_j|$, absolute contourness difference between i and j ; (3) $\delta tm_{ij} = |tm_i - tm_j|$, absolute textureness difference between i and j . The compatibility functions of the two processes will have potentially different parameters, capturing the distinct interaction dynamics of the two processes stated in Section 4.2. For instance, $\Psi_c(c_i, c_j)$ could be large (which means c_i and c_j are compatible) when $\delta\theta_{ij}$ is small (*i.e.* pixel i and j are aligned) and small when $\delta\theta_{ij}$ is large (edges not aligned), while for $\Psi_t(t_i, t_j)$ of the texture process, its value won't change a lot when $\delta\theta_{ij}$ changes. This essentially captures the distinct dynamics of how contour and texture processes respond to edge alignment.

In both processes, the compatibility between a negative labeling and a neighboring negative labeling, *e.g.*, for the labeling pair $(c_i, c_j) = (-1, -1)$, is fixed to 0.5. The reason is that in the proposed coupled Conditional Random Field the interactive dynamics of negative-to-negative labelings in one layer are already represented in the positive-to-positive compatibility in the opposite layer, and are coupled into the current layer through the coupling links. On one hand, fixing negative-to-negative

<ul style="list-style-type: none"> • Variables for pixel i <ul style="list-style-type: none"> c_i: labeling variable in the contour layer: <ul style="list-style-type: none"> $c_i = 1$: contour pixel; $c_i = -1$: non-contour pixel. t_i: labeling variable in the texture layer: <ul style="list-style-type: none"> $t_i = 1$: texture pixel; $t_i = -1$: non-texture pixel. cm_i: contourness measurement. tm_i: textureness measurement. $\delta\theta_{ij}$: angle between orientation of i and the line joining i and a neighboring pixel j. δcm_{ij}: absolute difference between the contourness of i and j. δtm_{ij}: absolute difference between the textureness of i and j. • Evidence function $\Phi_c(c_i I)$ $\Phi_c(c_i I) = \frac{1}{1 + e^{-c_i(\alpha_0 + \alpha_1 cm_i + \alpha_2 tm_i)}}$ • Compatibility function $\Psi_c(c_i, c_j I)$ $\Psi_c(c_i, c_j I) = \begin{cases} 0.5, & \text{if } (c_i, c_j) = (-1, -1) \\ A, & \text{otherwise} \end{cases}$ <p>where $A = \frac{1}{1 + e^{-c_i c_j(\tau_0 + \tau_1 \delta\theta_{ij} + \tau_2 \delta cm_{ij} + \tau_3 \delta tm_{ij})}}$</p> • Evidence function $\Phi_t(t_i I)$ $\Phi_t(t_i I) = \frac{1}{1 + e^{-t_i(\beta_0 + \beta_1 cm_i + \beta_2 tm_i)}}$ • Compatibility function $\Psi_t(t_i, t_j I)$ $\Psi_t(t_i, t_j I) = \begin{cases} 0.5, & \text{if } (t_i, t_j) = (-1, -1) \\ A, & \text{otherwise} \end{cases}$ <p>where $A = \frac{1}{1 + e^{-t_i t_j(\gamma_0 + \gamma_1 \delta\theta_{ij} + \gamma_2 \delta cm_{ij} + \gamma_3 \delta tm_{ij})}}$</p> • Compatibility function $\Psi_{ct}(c_i, t_i I)$ $\Psi_{ct}(c_i, t_i I) = \begin{cases} 0 & \text{if } c_i = t_i \\ 1 & \text{if } c_i \neq t_i \end{cases}$ <p><i>i.e.</i>, contour and texture processes are mutually exclusive.</p>

Table 4.2: Evidence and compatibility functions of the proposed coupled Conditional Random Field. See text for details.

compatibility removes modeling redundancy. On the other hand, this property is the clear distinction of the coupled Conditional Random Field compared with a single-layer Conditional Random Field: each layer of random field grid only focuses on representing the interactive dynamics within one type of process, without modeling the negative-to-negative compatibility which will have to introduce compromises to model both types of interactive dynamics. The negative-to-negative compatibility is empirically set to 0.5 in this thesis. In principle, this compatibility can be learned with a training set. The compatibility matrix $\Psi_{ct}(c_i, t_i|I)$ for the coupling links is fixed to make the two processes mutually exclusive. Note, however, this could be extended to allow non-mutually-exclusive labeling.

For clarity, the graphical model in Figure 4-2 shows a coupled Conditional Random Field with a 4-neighborhood system for each of the contour and texture processes. In practice, models defined on higher order neighborhood systems, capturing more information from neighboring pixels, are used.

4.4 Learning and Inference of Coupled Conditional Random Field

4.4.1 Model Learning with Maximum Pseudolikelihood

Assume only up to pairwise clique potentials are nonzero. With the functional forms in Table 4.2, the posterior of the coupled Conditional Random Field is given by the following factorized form:

$$P(C, T|I, \Theta) = \frac{1}{Z} \left\{ \prod_i \Phi_c(c_i|I) \Phi_t(t_i|I) \Psi_{ct}(c_i, t_i|I) \right\} \cdot \left\{ \prod_{(i,j) \in C_{edge}} \Psi_c(c_i, c_j|I) \right\} \cdot \left\{ \prod_{(i,j) \in T_{edge}} \Psi_t(t_i, t_j|I) \right\} \quad (4.1)$$

where i, j are indexes of edge pixels, C_{edge} indicates the set of inter-node links in the contour layer and T_{edge} for the texture layer. Z is the normalization constant (partition function). Θ is the set of parameters of the coupled Conditional Random Field.

The log-likelihood function is written as:

$$\mathcal{L}(\Theta) = \log \prod_{m=1}^M P(C^m, T^m | I^m, \Theta) \quad (4.2)$$

where M is the number of training images, and $m=1\dots M$ is the index of training samples. I^m represents m th training image. C^m and T^m are the contour and texture processes of I^m respectively.

In principle, parameters can be learned with a maximum-likelihood approach, *i.e.*,

$$\Theta_{ML}^* = \operatorname{argmax}_{\Theta} \mathcal{L}(\Theta) = \operatorname{argmax}_{\Theta} \log \prod_{m=1}^M P(C^m, T^m | I^m, \Theta) \quad (4.3)$$

Maximum-likelihood learning is complicated by the partition function Z . A full Maximum-likelihood learning process involves the estimation of feature expectations under model distribution, which incurs a summation over the entire set of possible labeling. Also the evaluation of Z is typically NP-hard. Exact maximum-likelihood in this case is intractable, thus the model learning has to resort to approximation techniques. There are many ways to approximate the exact maximum-likelihood learning. For homogeneous random fields such as the coupled Conditional Random Field proposed in this thesis, Besag [11] has devised an ingenious alternative to the full maximum-likelihood, named “maximum pseudolikelihood”.

The maximum pseudolikelihood approach simplifies model learning by approximating the true likelihood with a factorization of the local conditional likelihoods [11], *i.e.*

$$\Theta_{ML}^* \simeq \operatorname{argmax}_{\Theta} \log \prod_{m=1}^M \prod_i P(c_i^m, t_i^m | C_{N_i}^m, T_{N_i}^m, I^m, \Theta) \quad (4.4)$$

where i is the index of edge pixels and N_i represents the neighborhood of i . c_i^m and t_i^m are the contour and texture labels for pixel i . $C_{N_i}^m$ is the contour labeling of edge pixels in the neighborhood of i for the m th training sample; similarly for $T_{N_i}^m$.

Each of the local conditional likelihoods has the following form:

$$P(c_i, t_i | C_{N_i}, T_{N_i}, I, \Theta) = \frac{P(c_i, t_i, C_{N_i}, T_{N_i} | I, \Theta)}{Z_i} \quad (4.5)$$

$$Z_i = \sum_{c_i \in \{+1, -1\}, t_i \in \{+1, -1\}} P(c_i, t_i, C_{N_i}, T_{N_i} | I, \Theta) \quad (4.6)$$

where each $P(c_i, t_i, C_{N_i}, T_{N_i} | I, \Theta)$ has the same form as in Equation 4.1 with only terms for c_i, t_i and their immediate neighbors.

There is still a partition function for each of the local conditional likelihoods. However, now each of these partition functions only sums over 4 possible combinations of labels, *i.e.* the range of labeling (c_i, t_i) , making the computation tractable and potentially fit for parallel processing.

The choice of pseudo-likelihood approximation is due to the consideration of simplicity on one hand. Another nice property of pseudo-likelihood is its consistency of estimates in the large lattice limit [47], that is to say, for the “large graph” limit, the estimate by maximum pseudo-likelihood converges to the true parameters with probability one as the number of nodes on the lattice goes to infinity. Geman and Graffigne prove this consistency property based on learning from one large texture image. This property can be readily extended to multiple training images where the number of training edge points is sufficiently large.

4.4.2 Derivation of Maximum Pseudolikelihood Learning of Coupled Conditional Random Field

The learning by maximum pseudolikelihood (Equation 4.4) is a highly non-linear optimization procedure. Many non-linear optimization methods can be employed to estimate the optimal parameters. A typical practice is to use gradient-based algorithms, such as Gradient Ascent, which involves computation of the log-pseudolikelihood function's partial derivatives to the set of parameters. The following section gives the mathematical derivation of these derivatives.

Each of the local conditional probabilities in Equation 4.5 expands in the following form:

$$P(c_i^m, t_i^m | C_{N_i}^m, T_{N_i}^m, I^m, \Theta) = \frac{[\Phi_c(c_i^m | I^m) \Phi_t(t_i^m | I^m) \Psi_{ct}(c_i^m, t_i^m | I^m) \prod_{(i,j) \in C_{edge}} \Psi_c(c_i^m, c_j^m | I^m) \prod_{(i,j) \in T_{edge}} \Psi_t(t_i^m, t_j^m | I^m)]}{Z_i} \quad (4.7)$$

$$Z_i = \sum_{c_i \in \{+1, -1\}, t_i \in \{+1, -1\}} \Phi_c(c_i | I^m) \Phi_t(t_i | I^m) \Psi_{ct}(c_i, t_i | I^m) \cdot \prod_{(i,j) \in C_{edge}} \Psi_c(c_i, c_j^m | I^m) \prod_{(i,j) \in T_{edge}} \Psi_t(t_i, t_j^m | I^m) \quad (4.8)$$

Take the parameter α_0 in $\Phi_c(c_i|I)$ as an example. For each image site i ,

$$\begin{aligned}
& \frac{\partial \log P(c_i^m, t_i^m | C_{N_i}^m, T_{N_i}^m, I^m, \Theta)}{\partial \alpha_0} \\
= & \partial \log \left[\Phi_c(c_i^m | I^m) \Phi_t(t_i^m | I^m) \Psi_{ct}(c_i^m, t_i^m | I^m) \prod_{(i,j) \in C_{edge}} \Psi_c(c_i^m, c_j^m | I^m) \cdot \right. \\
& \left. \prod_{(i,j) \in T_{edge}} \Psi_t(t_i^m, t_j^m | I^m) \right] / \partial \alpha_0 - \\
& \partial \log \left[\sum_{c_i, t_i} \Phi_c(c_i | I^m) \Phi_t(t_i | I^m) \Psi_{ct}(c_i, t_i | I^m) \prod_{(i,j) \in C_{edge}} \Psi_c(c_i, c_j^m | I^m) \cdot \right. \\
& \left. \prod_{(i,j) \in T_{edge}} \Psi_t(t_i, t_j^m | I^m) \right] / \partial \alpha_0 \\
= & \frac{\partial \log \Phi_c(c_i^m | I^m)}{\partial \alpha_0} - \\
& \left[\sum_{c_i, t_i} \frac{\partial \Phi_c(c_i | I^m)}{\partial \alpha_0} \Phi_t(t_i | I^m) \Psi_{ct}(c_i, t_i | I^m) \prod_{(i,j) \in C_{edge}} \Psi_c(c_i, c_j^m | I^m) \prod_{(i,j) \in T_{edge}} \Psi_t(t_i, t_j^m | I^m) \right] \\
& / \left[\sum_{c_i, t_i} \Phi_c(c_i | I^m) \Phi_t(t_i | I^m) \Psi_{ct}(c_i, t_i | I^m) \prod_{(i,j) \in C_{edge}} \Psi_c(c_i, c_j^m | I^m) \prod_{(i,j) \in T_{edge}} \Psi_t(t_i, t_j^m | I^m) \right] \\
= & \frac{\partial \log \Phi_c(c_i^m | I^m)}{\partial \alpha_0} - \\
& \left[\sum_{c_i, t_i} \frac{\partial \log \Phi_c(c_i | I^m)}{\partial \alpha_0} \Phi_c(c_i | I^m) \Phi_t(t_i | I^m) \Psi_{ct}(c_i, t_i | I^m) \cdot \right. \\
& \left. \prod_{(i,j) \in C_{edge}} \Psi_c(c_i, c_j^m | I^m) \prod_{(i,j) \in T_{edge}} \Psi_t(t_i, t_j^m | I^m) \right] \\
& / \left[\sum_{c_i, t_i} \Phi_c(c_i | I^m) \Phi_t(t_i | I^m) \Psi_{ct}(c_i, t_i | I^m) \prod_{(i,j) \in C_{edge}} \Psi_c(c_i, c_j^m | I^m) \prod_{(i,j) \in T_{edge}} \Psi_t(t_i, t_j^m | I^m) \right] \\
= & \frac{\partial \log \Phi_c(c_i^m | I^m)}{\partial \alpha_0} - \sum_{c_i \in \{+1, -1\}, t_i \in \{+1, -1\}} \frac{\partial \log \Phi_c(c_i | I^m)}{\partial \alpha_0} P(c_i, t_i | C_{N_i}^m, T_{N_i}^m, I^m, \Theta)
\end{aligned} \tag{4.9}$$

The $\log \Phi_c(c_i|I)$ terms in Equation 4.9 is:

$$\log \Phi_c(c_i|I) = \log \frac{1}{1 + e^{-c_i(\alpha_0 + \alpha_1 c m_i + \alpha_2 t m_i)}}$$

So:

$$\begin{aligned} \frac{\partial \log \Phi_c(c_i|I)}{\partial \alpha_0} &= - \frac{\partial \log[1 + e^{-c_i(\alpha_0 + \alpha_1 c m_i + \alpha_2 t m_i)}]}{\partial \alpha_0} \\ &= - \frac{-c_i e^{-c_i(\alpha_0 + \alpha_1 c m_i + \alpha_2 t m_i)}}{1 + e^{-c_i(\alpha_0 + \alpha_1 c m_i + \alpha_2 t m_i)}} \\ &= c_i \left[1 - \frac{1}{1 + e^{-c_i(\alpha_0 + \alpha_1 c m_i + \alpha_2 t m_i)}} \right] \\ &= c_i [1 - \Phi_c(c_i|I)] \end{aligned} \tag{4.10}$$

Putting Equation 4.9 and 4.10 into the log-pseudolikelihood function's partial derivative with respect to α_0 :

$$\begin{aligned} & \frac{\partial \log \prod_{m=1}^M \prod_i P(c_i^m, t_i^m | C_{N_i}^m, T_{N_i}^m, I^m, \Theta)}{\partial \alpha_0} \\ &= \frac{\sum_{m=1}^M \sum_i \partial \log P(c_i^m, t_i^m | C_{N_i}^m, T_{N_i}^m, I^m, \Theta)}{\partial \alpha_0} \\ &= \sum_{m=1}^M \sum_i \left\{ \frac{\partial \log \Phi_c(c_i^m | I^m)}{\partial \alpha_0} - \sum_{(c_i, t_i)} \frac{\partial \log \Phi_c(c_i | I)}{\partial \alpha_0} P(c_i, t_i | C_{N_i}^m, T_{N_i}^m, I^m, \Theta) \right\} \\ &= \sum_{m=1}^M \sum_i \left\{ c_i^m [1 - \Phi_c(c_i^m | I^m)] - \sum_{(c_i, t_i)} c_i [1 - \Phi_c(c_i | I^m)] P(c_i, t_i | C_{N_i}^m, T_{N_i}^m, I^m, \Theta) \right\} \end{aligned} \tag{4.11}$$

The first item in Equation 4.11 is the empirical expectation of the term $(c_i [1 - \Phi_c(c_i|I)])$ (which is often referred as “the feature term” in learning a Markov Random Field) in the training set, and the second item in Equation 4.11 is the expectation of the feature under the model distribution given as the factorization form of Equation 4.4

and 4.5. When the non-linear optimization converges, that is, the partial derivative in Equation 4.11 is equal to zero, the feature $(c_i [1 - \Phi_c(c_i|I)])$'s empirical expectation is equal to the model expectation on the training set. Model learning behaves in the way of changing the model's parameter to maximally align the parameter's corresponding feature to the empirical mean that is observed in the training set.

Similarly for derivatives with respect to α_1 and α_2 ,

$$\frac{\partial \log \Phi_c(c_i|I)}{\partial \alpha_1} = c_i c m_i [1 - \Phi_c(c_i|I)]$$

$$\frac{\partial \log \Phi_c(c_i|I)}{\partial \alpha_2} = c_i t m_i [1 - \Phi_c(c_i|I)]$$

And,

$$\frac{\partial \log \prod_{m=1}^M \prod_i P(c_i^m, t_i^m | C_{N_i}^m, T_{N_i}^m, I^m, \Theta)}{\partial \alpha_1}$$

$$= \sum_{m=1}^M \sum_i \left\{ c_i^m c m_i^m [1 - \Phi_c(c_i^m | I^m)] - \sum_{(c_i, t_i)} c_i c m_i^m [1 - \Phi_c(c_i | I^m)] P(c_i, t_i | C_{N_i}^m, T_{N_i}^m, I^m, \Theta) \right\}$$

$$\frac{\partial \log \prod_{m=1}^M \prod_i P(c_i^m, t_i^m | C_{N_i}^m, T_{N_i}^m, I^m, \Theta)}{\partial \alpha_2}$$

$$= \sum_{m=1}^M \sum_i \left\{ c_i^m t m_i^m [1 - \Phi_c(c_i^m | I^m)] - \sum_{(c_i, t_i)} c_i t m_i^m [1 - \Phi_c(c_i | I^m)] P(c_i, t_i | C_{N_i}^m, T_{N_i}^m, I^m, \Theta) \right\}$$

The log-pseudolikelihood function's derivatives with respect to other parameters can be derived in the same manner, considering compatibility functions and evidence functions of the coupled Conditional Random Field model all have the form of logistic regression.

4.4.3 Model Learning with Tempered Maximum Pseudolikelihood

The proposed coupled Conditional Random Field is a complex image model. In practice, the learning of the complex coupled Conditional Random Field model is prone to over-fitting. To avoid over-fitting, a tempered maximum pseudolikelihood is used for learning the parameters of the coupled Conditional Random Field. Tempered maximum likelihood technique was first proposed by Hofmann [55] in tempered EM while learning the parameters of the probabilistic Latent Semantic Analysis model to improve generalization capability.

In each step, instead of maximizing the original pseudolikelihood (4.4), the tempered maximum pseudolikelihood maximizes a modified pseudolikelihood as follows:

$$\Theta_{ML\beta}^* \simeq \operatorname{argmax}_{\Theta} \log \prod_{m=1}^M \prod_i P^{\beta}(c_i^m, t_i^m | C_{N_i}^m, T_{N_i}^m, I^m, \Theta) \quad (4.12)$$

The tempered pseudolikelihood is equivalent to discounting the corresponding free energy by a multiplicative constant β . When β is small, or equivalently, when the temperature is high, the parameter learning is encouraged to move around the feasible space more freely. This is observed to have the effect of discounting each conditional probability in Equation (4.12) to make each of them contribute more evenly to the joint distribution. Unlike the ‘inverse annealing’ and cross-validation procedure in [55], the tempered maximum pseudolikelihood used for learning the coupled Conditional Random Field proceeds in the following manner, simply similar to deterministic annealing [96]:

1. Initialize β with a small constant and perform maximum pseudolikelihood to estimate parameters.
2. Using previous steps as initialization, increase β with a small step and perform maximum pseudolikelihood.
3. Run step 2 until β reaches 1.

4.4.4 Parameter Initialization

To initialize the parameters for the non-linear optimization of maximum log-pseudolikelihood, each of the evidence and compatibility functions in Table 4.2 is first trained separately with standard maximum likelihood estimation methods for learning logistic regression [83], assuming points are independent. More specifically, for example, the parameters α_0 , α_1 , and α_2 of the contour evidence function $\Phi_c(c_i|I) = \frac{1}{1 + e^{-c_i(\alpha_0 + \alpha_1 cm_i + \alpha_2 tm_i)}}$ can be learned individually by maximizing its likelihood on a set of training data, without considering the effect of other functions in Table 4.2. Then the learned parameters are used as a starting point for the non-linear optimization on the joint pseudolikelihood.

4.4.5 Model Inference

With the learned model, generating the most plausible labeling of contour and texture processes for test images, given their image measurements, can be solved with probabilistic inference. There are some alternative probabilistic inference criteria that can be used: Maximum Posterior Marginal (MPM) [54, 66], Minimum Mean Squared Error (MMSE) [41], and Maximum a Posteriori (MAP) [41] inference. In [54] and [66], MPM is argued to be more stable than a MAP solution. In our experiments, MPM inference and MAP inference are observed to perform very similarly to each other, with negligible differences. Maximum a Posteriori (MAP) inference is chosen for the experiments in this thesis.

The labeling problem in this thesis is defined on a loopy graph. For probabilistic inference on a loopy graph, exact inference can occur only in special cases, *e.g.*, Min-cut algorithm on a binary Ising-model random field. In general cases such as the one in this thesis, exact inference is intractable, hence approximate inference techniques have to be used instead. Many alternatives of approximate inference methods, such as Belief Propagation [41, 90], Gibbs sampling [46], and Graph Cut [19, 50], exist. In the experiments in this thesis, approximate Maximum a Posteriori (MAP) inference is carried out using loopy Belief Propagation.

4.5 Model Learning and Evaluation

In the following sections, the proposed Conditional Random Field model is trained with a set of labeled images. The parameters of the evidence and compatibility functions of contour and texture processes as in Table 4.2 are learned by maximizing the pseudo-likelihood on the set of training images. The learned parameters are shown to capture the distinct properties of contour and texture processes. Numerical evaluation of the learned models is carried out on a set of test images. In the current implementation, the neighborhood is set to 11×11 for both the contour and texture processes.

4.5.1 Model Learning and Analysis

The coupled Conditional Random Field is trained on a set of ground truth data, which are manually labeled. A Matlab-based labeling tool was designed to label edge points in images as either contour or texture points. Fourteen images were labeled as a training set, which is shown in Figure 4-3. These images are rich in both contour and texture, making them suitable for learning the interactions within and between the two processes. For each of the images, edge points are first extracted. With the labeling tool, most of the edge points are labeled as either contour or texture. In Figure 4-3, labeled contour edge points are shown in red, and labeled texture edge points are shown in green. Unlabeled edge points are shown in black.

To make clear the advantage of the proposed coupled Conditional Random Field vs. the single-layer Conditional Random Field, the single-layer Conditional Random Field in Figure 4-1 is also trained for comparison purposes. The parametrization of the single-layer Conditional Random Field is shown in Table 4.3. The evidence and compatibility functions have the same logistic regression forms as in the coupled Conditional Random Field, for a fair comparison. A noticeable difference is that, for the single-layer Conditional Random Field, the different interaction dynamics of contour process and texture process are represented within one compatibility function $\Psi_e(e_i, e_j|I)$, which, as will be shown later, leads to a forced trade-off between different

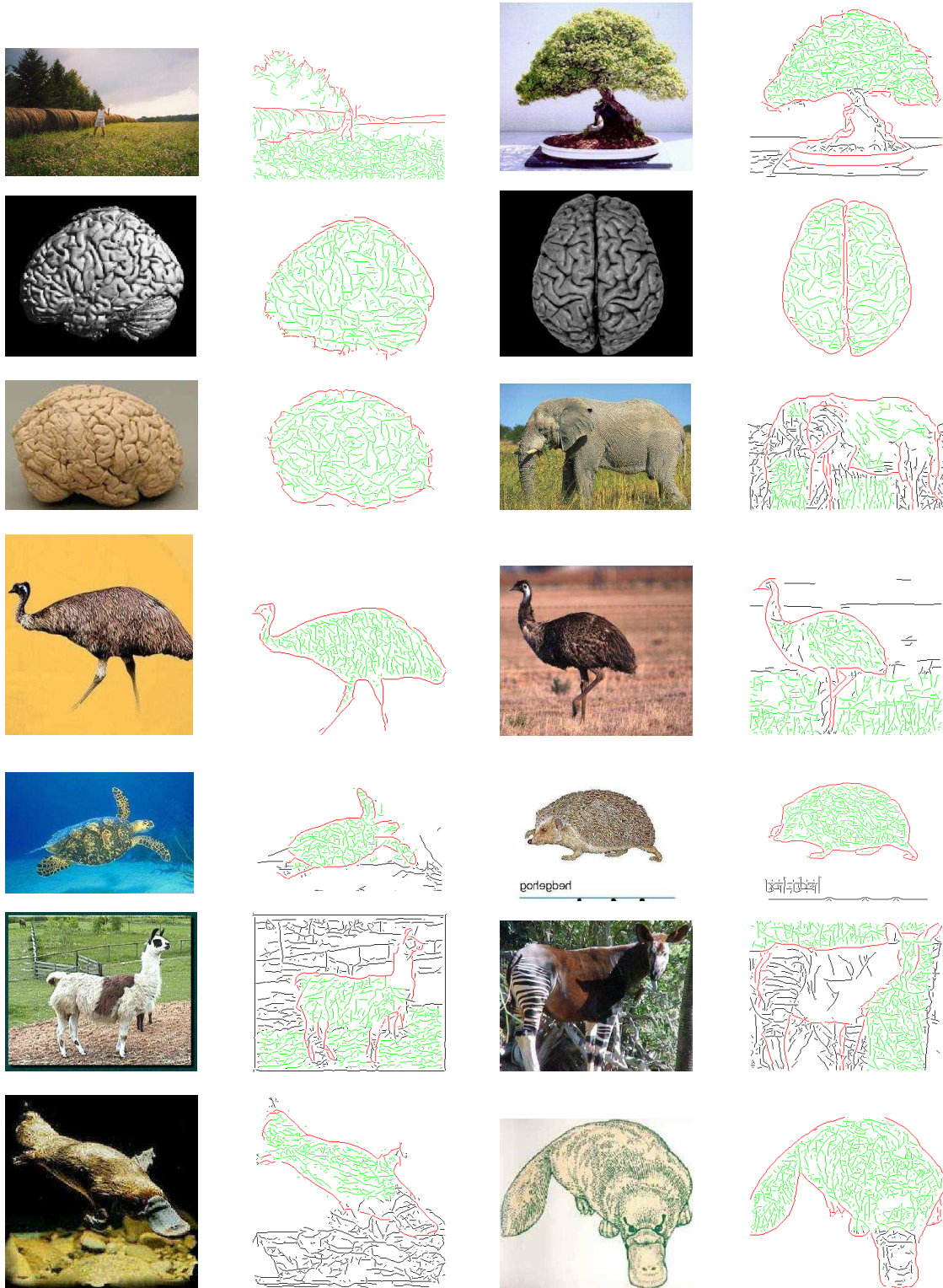


Figure 4-3: Training set for learning coupled Conditional Random Field and single-layer Conditional Random Field. Edge points are first extracted, then a majority of the extracted edge points are labeled as either contour or texture. Contour edges are shown in red color, while texture as yellow. White edges are left unlabeled.

<ul style="list-style-type: none"> • Variables for a pixel i e_i: labeling variable edge pixel e_i: $e_i = 1$: contour pixel; $e_i = -1$: non-contour pixel, <i>i.e.</i>, texture pixel. cm_i: contourness measure. tm_i: textureness measure. $\delta\theta_{ij}$: angle between orientation of i and the line joining i and a neighboring pixel j. δcm_{ij}: absolute difference between the contourness of i and j. δtm_{ij}: absolute difference between the textureness of i and j. • Evidence function of $\Phi_e(e_i I)$ $\Phi_e(e_i I) = \frac{1}{1 + e^{-e_i(\lambda_0 + \lambda_1 cm_i + \lambda_2 tm_i)}}$ • Compatibility function of $\Psi_e(e_i, e_j I)$ $\Psi_e(e_i, e_j I) = \frac{1}{1 + e^{-e_i e_j(\eta_0 + \eta_1 \delta\theta_{ij} + \eta_2 \delta cm_{ij} + \eta_3 \delta tm_{ij})}}$
--

Table 4.3: Evidence and compatibility functions of a single-layer Conditional Random Field model of contour and texture processes. See text for detailed explanation.

<p>a. Model-$\delta\theta$: compatibility only depends on angular difference: $\Psi_e(e_i, e_j I) = \frac{1}{1 + e^{-e_i e_j(\eta_0 + \eta_1 \delta\theta_{ij})}}$</p> <p>b. Model-$\delta cm$: compatibility only depends on contourness difference: $\Psi_e(e_i, e_j I) = \frac{1}{1 + e^{-e_i e_j(\eta_0 + \eta_2 \delta cm_{ij})}}$</p> <p>c. Model-$\delta tm$: compatibility only depends on textureness difference: $\Psi_e(e_i, e_j I) = \frac{1}{1 + e^{-e_i e_j(\eta_0 + \eta_3 \delta tm_{ij})}}$</p> <p>d. Model-<i>all</i>: compatibility depends on all measurements: $\Psi_e(e_i, e_j I) = \frac{1}{1 + e^{-e_i e_j(\eta_0 + \eta_1 \delta\theta_{ij} + \eta_2 \delta cm_{ij} + \eta_3 \delta tm_{ij})}}$</p>

Table 4.4: Different compatibility functions of a single-layer Conditional Random Field.

dynamics and worse decomposition results. The same set of training samples is used for training the single-layer Conditional Random Field model.

In the full models in Tables 4.2 and 4.3, the compability functions depend on three image measurements - (1) $\delta\theta_{ij}$, angular difference between the orientation of edge pixel i and the line joining i and j ; (2) $\delta cm_{ij} = |cm_i - cm_j|$, absolute contourness difference between pixels i and j ; (3) $\delta tm_{ij} = |tm_i - tm_j|$, absolute textureness difference between pixel i and j . Compatibility functions in different models (coupled Conditional Random Field and single-layer Conditional Random Field) could have quite distinct dependencies on the three measurements, as discussed in Section 4.2. To better evaluate these different dependencies in different models, each model is also trained with the compatibility function dependent on only one of the three measurements. That is to say, for instance, the single Conditional Random Field model is trained with different implementation of compatibility function as shown in Table 4.4. The evidence function remains the same for all instances, *i.e.* $\Phi_e(e_i|I) = \frac{1}{1 + e^{-e_i(\lambda_0 + \lambda_1 cm_i + \lambda_2 tm_i)}}$. Similarly, the coupled Conditional Random Field model is also trained with different compatibility functions, each of which depends on a different set of image measurement(s). For simplicity, the four cases in Table 4.4 are referred to as Model- $\delta\theta$, Model- δcm , Model- δtm and Model-*all* respectively.

The learned parameters are listed in Table 4.5. There are several noticeable facts in the learned parameter of different models:

1. For Model- $\delta\theta$ where compatibility functions only depend on the angular difference $\delta\theta_{ij}$:

for the coupled Conditional Random Field, the learned parameters of the compatibility functions for contour process and texture process are different, with $\tau_0=5.4240$ and $\tau_1=-5.3993$ for contour and $\gamma_0=6.7777$ and $\gamma_1=-3.7558$ for texture, which captures the disparate interaction dynamics within the two processes respectively;

whereas the single-layer Conditional Random Field, with learned compatibility

Model- $\delta\theta$							
coupled CRF				single-layer CRF			
Φ_c	α_0	α_1	α_2	Φ_e	λ_0	λ_1	λ_2
	-7.4036	2.7973	7.1600		-4.8556	2.9285	6.1034
Φ_t	β_0	β_1	β_2				
	7.4036	-2.7973	-7.1600				
Ψ_c	τ_0	τ_1		Ψ_e	η_0	η_1	
	5.4240	-5.3993			0.7571	-0.5324	
Ψ_t	γ_0	γ_1					
	6.7777	-3.7558					

(a) Learned parameters of Model- $\delta\theta$ for both cCRF and single-layer CRF.

Model- δcm							
coupled CRF				single-layer CRF			
Φ_c	α_0	α_1	α_2	Φ_e	λ_0	λ_1	λ_2
	-4.4558	1.9039	7.5505		-5.4363	1.5867	7.5817
Φ_t	β_0	β_1	β_2				
	4.4558	-1.9039	-7.5505				
Ψ_c	τ_0	τ_2		Ψ_e	η_0	η_2	
	1.4656	-13.3836			2.1681	-12.7884	
Ψ_t	γ_0	γ_2					
	2.9286	-12.8005					

(b) Learned parameters of Model- δcm for both cCRF and single-layer CRF.

Model- δtm							
coupled CRF				single-layer CRF			
Φ_c	α_0	α_1	α_2	Φ_e	λ_0	λ_1	λ_2
	-6.3037	3.6045	5.5762		-4.9371	4.1724	4.1971
Φ_t	β_0	β_1	β_2				
	6.3037	-3.6045	-5.5762				
Ψ_c	τ_0	τ_3		Ψ_e	η_0	η_3	
	4.1161	-8.9647			1.5933	-4.3612	
Ψ_t	γ_0	γ_3					
	4.7566	-8.1626					

(c) Learned parameters of Model- δtm for both cCRF and single-layer CRF.

Model- <i>all</i>									
coupled CRF				single-layer CRF					
Φ_c	α_0	α_1	α_2	Φ_e	λ_0	λ_1	λ_2		
	-2.9588	2.7014	5.5123		-4.1046	2.4582	5.1092		
Φ_t	β_0	β_1	β_2						
	2.9588	-2.7014	-5.5123						
Ψ_c	τ_0	τ_1	τ_2	τ_3	Ψ_e	η_0	η_1	η_2	η_3
	2.7942	-2.3865	-8.8175	-3.1492		2.7313	-1.3540	-8.8621	-1.9071
Ψ_t	γ_0	γ_1	γ_2	γ_3					
	3.7624	-0.9594	-8.6021	-2.1545					

(d) Learned parameters of Model-*all* for both cCRF and single-layer CRF.

Table 4.5: Learned parameters for different models of coupled Conditional Random Field and single-layer Conditional Random Field.

parameters of $\eta_0=0.7571$ and $\eta_1=-0.5324$, makes a forced compromise while using only one function to account for both dynamics in the two otherwise distinct processes.

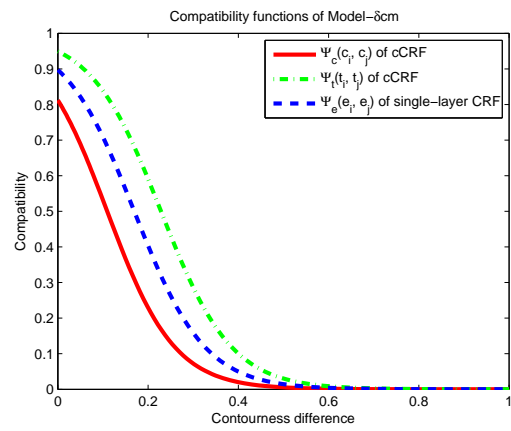
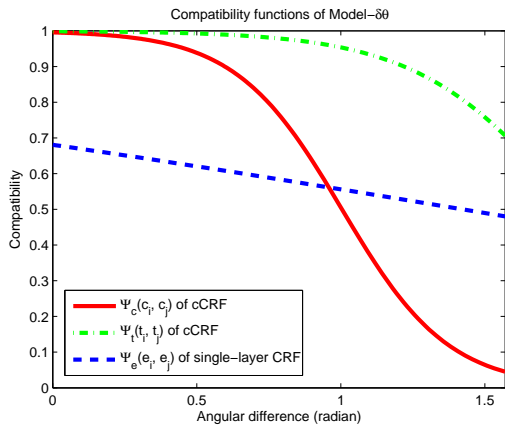
2. For Model- δcm and Model- δtm :

for the coupled Conditional Random Field, the learned parameters of contour and texture processes are slightly different; while for the single-layer Conditional Random Field, the learned parameters are again compromises to those in the coupled Conditional Random Field.

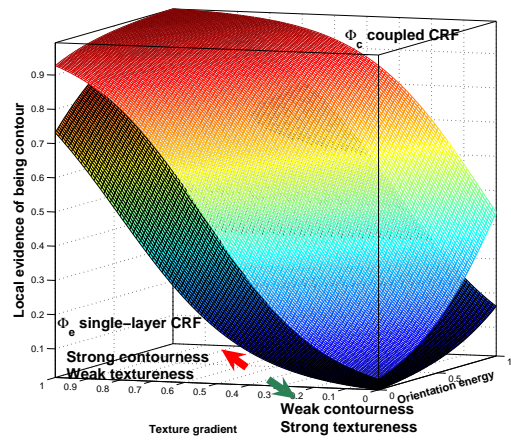
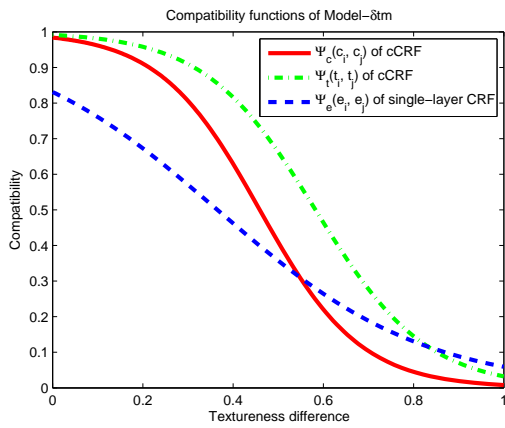
3. For Model-*all* where compatibility functions depend on all three measurements:

for the coupled Conditional Random Field, for contour compatibility, the learned parameter for dependency on angular difference $\delta\theta_{ij}$, which is $\tau_1=-2.3727$, is quite different from the corresponding parameter for texture which is $\gamma_1=-0.9708$. Other learned parameters of the compatibility functions are comparable for both processes; while for the single-layer Conditional Random Field, the learned parameters are an apparent compromise, with $\eta_1=-1.3580$, which lies between τ_1 and γ_1 .

To visualize the above differences, the learned compatibility functions of Model- $\delta\theta$, Model- δcm and Model- δtm are drawn in Figure 4-4. Figure 4-4(a) clearly shows that $\Psi_c(c_i, c_j)$ (red curve) and $\Psi_t(t_i, t_j)$ (green curve) are distinct. Roughly speaking, for small angular differences, *e.g.*, less than 0.5 radian (28.6 degree), $\Psi_c(c_i, c_j)$ gives high compatibility of more than 0.9; whereas for large angular differences, *e.g.*, larger than 1 radian (57.3 degree), the compatibility is smaller than 0.5. This means the contour compatibility function $\Psi_c(c_i, c_j)$ encourages local alignment of edge points, which is consistent with intuition. On the contrary, the compatibility function $\Psi_t(t_i, t_j)$ of texture process remains large for nearly all angular differences ($0 \sim \frac{\pi}{2}$). As a comparison, the compatibility function $\Psi_e(e_i, e_j)$ (blue curve) of the single-layer Conditional Random Field is forced to account for two different interaction dynamics hence lies between the two different compatibility functions.



(a) Compatibility functions of Model- $\delta\theta$ (b) Compatibility functions of Model- δcm



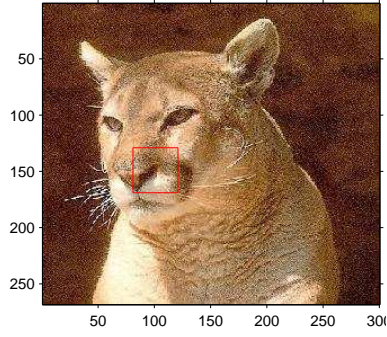
(c) Compatibility functions of Model- δtm (d) Evidence functions of Model-*all*

Figure 4-4: Comparisons of different models (better view in color).

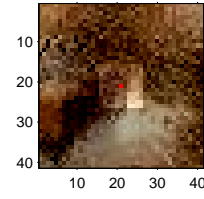
Figure 4-4(b) shows that the dependencies of $\Psi_c(c_i, c_j)$ and $\Psi_t(t_i, t_j)$ on δcm are comparable for contour and texture processes. Figure 4-4(c) reveals the same observation for the dependencies of $\Psi_c(c_i, c_j)$ and $\Psi_t(t_i, t_j)$ on δtm . Again, the compatibility functions of the single-layer Conditional Random Field are compromises of the two processes.

Figure 4-4(d), which plots the learned evidence function Φ_c of the coupled Conditional Random Field and Φ_e of the single-layer Conditional Random Field (whose parameters are shown in Table 4.5(d)), shows another important difference between the coupled Conditional Random Field and the single-layer Conditional Random Field. In the 3D plots in Figure 4-4(d), the evidence function Φ_c of the coupled Conditional Random Field lies above Φ_e of the single-layer Conditional Random Field. This indicates that the single-layer Conditional Random Field is stricter in assigning local evidence of contourness, *i.e.*, only edge points with strong contourness and weak texture measurements are given larger local evidence of being contour, whereas the coupled Conditional Random Field relaxes this compared with the single-layer Conditional Random Field, allowing a much wider range of measurements to be considered as possible contour. This is also intuitively correct, since the single-layer Conditional Random Field has no other strong cues of detecting contour while the coupled Conditional Random Field is able to rectify contour with local angular alignment in the compatibility function.

An example is shown in Figure 4-5. A small patch from the cougar face image in Figure 4-5(a) is expanded to see the behavior of contour and texture compatibility functions in the coupled Conditional Random Field as compared with a single-layer Conditional Random Field. The contour compatibility in cCRF of the highlighted small patch in Figure 4-5(b) is shown in Figure 4-5(c). The compatibility values shown in Figure 4-5(c) are the contour compatibilities of nearby edge points to the center edge point (marked with a red dot in Figure 4-5(b)). It can be seen from Figure 4-5(c) that, in the contour channel, only edge points that are well aligned with the center point have high compatibilities with the center point. Other points off the local contour have low compatibilities. For the compatibility in texture channel as in Figure



(a) Cougar face example



(b) A patch in the cougar face

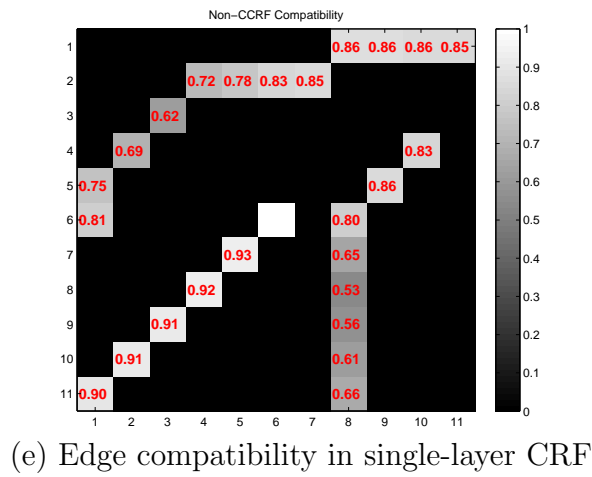
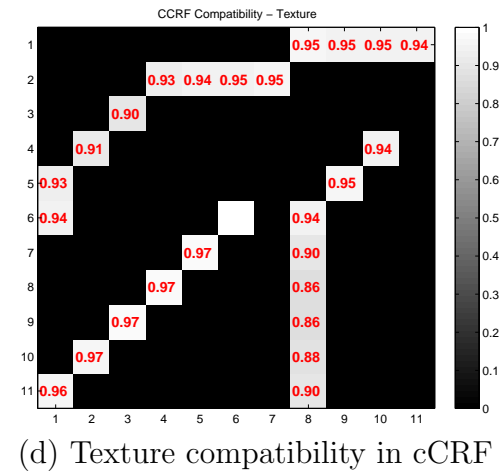
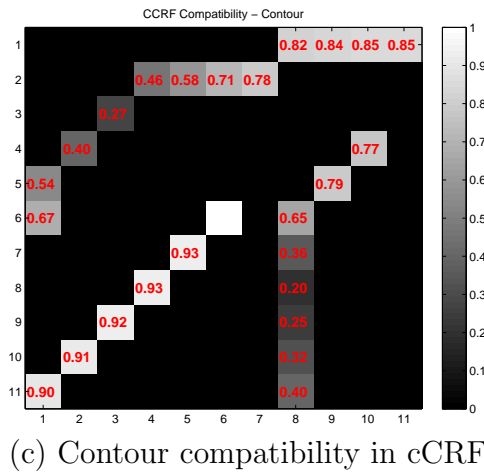


Figure 4-5: Comparisons of compatibility functions of different models.

4-5(d), since the texture channel allows random layout, all points in Figure 4-5(d) have quite good compatibilities because those points all come from similar texture regions. On the contrary, if all compatibilities are modeled in a single-layer CRF, since the compatibility function has to represent both characteristics, the compatibility values in Figure 4-5(e) are apparent trade-off between Figure 4-5(c) and (d).

4.5.2 Model Evaluation

In order to quantitatively evaluate the proposed coupled Conditional Random Field model and compare it with the single-layer Conditional Random Field, 17 different images are used as a test set. Similar to the training set, the images in this test set are also rich in both contour and texture, suitable for evaluating contour and texture interaction. Edges are extracted and labeled using the labeling tool. Then, probabilistic inference by loopy Belief Propagation is applied to find a Maximum-a-Posteriori (MAP) estimate of contour and texture labeling. The decomposed contour and texture are compared with the ground truth labels. Both full models in Table 4.2 and 4.3 (coupled Conditional Random Field and single-layer Conditional Random Field respectively) are tested and compared.

The performance of different models is evaluated with precision-recall rates and corresponding F-measure [94]. To see how different models perform on different visual cues, for each model, precision-recall and F-measure are evaluated on three settings: ‘contour process only’, which measures the models’ performance on contour detection; ‘texture process only’, which measures the models’ performance on texture detection; and ‘both contour-texture processes’, which measures the models’ performance on both contour detection and texture detection, *i.e.*, the performance of contour-texture decomposition. Precision of contour detection is the probability that a model-generated contour edge pixel is a true contour pixel, measuring how much noise is in the output of the model; recall is the probability that a true contour edge pixel is decomposed as contour by the model, measuring how much ground truth is detected. F-measure is the harmonic mean of precision and recall, giving a one-number summary of performance. Similarly for the texture channel. More specifically:

Precision of contour process:

$$\text{Precision}_{\text{contour}} = \frac{P_c}{M_c}$$

where M_c is the total number of detected contour points by the model, and P_c is the number of detected contour points that are labeled as contour.

Precision of texture process:

$$\text{Precision}_{\text{texture}} = \frac{P_t}{M_t}$$

where M_t is the total number of detected texture points by the model, and P_t is the number of detected texture points that are labeled as texture.

Recall of contour process:

$$\text{Recall}_{\text{contour}} = \frac{R_c}{N_c}$$

where N_c is the total number of labeled contour points in an image, and R_c is the number of labeled contour points that are correctly identified by the model.

Recall of texture process:

$$\text{Recall}_{\text{texture}} = \frac{R_t}{N_t}$$

where N_t is the total number of labeled texture points in an image, and R_t is the number of labeled texture points that are correctly identified by the model.

In each case, the final precision ‘ $\text{Precision}_{\text{avg}}$ ’ and recall ‘ $\text{Recall}_{\text{avg}}$ ’ of a model is determined by the average of per-image precision and recall rates on the entire test set. For each case, the F-measure is:

$$F = \frac{2 \cdot \text{Precision}_{\text{avg}} \cdot \text{Recall}_{\text{avg}}}{\text{Precision}_{\text{avg}} + \text{Recall}_{\text{avg}}}$$

For contour-texture decomposition, the precision-recall is the overall precision-recall considering both processes. Since each image is only partially labeled, the denominators have to be the same for both precision and recall, which is the number of

labeled points. This leads both precision and recall to the same number. According to the definition of F-measure, in this case, F-measure of contour-texture decomposition is the same as precision and recall.

The results of the above evaluation are shown in Table 4.6. For contour detection as shown in Table 4.6(a), the coupled Conditional Random Field gives a prominent improvement on contour recall, with 25.6% improvement compared with the single-layer Conditional Random Field. In one number, the coupled Conditional Random Field outperforms the single-layer Conditional Random Field in terms of F-measure by a difference of 0.1113. For texture detection as shown in Table 4.6(b), the coupled Conditional Random Field also exceeds the performance of the single-layer Conditional Random Field with an F-measure of 0.8986. Over all, as in Table 4.6(c), the coupled Conditional Random Field, with 87.53% decomposition precision, is much better than the decomposition by the single-layer Conditional Random Field.

	Recall _{contour}	Precision _{contour}	F _{contour}
coupled CRF	83.93%	80.97%	0.8243
single-layer CRF	58.33%	91.69%	0.7130

(a) Performance of models on contour process

	Recall _{texture}	Precision _{texture}	F _{contour}
coupled CRF	89.65%	90.07%	0.8986
single-layer CRF	97.67%	80.42%	0.8821

(b) Performance of models on texture process

	Recall _{decomp} = Precision _{decomp} = F _{decomp}
coupled CRF	87.53%
single-layer CRF	83.20%

(c) Performance of models on contour texture decomposition

Table 4.6: Performance evaluation of different models.

Figure 4-6 compares side-by-side the precision and recall rates of the two models for each of the test images. Figure 4-6(a) shows the per-image recall-rates and precision-rates for contour detection. Cyan bars are rates for the coupled Conditional Random Field model, with magenta bars for the single-layer Conditional Random

Field. It is clear in the graph that recall-rates of the coupled Conditional Random Field consistently outperform those of the single-layer Conditional Random Field. Figure 4-6(b) and 4-6(c) show the per-image recall-rates and precision-rates for texture detection and overall contour-texture decomposition respectively. In each case, the coupled Conditional Random Field either outperforms or is comparable to the single-layer Conditional Random Field.

For visual comparison, the decomposition results by the two models on the test images are shown in Figure 4-7, which clearly shows that the coupled Conditional Random Field, with the capability of accounting for different dynamics in different layers, has much better decomposition, especially in the contour channels. Whereas the single-layer Conditional Random Field misses many contour edges and over-estimates texture process, especially for contour points where there exist many nearby texture points. This clearly shows the importance of the proposed coupled Conditional Random Field model.

Figure 4-8 gives some examples of typical images where the proposed method's performance degrades. For instance, the spines of the hedgehog in Figure 4-8 have very high contrast, thus the local evidence of contour dominates the coupled Conditional Random Field inference and these pixels are grouped as contours. The contour of the cougar in Figure 4-8 has low contrast against the background thus a major part of the contour is missed. To recover these kinds of contours and textures correctly, we speculate that other image properties, such as regions and class shape models, should be incorporated into the framework.

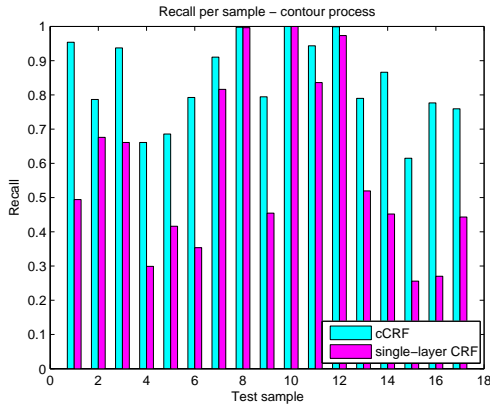
4.6 Summary

In this chapter, a novel coupled Conditional Random Field is proposed to model and decompose the contour and texture processes in natural images. The coupled Conditional Random Field model uses separate layers of random field grids to represent different processes. This structure allows each layer to focus on the interactive dynamics of each individual process. The interaction between different processes is

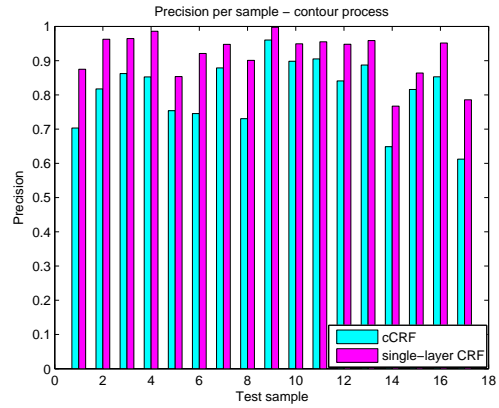
modeled by the coupling links between different layers. The forms of the evidence and compatibility functions in the coupled Conditional Random Field are discriminative logistic regression functions. Learning and inference methods for the coupled Conditional Random Field are developed.

The importance of the proposed coupled Conditional Random Field is first shown with the analysis of the learned model parameters. As compared with a single-layer Conditional Random Field, the coupled Conditional Random Field is able to capture the distinct interactive dynamics of different processes. On the contrary, without separate layers for each individual process, the single-layer Conditional Random Field is forced to introduce unacceptable compromises using a single compatibility function to model the disparate dynamics of different processes. An empirical evaluation is also carried out on a set of labeled test images. The F-measure based on the precision-recall rates shows that the proposed coupled Conditional Random Field outperforms the single-layer Conditional Random Field in contour and texture decomposition.

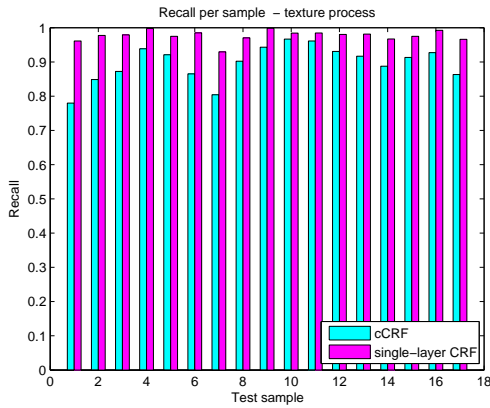
In the proposed computational model of object recognition, the contour and texture channels decomposed by the coupled Conditional Random Field model are first matched separately and then combined for the purpose of object recognition. In the next chapter, suitable appearance and geometric features are introduced for matching each individual channel across different objects. Adaptive combination of multiple visual cues will be developed in Chapter 6.



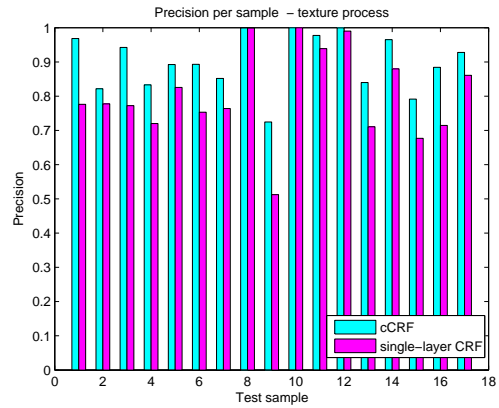
(a) Per-image recall of contour detection



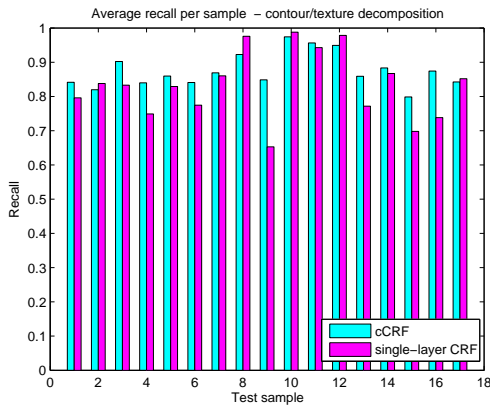
Per-image precision of contour detection



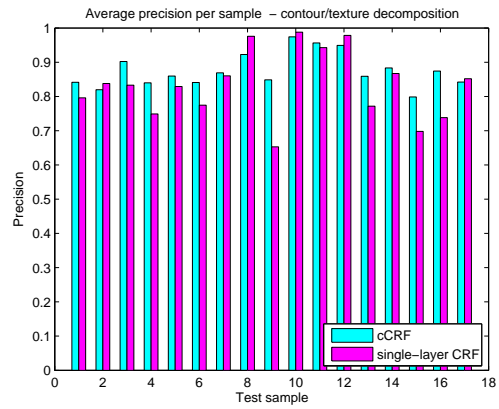
(b) Per-image recall of texture detection



Per-image precision of texture detection



(c) Per-image recall of decomposition



Per-image precision of decomposition

Figure 4-6: Comparisons of per-image precision and recall rates of coupled Conditional Random Field and single-layer Conditional Random Field.



Image and edge labeling



Contour and texture decomposition by coupled CRF



Contour and texture decomposition by single-layer CRF

Figure 4-7 (a)

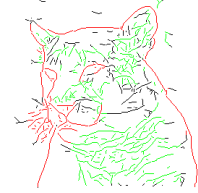
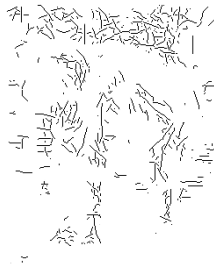
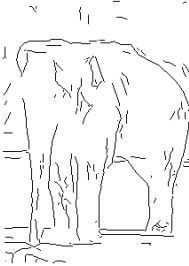


Image and edge labeling



Contour and texture decomposition by coupled CRF



Contour and texture decomposition by single-layer CRF

Figure 4-7 (b)

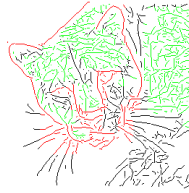
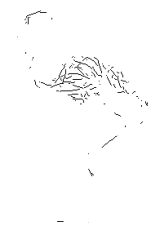


Image and edge labeling



Contour and texture decomposition by coupled CRF



Contour and texture decomposition by single-layer CRF

Figure 4-7 (c)

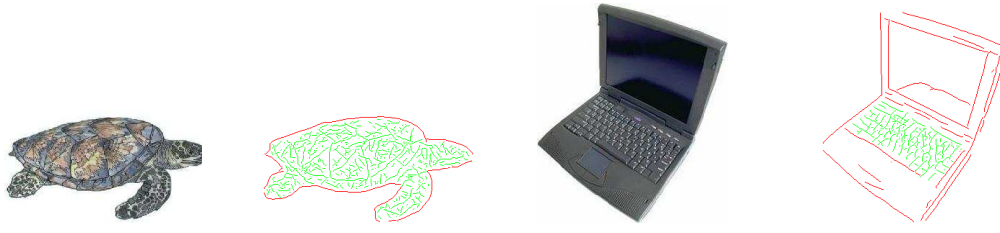
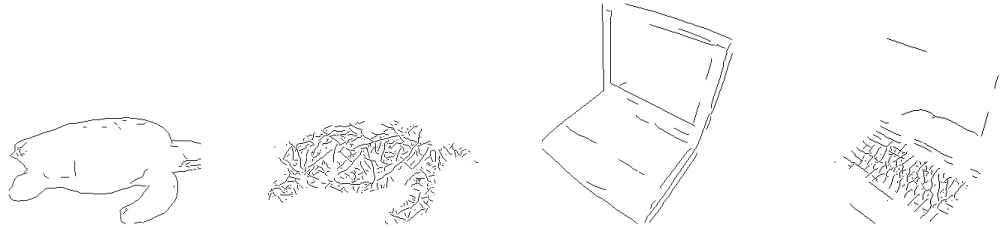
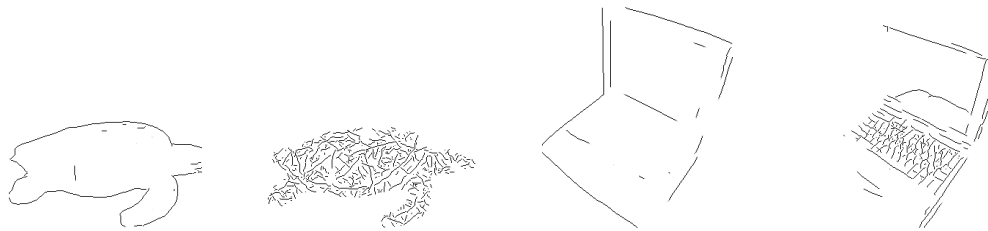


Image and edge labeling



Contour and texture decomposition by coupled CRF



Contour and texture decomposition by single-layer CRF

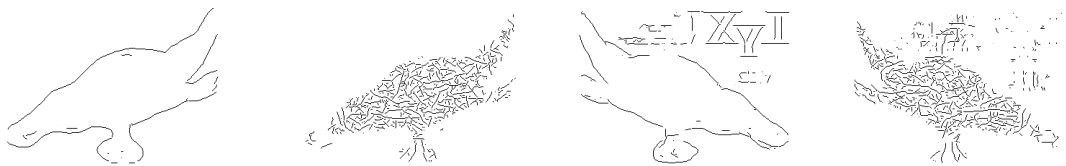
Figure 4-7 (d)



Image and edge labeling



Contour and texture decomposition by coupled CRF



Contour and texture decomposition by single-layer CRF

Figure 4-7 (e)

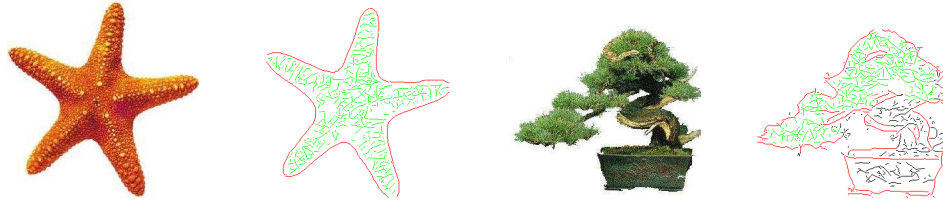
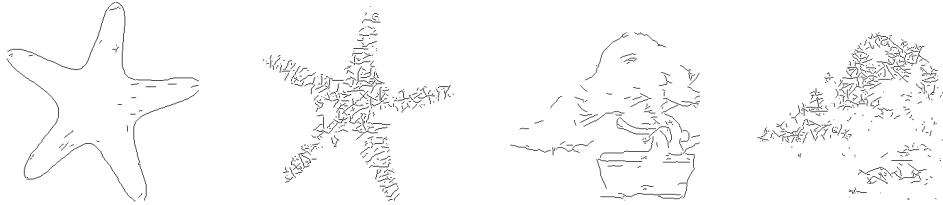
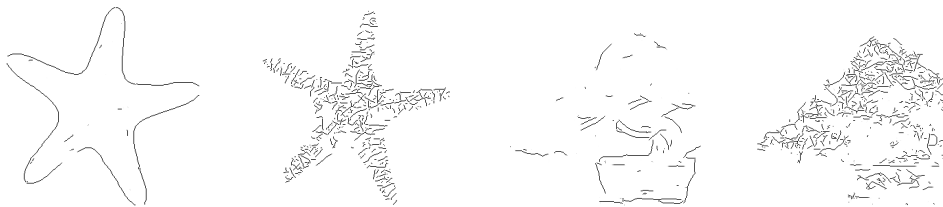


Image and edge labeling



Contour and texture decomposition by coupled CRF



Contour and texture decomposition by single-layer CRF

Figure 4-7 (f)



Image and edge labeling



Contour and texture decomposition by coupled CRF



Contour and texture decomposition by single-layer CRF

Figure 4-7 (g)

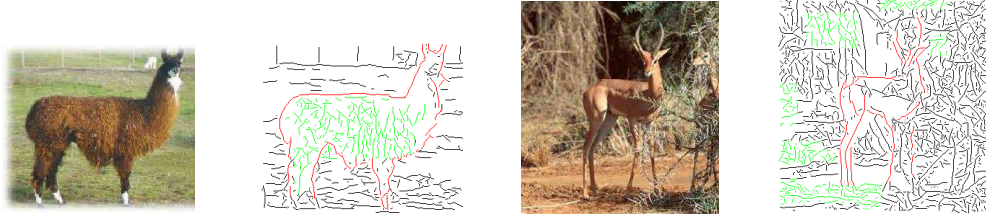
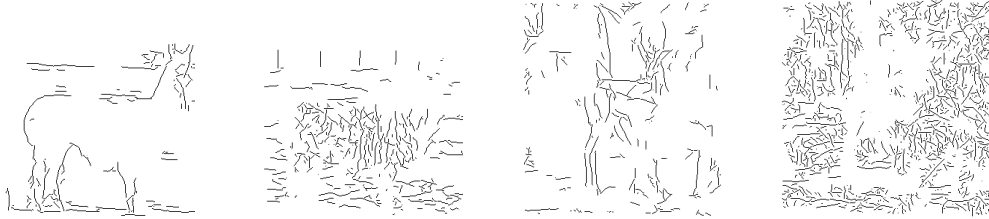
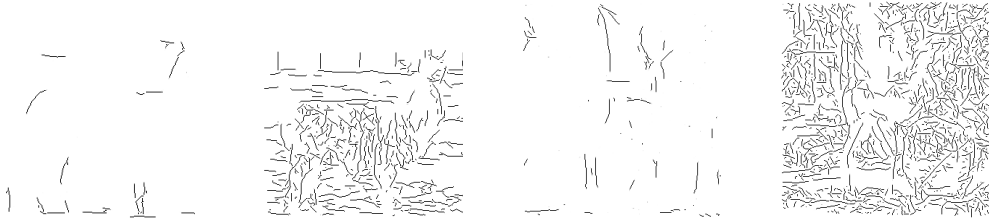


Image and edge labeling



Contour and texture decomposition by coupled CRF



Contour and texture decomposition by single-layer CRF

Figure 4-7 (h)



Image and edge labeling



Contour and texture decomposition by coupled CRF



Contour and texture decomposition by single-layer CRF

Figure 4-7 (i)

Figure 4-7: Comparison of contour and texture decomposition by the coupled Conditional Random Field and the single-layer Conditional Random Field models.

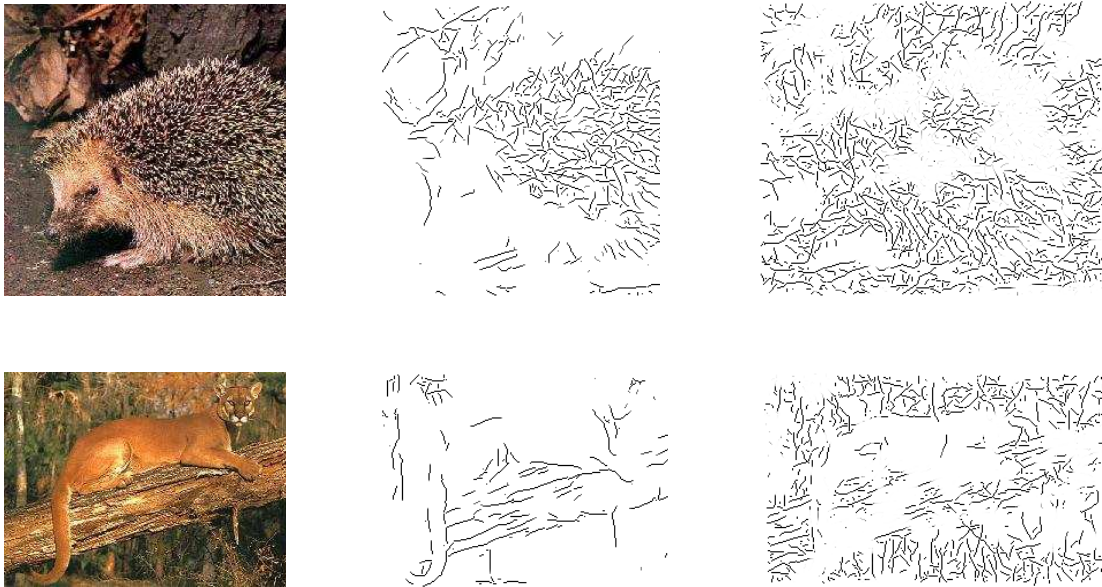


Figure 4-8: Some examples where coupled Conditional Random Field's decomposition performance degrades.

Chapter 5

Matching Decomposed Visual Cues

With the coupled Conditional Random Field model in Chapter 4, visual information such as contour and texture in images are decomposed into different channels. Each of the decomposed visual information captures a distinct perceptual aspect of objects. Thus, with the decomposition, we are able to investigate different visual stimuli separately to fully leverage each perceptual cue. This chapter introduces the features and the matching schemes for the decomposed contour and texture channels, and empirically evaluates the effects that various parameters of the matching schemes exhibit on the performance of object recognition.

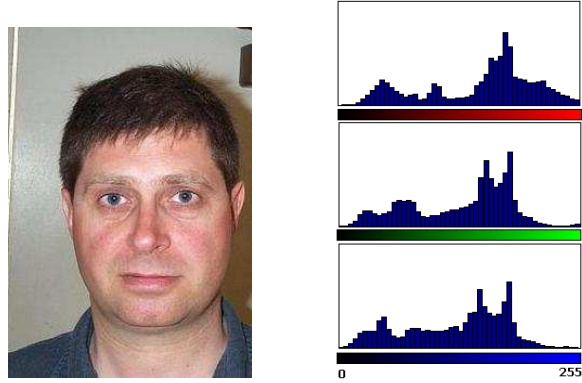
5.1 Choices of Matching Schemes

In Chapter 3, low-level measurements are introduced for the purpose of measuring contourness and textureness, and modeling and decomposing contour and texture in images with the proposed coupled Conditional Random Field. For the task of object recognition, the appearance and geometry features in these decomposed visual channels can be used to match different objects. In general, there are three levels of features and matching schemes for representing and recognizing visual resemblance: global feature matching, semi-local feature matching and local feature matching. Global features, such as color histogram [45, 104, 106] and eigenspace representation [29, 84], generally compress all the visual information in an image into

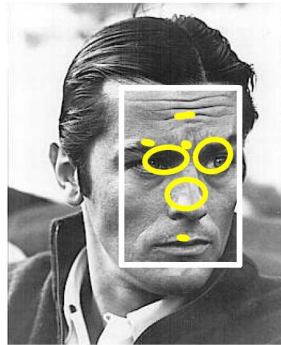
a holistic representation derived in a global statistical manner. For example, Figure 5-1(a) shows the RGB-histogram feature for a face image, where RGB values of all pixels in the image are aggregately counted into histogram bins. Semi-local features are typically defined as certain geometric configurations of local regions that are stable within multiple object instances of the same class, and in some cases, across a range of views of the objects. Figure 5-1(b) illustrates that some facial regions such as eyes, nose and eyebrows have a unique appearance and geometric composition and form a semi-local feature that is characteristic to the class of face objects. Local appearance features describe what relatively small regions or patches in an object image look like and represent the object with the collection of appearance descriptors of these local regions. Figure 5-1(c) draws a set of local features derived from a detector based on extrema of Different-of-Gaussians operators proposed by Lindeberg [74] and Lowe [77].

Recent advances have demonstrated the effectiveness of local appearance features compared with global features and semi-local features, in application fields such as image indexing and retrieval [101], wide-baseline stereo [99], video matching and search [103], object identification and categorization [38, 77]. Using local appearance features has many important advantages:

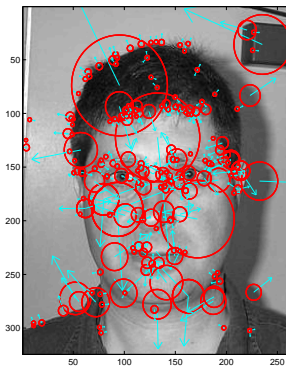
- **Robustness to clutter and occlusion:** Since local appearance features are typically small in spatial support, a significant portion of local features will remain within or around the object to be described, without including much of the background into descriptors. So even when an object is presented in a non-uniform or changing background, local appearance features can still remain largely unaffected and hence make matching across different images more robust.
- **Expressiveness:** Local features not only capture the local geometric configuration of salient shape elements such as points, lines and curves, but also represents the appearance of local regions. These rich high-dimensional descriptors provide strong cues for analyzing image content.
- **Compactness:** The number of extracted local features in general is much



(a) Global color histogram feature. RGB-values of pixels in the face image are formed into three histograms, one for each of the RGB color channels.



(b) Semi-local feature. The facial region marked by yellow circles forms a stable geometric configuration, or, a semi-local feature.



(c) Local features detected by a Different-of-Gaussians operator. Red circles are detected local features. Radiuses of circles represent detected scale. Blue arrows indicate detected feature orientation.

Figure 5-1: Examples of global features, semi-local features and local features of images.

smaller than the number of pixels in the original images. Image representation based on the collection of local features is typically compact while being expressive. Thus the processing time and storage requirements are significantly reduced with most of the visual information preserved.

Complementary to appearance matching, shape matching based on the geometry of objects has been extensively studied and shown to achieve good recognition performance in many applications too. Roughly speaking, there are two categories of shape matching: *exact shape matching* such as matching based on Hausdorff Distance [60, 98], Chamfer matching [15, 107], shape context [7] and deformable models [29], and *weak geometric matching* such as constellation model [38], implicit shape model [71], spatial pyramid matching [70] and boosted part-based model [5]. Exact shape matching is effective when objects are imaged with a clean uniform background, or objects can be segmented out from background, or clean exemplar objects can be obtained beforehand. With background clutter and no clean exemplar object, the performance of exact shape matching significantly degrades. Weak geometric matching can tolerate occlusion and background clutter to a greater extent than exact shape matching. Many weak geometric matching schemes depend on local features to detect salient object ‘parts’ and add additional layers of geometric representation on top of local appearance features to improve expressiveness and discriminability.

In this thesis, the visual content decomposition by the coupled Conditional Random Field enables many of the above matching schemes to be used in a collective and complementary way to fully utilize different characteristics of multiple visual stimuli. That is, we are able to use local appearance features to describe object images, and apply weak geometric matching to recognize the different visual content contained in contour and texture channels. In the meantime, even with background clutter and no clean exemplar object, since the salient contour structures in images are decomposed into a separate, clean and sparse channel, the decomposition scheme enables us to employ exact shape matching on the decomposed contours. This is shown to be complementary to weak geometric matching with local appearance features. Another merit of this ‘recognition-through-decomposition-and-fusion’ scheme, besides

empowering multiple complementary matching schemes to maximally leverage each individual channel, is its capability to selectively combine various matching schemes to adapt to different characteristics of different classes of objects to further improve recognition performance, which will be discussed in Chapter 6.

5.2 Local Features

As basic building blocks of the visual matching schemes, local appearance features are used for describing the appearance of each of the contour and texture channels. With contour and texture decomposition, the information in the original images is represented in a cleaner and clearer way. The contour channel captures salient curvilinear structures in the images, such as occluding boundaries of foreground objects and structured objects in the background. The texture channel encapsulates the non-structured yet oftentimes perceptually coherent collection of elements in many images, most of which come from characteristic texture of foreground objects such as keyboard patterns of laptops and background such as grassland. Thus the decomposition of visual information empowers more sensible feature extraction and matching, with different visual channels emphasizing different characteristics of an object of interest.

5.2.1 Feature Point Extraction

Before determining how to describe appearances, we need to determine what feature extraction scheme should be used, *i.e.*, how to determine the spatial location and scale of feature points, which is typically termed as ‘feature detector’.

Invariant Features

Over the last decade, since the seminal work of Schmid and Mohr [101] and Lowe [77] of applying multi-scale corner and blob detectors in object recognition, there has been a flurry of research and applications in invariant feature detectors. Invariant feature detectors [64, 74, 77, 81] process images to obtain a set of scale- or affine-

invariant regions, which are theoretically repeatable over images taken under different imaging conditions which bring geometric distortion to the images. These scale- or affine- invariant features are expected to normalize extracted patches by reversing corresponding geometric distortion, leading to visually comparable normalized feature regions.

One noticeable fact about invariant feature detectors is that most of these scale- or affine- invariant feature detectors were first derived for the task of matching the same objects or scenes under different viewing conditions, which is an ‘object identification’ problem. That is, the observed objects or scenes remain the same in different images hence the shape and appearance of the objects or scenes of interest remain largely unchanged up to a view-induced geometric distortion. By uncovering the underlying geometric distortion, invariant feature detectors are expected to extract the same salient points across different images of the same objects or scenes.

Due to the success of invariant features in ‘object identification’, researchers have been extending the application of invariant features to the problem of object categorization, (*i.e.*, to recognize different object instances of one class versus object instances of other classes,) with the expectation that salient invariant features would capture most of the representative and discriminative visual information contained in objects, and in the meantime be highly repeatable within different instances of the same object class. While the applications of invariant feature detectors have shown to be useful in many object categorization schemes [2, 38, 62, 113], there has been some doubt about the effectiveness of invariant features in object categorization. One critique is that, since object categorization deals with different object instances, invariant features are not truly invariant in different object instances of the same class, considering the large appearance variation of objects within the same class. Hence invariant feature detectors mainly act as a means to reduce the amount of information to be processed while keeping most of the salient information in the extracted representation. Another critique is invariant feature detectors often result in a very sparse set of features, especially when object images are relatively small. This practice of ‘thresholding’ information too early and too aggressively is suspected to be

less effective at representing and recognizing object images.

Densely Sampled Features

Due to the aforementioned shortcomings of invariant feature detectors, many approaches of object categorization have reverted to dense sampling of features on object images and have shown the effectiveness of this feature extraction method in object categorization [9, 70]. Dense sampling in the spatial and/or the scale space can achieve the same effect of reducing computational time while retaining most of the salient information in images. And dense sampling often extracts more features than invariant feature detectors and lets machine learning techniques in subsequent stages to utilize the features in a selective way. This avoids the problem of ‘pre-mature’ compression of information as in invariant feature detectors.

Given the above consideration, this thesis uses dense sampling to extract features.

5.2.2 Local Appearance Descriptor

As shown by the empirical evaluations by Mikolajczyk and Schmid [82], the SIFT (Scale Invariant Feature Transform) descriptor [77] is the most robust and discriminative feature in the task of visual matching, among many other alternatives. The descriptor used in this thesis is derived from this popular SIFT descriptor.

Original SIFT Descriptor

In its original form, the SIFT descriptor takes each local region, finds gradients of image grayvalues and then normalizes for orientation by finding the dominant gradient orientation and rotating the region to be aligned along the dominant orientation. Then 8-bin histograms of gradient orientation are formed in each cell of a 4×4 spatial grid on the local region. In forming the histograms, each pixel’s gradient orientation is weighted by its gradient magnitude. To put more weights on pixels near the center of the local region, each pixel is also weighted using a Gaussian function based on the pixel’s distance to the center. Robustness is also achieved by

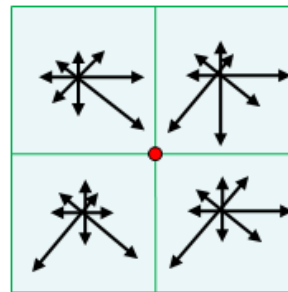
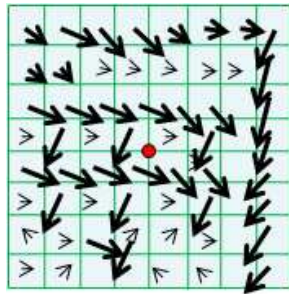
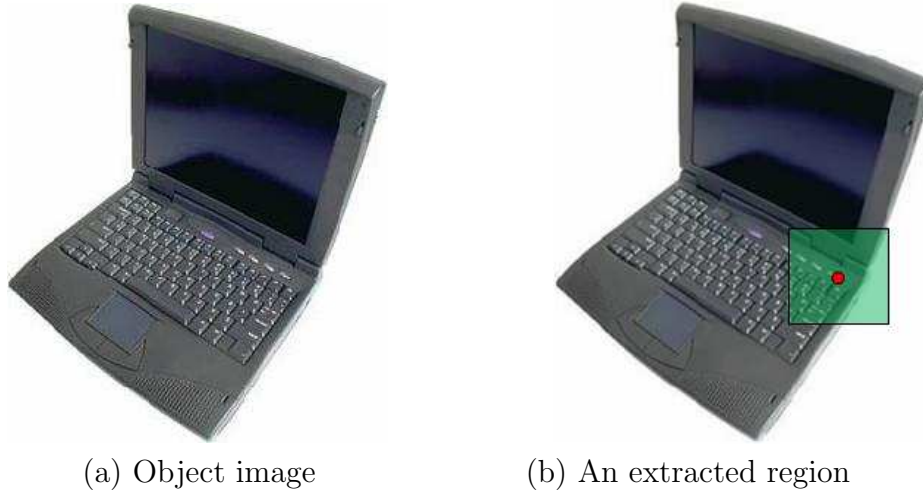


Figure 5-2: A schematic illustration of definition of the SIFT descriptor. See text for details.

soft-assigning pixel gradient to orientation bins based on the distances to sub-region boundaries and the distances to orientation bin centers. A 128 dimensional feature vector is formed by concatenating the 8-bin weighted gradient orientation histograms from all 16 cells. This 128 dimensional feature vector acts as a rich and expressive descriptor of the local region. Figure 5-2 gives an illustration of this process for a 32-dimensional SIFT descriptor. For the image of a laptop in Figure 5-2(a), a local region is extracted as indicated by the green square in Figure 5-2(b). Figure 5-2(c) is a schematic illustration of gradient magnitude and direction within the extracted region. The SIFT descriptor in Figure 5-2(d) is defined on a 2×2 spatial grid on the region, with each cell covering the spatially corresponding pixels in Figure 5-2(c). Each cell of the 2×2 spatial grid computes a weighted histogram of the gradient

within the corresponding 4×4 sub-squares in Figure 5-2(c). Concatenating the 8-bin histograms in the 4 cells in Figure 5-2(d) gives a 32-dimensional SIFT descriptor of the extracted region.

Lowe [77] and Mikolajczyk and Schmid [82] also show that the SIFT descriptor is robust to slight local deformation. For example, in Figure 5-2(d), if pixels in each of the 2×2 spatial cells move around to a small extent due to a slight affine transformation, as long as most of the pixels remain in the same cell before and after the small geometric transformation, the histogram in each cell by and large remains unchanged hence the overall descriptor remains stable.

SIFT Descriptor in Decomposed Contour and Texture Channels

While the original SIFT descriptor has been shown to be effective in many circumstances, many applications of the SIFT descriptor, such as many bag-of-feature approaches for image matching and retrieval, use the SIFT descriptor on local patches of the original image, which mix all pixels in a local patch and describe the patch as an integral entity. This practice essentially gives uniform weights to all the information contained in a patch. As stated in Section 1.1, in object recognition, it is more sensible that different visual cues should play different roles in discriminating various object classes. This characteristic can be represented and achieved with the contour and texture decomposition.

We use a SIFT-like descriptor on each of the decomposed contour and texture channels. In this descriptor, each image patch is divided into 3×3 cells, on each of which an 8-bin histogram of edge orientation is computed. Then the 9 histograms are concatenated into a 72-dimensional vector. The two SIFT vectors from the contour and texture channels form the descriptor for the patch. Figure 5-3 gives a schematic illustration of this process for 32-dimensional SIFT descriptors on the decomposed channels. First, the original laptop image in Figure 5-3(a) is decomposed by the proposed coupled Conditional Random Field model, resulting in two separate visual channels of contour and texture as in Figure 5-3(b) and (c) respectively. For some sampling point as indicated by the red dot in Figure 5-3(b) and (c), a local region

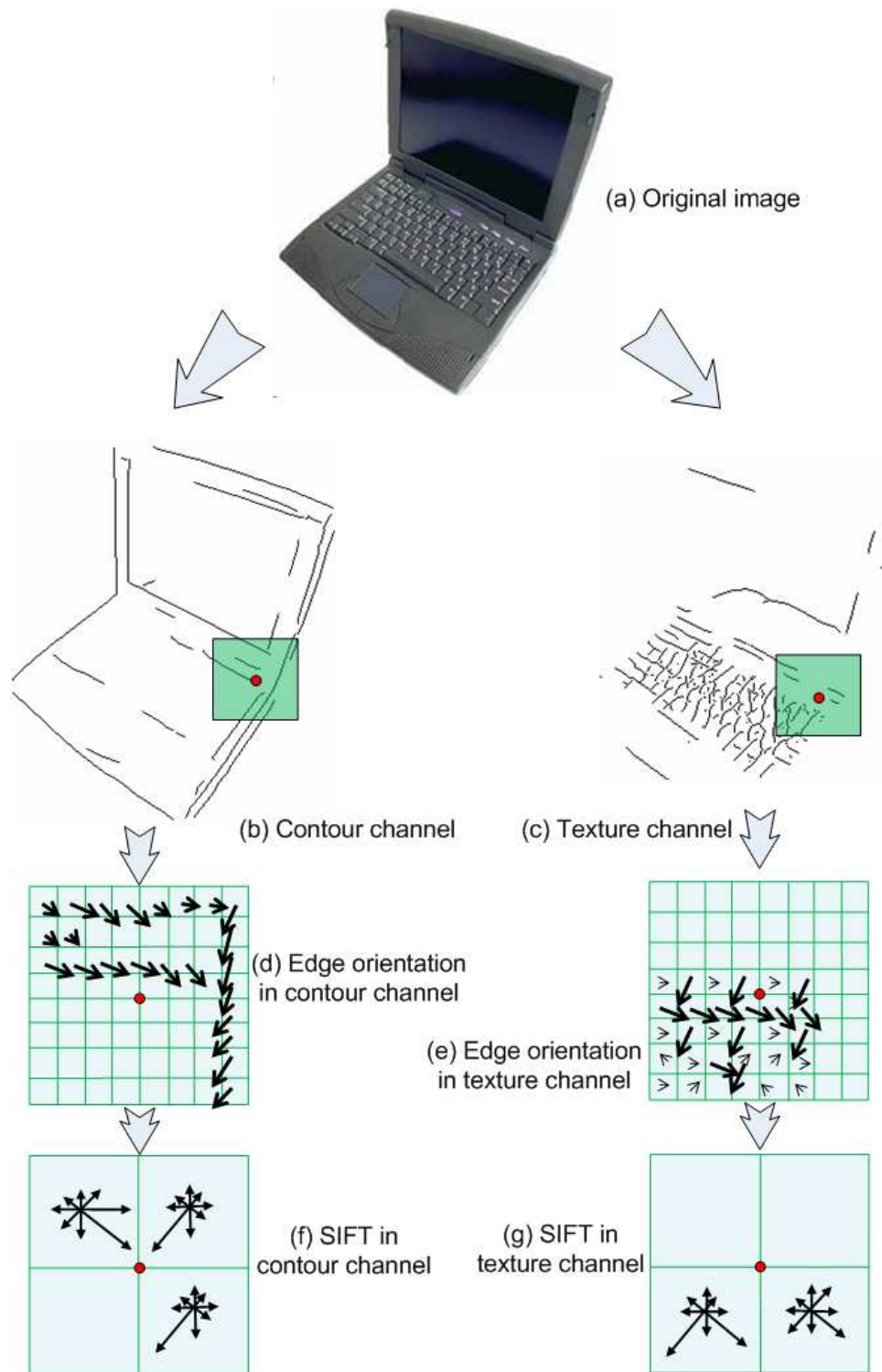


Figure 5-3: A schematic illustration of the SIFT descriptors on decomposed channels of contour and texture. See text for details.

(green square) is defined. Take the contour channel as an example. As illustrated by Figure 5-3(d), only edges belonging to the contour channel are considered in the edge orientation of the contour channel. And the SIFT descriptor in the contour channel only aggregates the weighted edge orientation for contour edges, to form the histograms in Figure 5-3(f). The concatenation of the histograms in Figure 5-3(f) forms the SIFT descriptor for the patch in the contour channel. Similarly for the SIFT descriptor in the texture channel, as illustrated in Figure 5-3(e) and (g).

There are two differences between the SIFT descriptor defined above and the original SIFT descriptor. The first difference to the original SIFT descriptor is that edge orientation in the descriptors is weighted by the edge pixels' probabilities of being contour or texture depending which channel is being described, whereas in the original SIFT, orientation is weighted by gradient magnitude. On one hand, using probabilities of being contour or texture approximately has the same effect as using gradient magnitude, *i.e.*, giving less weights to less confident points. On the other hand, by local belief propagation in the proposed coupled Conditional Random Field model, probabilities of edge pixels with weak evidence of contour or texture can be boosted while probabilities of false contour and texture pixels can be suppressed. Hence it is reasonable to expect the final probabilities from the MAP inference of coupled Conditional Random Field give better estimates of confidence in computing edge orientation histograms.

The second difference to the original SIFT descriptor lies in the normalization of dominant orientation. The original SIFT descriptor achieves rotational invariance by the aforementioned dominant orientation normalization. However, for the purpose of generic object categorization, the patch-wise rotational normalization is a dubious practice for many rigid or slightly non-rigid objects where only a single global rotational transformation exists. First, this kind of patch-wise rotational normalization is expected to reduce the discriminability of the descriptor since one degree of freedom is eliminated. Moreover, dominant orientations typically are inconsistent across different patches, as can be seen from the blue arrows in Figure 5-1(c). In the descriptor in this thesis, the dominant orientation normalization is removed. One

reason is that, for many applications the observed objects are imaged under their corresponding ‘canonical poses’ [23, 89]. Most of these canonical poses only involve very slight rotational variation. Another reason is, for objects under non-canonical poses, it could be more reasonable to first infer the global rotation of the objects and then use the global transformation to normalize the pose of the observed objects. After global rotational normalization, local patch-wise rotational normalization is no longer needed.

An additional merit of using the modified SIFT above is that the descriptors are computed on decomposed edge maps of the contour and texture channels, which are much sparser than the original image. Computing SIFT on edge maps doesn’t lose much information compared with computing SIFT on the original grayscale image, since large gradients dominate the original SIFT and edge maps are close approximation of large gradients. Without losing much of the discriminability of SIFT, we gain speed improvement due to the sparsity in computation of SIFT on the edge maps.

The soft-assignment of histograms to subregions and orientation bins, and Gaussian weighting based on pixel distances to the center pixel, as in the original SIFT, are kept because these techniques allow better robustness against small local geometric changes.

Although the descriptor definition is the same for the two channels, because the contents are different for the decomposed channels, the underlying structures captured by the descriptors stress different aspects of the object under analysis and have different semantic meanings in contour and texture channels. As illustrated by Figure 5-3(b) and (c), the contour channel has a sparse representation of prominent object contours, thus the descriptors in the contour channel emphasize more on local shape information, such as characteristic layouts of local elementary geometric components of lines, curves and corners etc., of the object. And the texture channel typically separates out characteristic elements such as fur, feather, keyboard patterns, leaves and grassland, which are also perceptually salient for related object classes and/or background. So the descriptors in the texture channel mainly focus on these important textural information contained in objects and backgrounds, without negatively

impacting the contour information in the other channel.

Build Visual Vocabularies

To match different images based on the SIFT descriptors, this thesis employs a vocabulary-based matching approach which will be discussed in Section 5.3.1. Here the definition and derivation of a visual vocabulary are introduced. As commonly termed in the field of object recognition, ‘visual words’ represent quantized local appearance descriptors. A ‘visual vocabulary’ is a collection of visual words which are quantized from local appearance descriptors of a set of training images. Using a quantized visual vocabulary usually helps to efficiently establish correspondences of local image patches in object identification and categorization. All local patches with the same visual word label are considered as matched patches. It is observed in practice that the ‘granularity’ of a visual vocabulary has some impact on the capability of generalization: a coarse quantization of descriptors tend to tolerate well intra-class variations, while a fine quantization of descriptors tend to capture more detailed differences in appearance of local patches and give more discriminative capability.

In this thesis, visual vocabularies are built from a small set of training samples of a dataset. For each of the contour and texture channels, a visual vocabulary is learned by clustering SIFT descriptors in the corresponding channel from the set of training samples. Clustering is carried out with a K-means algorithm.

5.2.3 Color Feature

In many classes of objects, color is also a significant visual cue for recognition. To extract color-based features, a HSV (Hue-Saturation-Value) representation of color is used. The HSV representation describes perceptual color relationships more accurately than RGB, while remaining computationally simple. This color description is more familiar for humans, in terms of the concepts of what kind of color it is, how saturated the color is, and whether the color is bright or dark.

A color dictionary is built in the following way. For the HSV space, Hue is

quantized into h_c bins, Saturation are quantized into s_c bins, and Value are quantized into v_c bins respectively. Thus $h_c \times s_c \times v_c$ color words are generated, which form a color dictionary as illustrated by Figure 5-4(1).

The extraction of color features used in this thesis is illustrated in Figure 5-4(2)-(6). For each extracted local patch, the RGB values of all pixels of the patch are transformed into the HSV space as in Figure 5-4(2), (3) and (4). In Figure 5-4(5), for the small local patch, average HSV values are computed respectively. By looking at which bin each of the HSV values falls into, the corresponding color word can be assigned to the local patch as in Figure 5-4(6).

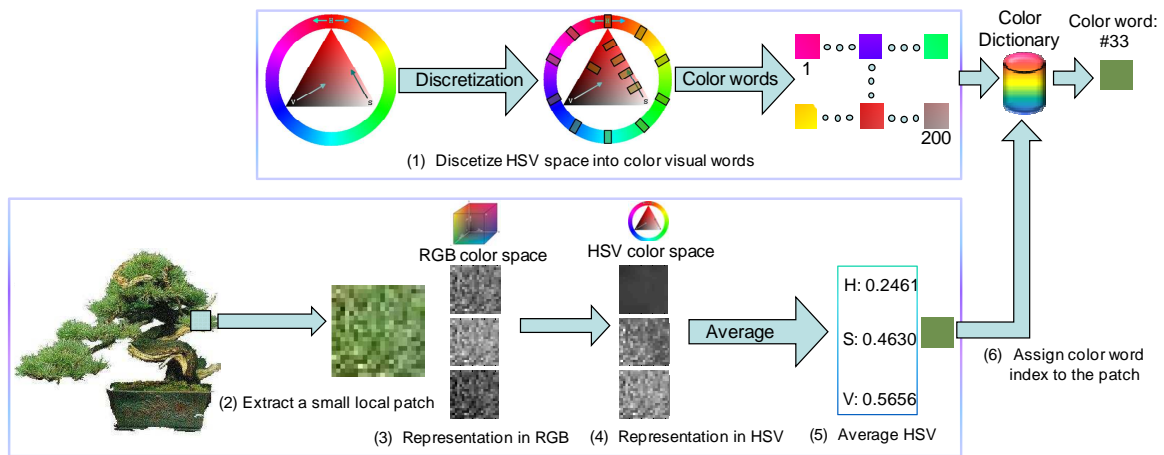


Figure 5-4: Illustration of the definition of color features. (1): The HSV color space is discretized. Each quantized color is regarded as a color word. All color words form the color dictionary. (2): For an image, a small local patch is extracted. (3) and (4): The RGB colors in the local patch are transformed to HSV space. (5) Average Hue, Saturation and Value are computed respectively on the local patch. (6) The color dictionary is referenced to assign the corresponding color word index to the average HSV of the patch.

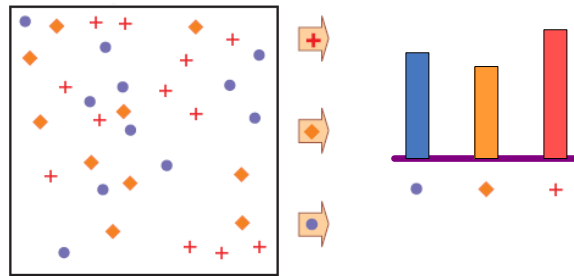
5.3 Matching Individual Decomposed Channels

5.3.1 Spatial Pyramid Matching of Local Features

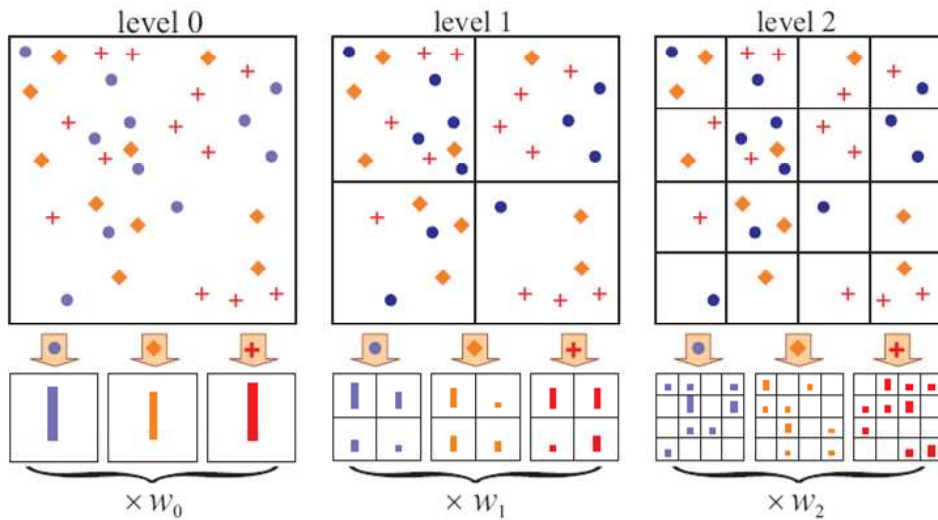
With the above feature extraction, an object image can be represented by the collection of local appearance features and color features, together with their corresponding spatial locations. Different images can be matched based on this representation. One method is to use the collection of features as a ‘bag-of-features’, *i.e.*, regarding the features as an orderless collection without considering the sequential or spatial information, and matching the global statistics of the set of orderless features. When the features used are invariant features, the ‘bag-of-features’ approach is an invariant matching scheme in theory. Figure 5-5(a) shows a toy example of the bag-of-features representation.

Another approach is to add order or spatial information in the matching scheme to achieve rough geometric constraints, which typically provide additional discriminative cues during matching. Some methods [49] simply append corresponding spatial locations to extracted features to get an augmented feature matching scheme. Lazebnik [70] proposes a “global” representation based on aggregating statistics of local features over fixed subregions and a kernel-based matching scheme that computes a rough geometric correspondence. The method is based on the efficient *pyramid matching* scheme proposed by Grauman and Darrell [48], with an extension of repeatedly subdividing an image and computing histograms of local features at increasingly fine resolution, termed ‘*spatial pyramid matching*’. While these spatially-augmented matching methods are non-invariant since absolute or relative spatial locations of features are used in matching, they are shown in practice to achieve better recognition performance in challenging data sets, especially when objects are imaged under their canonical poses and fill a majority of the image areas. Figure 5-5(b) give a schematic illustration of the spatial-pyramid representation.

As the toy example in Figure 5-5(b) shows, in spatial pyramid matching, an image is subdivided into several levels of resolution. At each level l , the image is divided by an $n_l \times n_l$ regular grid, where $n_l = 2^l$. Higher levels give more detailed subdivisions



(a) Bag-of-features representation.



(b) Spatial-pyramid representation.

Figure 5-5: Illustration of bag-of-features and spatial-pyramid representation. (a) In the bag-of-features representation, global statistics, such as occurrence frequencies, are derived from the ensemble of features in the entire image. (b) In the spatial-pyramid representation, an image is divided into three levels of resolution. For each resolution, statistics such as occurrence frequencies in each spatial cell can be calculated. Statistics from all levels collectively form the representation. Different levels of resolution can have different weights in the representation.

of the image. At each level, feature statistics such as the number of occurrences are calculated in each cell. When the image is matched against another image, the feature statistics in a certain cell at one level are only matched to the same cell of the other image. With this matching method, a higher level of resolution effectively puts more spatial constraints in feature matching, since features are considered as matched features only when they fall into the same small spatial cell. Typically, higher levels of resolution have larger weights in matching, *e.g.*, $\mathcal{W}_2 > \mathcal{W}_1 > \mathcal{W}_0$ in Figure 5-5(b). With this weight setting, the matching scheme places more emphasis on spatial correspondences, which is not addressed by the bag-of-features matching scheme. This gives the spatial pyramid matching a property of achieving rough geometric matching.

Considering its good performance and efficiency, ‘spatial pyramid matching’ is used in this thesis to match the local appearance features and color features in both contour and texture channels. More formally, spatial pyramid matching defines a matching kernel in the following manner:

Let X and Y be two sets of quantized visual word indices for extracted features of two images. For a given resolution l , a spatial histogram can be formed for each visual word, as illustrated in Figure 5-5(b). Each of these histograms can be matched to the histogram of the same visual word at the same resolution of another image, using the measure of *histogram intersection*. And the matching score of this resolution l is the sum of all *histogram intersection* scores for all visual words. Mathematically, the matching score for resolution l is:

$$\mathcal{I}^l(H_X^l, H_Y^l) = \sum_{n=1}^N \sum_{i=1}^{2^l} \min(H_{X_n}^l(i), H_{Y_n}^l(i))$$

where i is the index of histogram bin and l is the resolution level; n represents the index of visual words, and N is the size of visual vocabulary; $H_{X_n}^l$ and $H_{Y_n}^l$ denote the spatial histograms of X and Y at the resolution l for n th visual word. For multiple resolution $l = 0, 1, \dots, L - 1$, the weight associated with level l is set to $\frac{1}{2^{L-l}}$, giving larger weights for finer resolutions. The complete *pyramid matching* score is given by a weighted aggregation of matching scores from all resolutions:

$$\begin{aligned}
k^L(X, Y) &= \mathcal{I}^L + \sum_{l=0}^{L-1} \frac{1}{2^{L-l}} (\mathcal{I}^l - \mathcal{I}^{l+1}) \\
&= \frac{1}{2^L} \mathcal{I}^0 + \sum_{l=0}^{L-1} \frac{1}{2^{L-l+1}} \mathcal{I}^l.
\end{aligned} \tag{5.1}$$

5.3.2 Shape Matching with Robust Oriented Chamfer Distance

In the spatial pyramid matching scheme above, the SIFT descriptors capture local appearance and the matching pyramid applies rough geometric constraints to recognition by matching position-word distributions at different resolutions. Neither of the descriptors or the spatial pyramid matching scheme employs exact shape matching for recognition. However, for many classes, shape correspondence is also a salient visual cue for recognizing object instances. In this thesis, in addition to local appearance and rough geometric matching, we also leverage shape matching in the contour channel with robust chamfer matching. Usually chamfer matching performs poorly in cluttered images. This shortcoming of chamfer matching is, to a large extent, mitigated by the fact that salient contours in an image are decomposed into a separate, clean and sparse channel.

Chamfer matching works reasonably well for recognizing rigid or slightly deformable objects. In its original form, chamfer distance is a symmetrical similarity measure defined as follows:

Given two point sets $\mathcal{U} = \{\mathbf{u}_i\}_{i=1}^n$ and $\mathcal{V} = \{\mathbf{v}_j\}_{j=1}^m$, the chamfer distance function in one direction is the mean of the distances between each point $\mathbf{u}_i \in \mathcal{U}$ and its corresponding closest point in \mathcal{V} :

$$d_{chamfer}(\mathcal{U}, \mathcal{V}) = \frac{1}{n} \sum_{\mathbf{u}_i \in \mathcal{U}} \min_{\mathbf{v}_j \in \mathcal{V}} \|\mathbf{u}_i - \mathbf{v}_j\| \tag{5.2}$$

The chamfer distance in the other direction, $d_{chamfer}(\mathcal{V}, \mathcal{U})$ can be defined in a similar way. The symmetric chamfer distance is computed by averaging $d_{chamfer}(\mathcal{U}, \mathcal{V})$ and $d_{chamfer}(\mathcal{V}, \mathcal{U})$. The chamfer distance between two point sets can be efficiently

computed using distance transforms [16].

Also, researchers have found that limiting contributions from outliers and adding orientation in the matching procedure will greatly improve the performance. Similar to the spirit of [102, 107], the robust oriented chamfer distance used in this thesis is defined as follows:

$$d(X, Y) = \frac{1}{N_x} \sum_{x_i \in X} \max(\min_{y_j \in Y} \|x_i - y_j\|, \tau) + \lambda \frac{1}{N_x} \sum_{x_i \in X} \left(1 - e^{-\frac{\delta\theta_{x_i y_j}^2}{2\sigma_\theta^2}}\right) \quad (5.3)$$

where x_i and y_j are positions of edge pixels in image X and Y respectively, N_x is the number of edge pixels in image X , and $\delta\theta_{x_i y_j}$ is the difference between the orientation of pixel i and its closest match pixel j in image Y , where the orientation of pixels i and j is defined by the direction with maximum orientation energy as described in Section 3.2.1. The first term in Equation 5.3 is the truncated chamfer distance, and the second term is a Gaussian penalty for orientation differences. τ is the threshold for truncation, and λ is a relative weight for orientation match. To account for misalignment, X is slightly translated and rotated and the best match to Y is kept. To make a symmetrical distance, $d(Y, X)$ is also computed in a similar way and the average of $d(X, Y)$ and $d(Y, X)$ is taken as the similarity between X and Y . This process is schematically illustrated in Figure 5-6.

To compute kernel entries from pair-wise chamfer distances, another Gaussian form is used

$$K_{rchamfer}(X, Y) = e^{-\frac{[(d(X, Y) + d(Y, X))/2]^2}{2\sigma_k^2}} \quad (5.4)$$

where $K(X, Y)$ is the chamfer matching kernel entry.

In practice, it is observed that although the robust chamfer matching is somewhat crude in matching shapes, adding this channel is complementary to the rough geometric matching by spatial pyramid matching, and helps to achieve better recognition performance. It is expected that better shape modeling and matching schemes can be integrated in this framework and provide an extra performance boost.

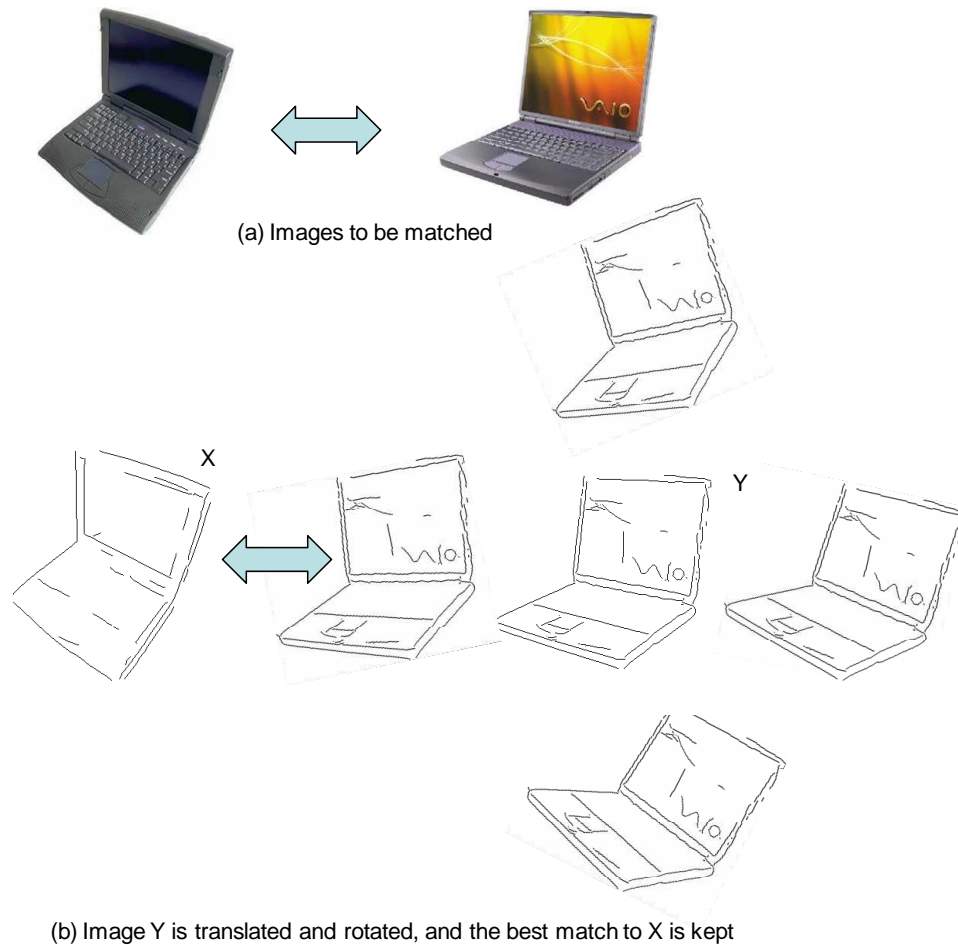


Figure 5-6: Illustration of computing the robust chamfer distance in one direction. Two laptop images to be matched are shown in (a). Robust chamfer distance is computed between the contour channels X and Y of the two images. To compute the robust chamfer distance from Y to X , Y is transformed by a series of translation and rotation, and the distances $d(X, Y)$ of these transformed images to X are computed with Equation 5.3. The best match of these distances $d(X, Y)$ is kept as the robust chamfer distance from Y to X .

5.4 Experiments

The different matching schemes used in this thesis are evaluated on a widely used dataset: Caltech-101 [37]. This dataset consists of images from 101 object categories, and contains from 31 to 800 images per category. Most images are of medium resolution, around 300×300 . Significant amounts of both inter-class variability and intra-class variation exist in the object classes of Caltech-101. Although many aspects of object recognition such as clutter, pose, and scale changes are lacking in this dataset, Caltech-101 is one of the most challenging datasets and the most widely adopted testbed for multi-class object categorization algorithms. Evaluation on this dataset helps to compare the relative performance of the proposed method to the state of the art.

Using the test convention for Caltech-101, the evaluation runs with different numbers of training samples per class, and tests on up to 50 images per class. For each experiment, the algorithm is evaluated with 10 runs with different randomly selected training and test samples, and the average of per-class recognition rates is reported. One caveat is that the class of Faces in Caltech-101 is much larger than other classes. To avoid image size artifacts, in our experiments face images are scaled down to around 300×300 while preserving aspect ratio.

Another parameter to be set is the maximum level of resolution L in the spatial pyramid matching scheme. In their original work [70], Lazebnik *et al.* tested on $L = \{0, 1, 2\}$ and found $L = 2$ achieves better recognition results than coarser resolutions. In the experiments in this thesis, for a fair comparison, L is also set to 2 as in [70]. Note, however, Bosch *et al.* [18] reported that using $L = 3$ gives additional performance improvements.

With the matching kernels in the decomposed channels, a one-versus-one multi-class Support Vector Machine (SVM) [27] is trained and used for classification. For each matching scheme, the parameters such as the image patch size and the vocabulary size are observed to have different impact on the classification results. The following sections empirically demonstrates the effects of these parameters.

5.4.1 Spatial Pyramid Matching of Local Appearance Features

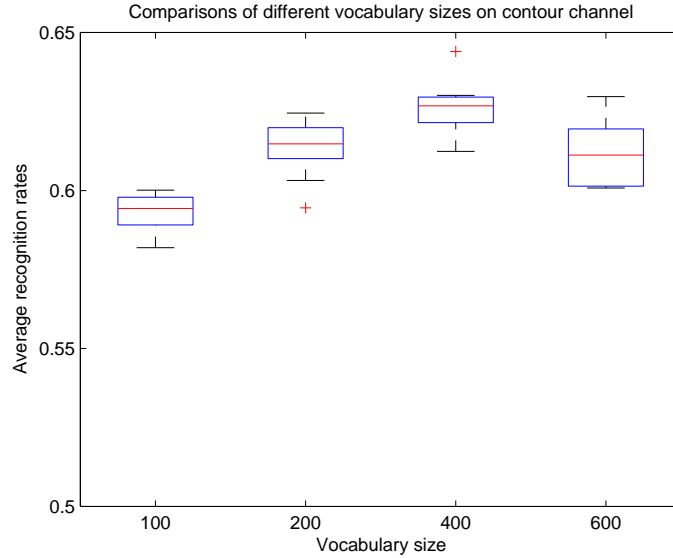
Effect of Visual Vocabulary Size of SIFT Features

The first experiment is used to evaluate the effect that different visual vocabulary sizes exhibit on recognition performance. A set of 505 training images is formed by randomly selecting 5 images from each class of the Caltech-101. In this experiment, SIFT patch size is fixed to 50 for both contour and texture channels. Visual vocabularies are generated by clustering SIFT descriptors from the 505 training images. Descriptors from each channel are clustered into 100, 200, 400 and 600 visual words respectively. The classification performance of contour and texture channels for these vocabulary sizes are shown in Table 5.1 for 30 training samples per class. Figure 5-7 draws the corresponding box-plots.

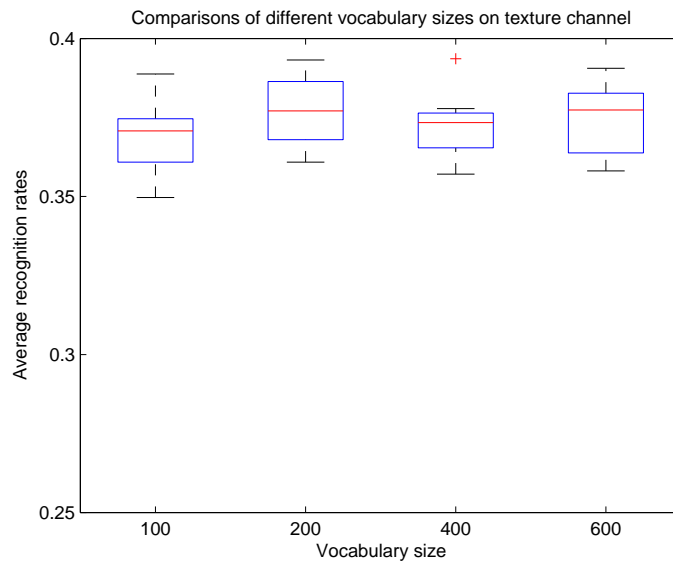
Overall, as seen from Table 5.1 and Figure 5-7, with respect to changes of visual vocabulary size, recognition performance only fluctuates within a small range. Mid-sized visual vocabulary gives slightly better improvement in recognition performance compared with too coarse and too fine vocabularies, especially in the contour channel as in Figure 5-7(a). The reason could be that, as stated in Section 5.2.2, too coarse vocabularies over-stress tolerance of intra-class variations, and too fine vocabularies are sensitive to intra-class variations and lose generalization capability.

Vocab. Size	100	200	400	600
Reco. rate (contour)	59.27% (0.64)	61.37% (0.94)	62.63% (0.84)	61.16% (1.03)
Reco. rate (texture)	36.96% (1.13)	37.68% (1.09)	37.30% (0.97)	37.53% (1.12)

Table 5.1: Comparison of average per-class recognition rates of appearance matching on Caltech-101 in contour and texture channels respectively, with different visual vocabulary sizes. Numbers in parenthesis are standard deviation.



(a) Performance comparisons of different vocabulary sizes on appearance matching in the contour channel.



(b) Performance comparisons of different vocabulary sizes on appearance matching in the texture channel.

Figure 5-7: Box plots of performance comparisons of different vocabulary sizes on appearance matching in contour and texture channels. In each case, the box draws the first quartile, median and third quartile of recognition rates, the whiskers show the extent of the non-outlier recognition rates, and the outliers (if any) are marked with red cross. Vocabulary size doesn't have a great impact on the recognition performance, although medium-sized vocabularies are slightly better.

Effect of Size of Densely Sampled Features

This experiment evaluates the effect of local patch size of the densely sampled features. To extract features, we use dense sampling on a regular grid with spacing of 8 pixels. For each position on the grid, a patch of $S_f \times S_f$ pixels is extracted, where S_f is the size of the extracted local patches. 72-dimensional SIFT descriptors are computed as in Section 5.2.2. In this evaluation, the size of the visual vocabulary is set to 400 for the contour channel, since this vocabulary size is slightly better than other sizes as shown in Table 5.1. Similarly, a visual vocabulary of size 200 is used for the texture channel.

The classification performance of contour and texture channels is evaluated separately. The second row of Table 5.2 lists the average recognition rates on Caltech-101 on the contour channel alone with different patches sizes, for 30 training samples per class. And the average recognition rates for the texture channel alone are listed in the third row of Table 5.2. The corresponding box-plots are shown in Figure 5-8.

It is observed that when the patch size is too small or too large, the recognition performance decreases compared with those of medium-sized patches. For small patches, the reason is postulated to be that small patches are not rich enough in information, since only primitive elements such as a segment of line or curve exists in small patches and these primitive elements or the collection of these primitive small patches are not distinctive enough to differentiate various classes of objects, especially when the decomposed channels are sparser than the original images. That is to say, small patches are not sufficient in capturing inter-class differences. On the contrary, for large patches, the reason for the degraded performance could be that large patches are not able to accommodate intra-class variations. It is more possible for large patches to enclose appearance information in large image regions that are specific to a particular object instance of a class but not repeatable within other object instances of the same class. Thus to a great extent, using large patches loses the generalization capability.

Table 5.3 lists the performance of some of the current best methods on the Caltech-

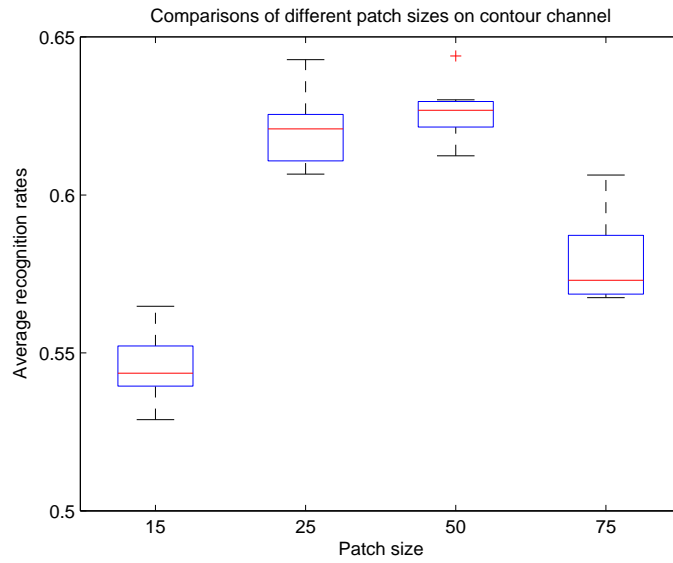
Patch size S_f	15	25	50	75
Reco. rate (contour)	54.54% (1.07)	62.06% (1.20)	62.63% (0.84)	57.98% (1.35)
Reco. rate (texture)	28.65% (0.71)	31.41% (1.02)	37.68% (1.09)	30.88% (1.19)

Table 5.2: Average per-class recognition rates on Caltech-101 with SIFT descriptors in contour and texture channels respectively. Different patch sizes are tested. Numbers in parenthesis are standard deviation.

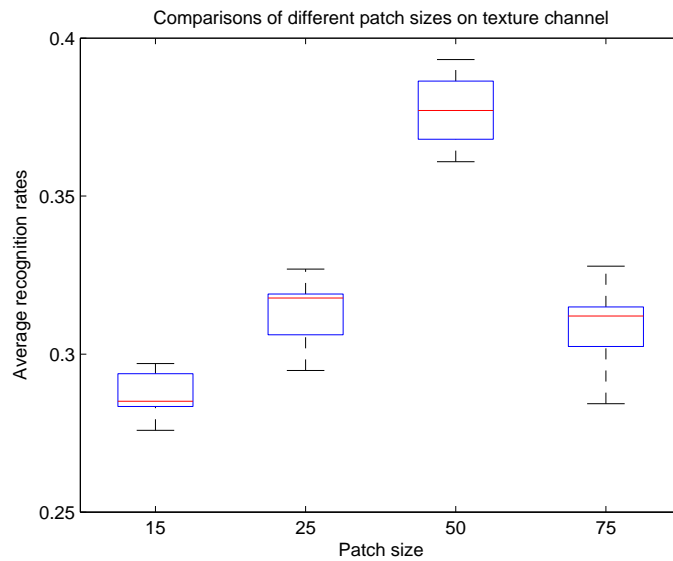
101 dataset. It is noticeable that for the ‘recognition-through-decomposition-and-fusion’ scheme in this thesis, even with the sparse contour channel alone, the average recognition rate for 30 training samples per class is already achieving relatively comparable performance to many previous best approaches on this dataset. Especially when compared with the original spatial pyramid matching with descriptors on original image patches [70], there is only a 1.97% decrease in performance when only the contour channel is used. This suggests that salient contours play an important role and are the dominant visual information in recognizing the objects in Caltech-101.

Training sample	Contour only	[51]	[117]	[70]	[113]	[88]
15	56.55%(0.46)	59%	59.05%(0.56)	56.4%	44%	51%
30	62.63%(0.84)	67.6%(1.4)	66.23%(0.48)	64.6%(0.8)	63	56%

Table 5.3: Comparison of average per-class recognition rates of appearance matching on Caltech-101 of some current best methods. Numbers in parenthesis are standard deviation.



(a) Performance comparisons of different patch sizes on appearance matching in the contour channel.



(b) Performance comparisons of different patch sizes on appearance matching in the texture channel.

Figure 5-8: Box plots of performance comparisons of different patch sizes on appearance matching in contour and texture channels. In each case, the box draws the first quartile, median and third quartile of recognition rates, the whiskers show the extent of the non-outlier recognition rates, and the outliers (if any) are marked with red cross. It is observed that medium-sized patches give better recognition performance.

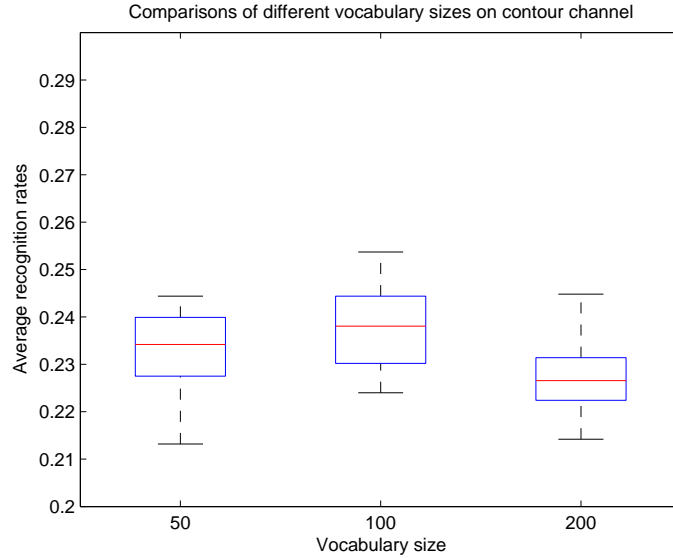
5.4.2 Spatial Pyramid Matching of Local Color Features

Effect of Color Quantization

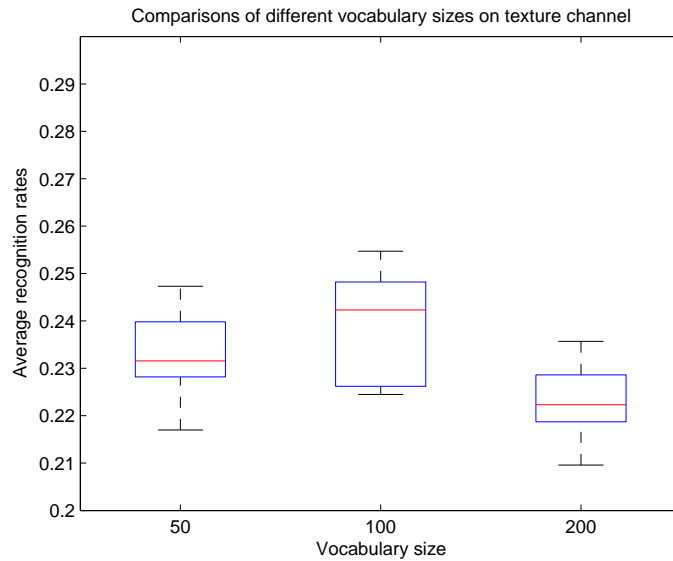
In the color matching scheme used in this thesis, the Hue-Saturation-Value color space is quantized into $h_c \times s_c \times v_c$ color words, with Hue discretized into h_c bins, Saturation into s_c bins and Value into v_c bins. The effects of the granularity of HSV space quantization are tested on three settings: $\{h_c=5, s_c=5, v_c=2\}$, $\{h_c=5, s_c=5, v_c=4\}$ and $\{h_c=10, s_c=10, v_c=2\}$, which give color vocabulary size of 50, 100 and 200 respectively. These quantizations are chosen arbitrarily, with different levels of discretization of the HSV space. Finer or coarser quantizations are expected to perform worse, for reasons similar to those described in Section 5.4.1. For 30 training samples per class, the performance comparisons are shown in Table 5.4 and Figure 5-9. Vocabulary size, or equivalently, the granularity of color space quantization, does not have a significant impact on the recognition performance, with medium-sized vocabulary marginally better.

Vocab. Size	50	100	200
Reco. rate (contour)	23.23% (0.98)	23.82% (1.01)	22.69% (0.89)
Reco. rate (texture)	23.26% (0.99)	23.95% (1.00)	22.36% (0.81)

Table 5.4: Comparison of average per-class recognition rates of color matching on Caltech-101 in contour and texture channels respectively, with different visual vocabulary sizes. Numbers in parenthesis are standard deviation.



(a) Performance comparisons of different vocabulary sizes on color matching in the contour channel.



(b) Performance comparisons of different vocabulary sizes on color matching in the texture channel.

Figure 5-9: Box plots of performance comparisons of different vocabulary sizes on color matching in contour and texture channels. In each case, the box draws the first quartile, median and third quartile of recognition rates, the whiskers show the extent of the non-outlier recognition rates, and the outliers (if any) are marked with red cross. Vocabulary size, or equivalently, the granularity of HSV quantization, doesn't have a great impact on the recognition performance.

5.4.3 Robust Chamfer Matching on Contour Channels

In principle, the parameters of the robust chamfer matching can be learned by cross-validation on the training data. In the current implementation in this thesis, the robust chamfer matching on the contour channel is carried out with heuristically set parameters as follows. The parameters of τ and λ in the robust chamfer matching are both set to 40. $\tau = 40$ simply means outliers that are 40 pixels away from their closest matches are limited to a Euclidean distance of 40. Making $\lambda = \tau$ implies that the contribution of a complete mis-match in orientation is commensurate to that of a distance outlier. σ_θ in Equation 5.3 is set to 20 degrees, and σ_k in Equation 5.4 is 20. A series of different numbers of training samples per class $\{1, 5, 10, 15, 20, 25, 30\}$ is used for evaluation. For comparison, spatial pyramid matching of SIFT features and color features on the contour and texture channels are also evaluated for these training sample numbers. Table 5.5 and Figure 5-10 give the comparisons of these matching schemes introduced in this Chapter. It can be seen that, when individual visual cues are used separately, the spatial pyramid matching with SIFT descriptors and the robust chamfer matching on the contour channel have significantly better recognition performance. And color matchings are the worst among the used matching schemes. This corroborates that the observation that the shape contour, compared with texture and color, is the most prominent visual information to distinguish object classes in the dataet of Caltech-101.

Training sample	1	5	10	15	20	25	30
Contour-Chamfer	18.64	41.39	49.56	53.30	55.60	57.46	58.91
Contour-SIFT	22.87	44.12	52.72	56.55	58.99	61.04	62.63
Texture-SIFT	8.92	20.80	27.42	31.41	34.06	36.16	37.68
Contour-Color	2.19	10.98	15.23	18.33	20.42	21.83	23.36
Texture-Color	2.48	11.28	16.05	19.03	20.95	22.65	23.95

Table 5.5: Comparison of average per-class recognition rates of different kernels. Numbers are in percentile.

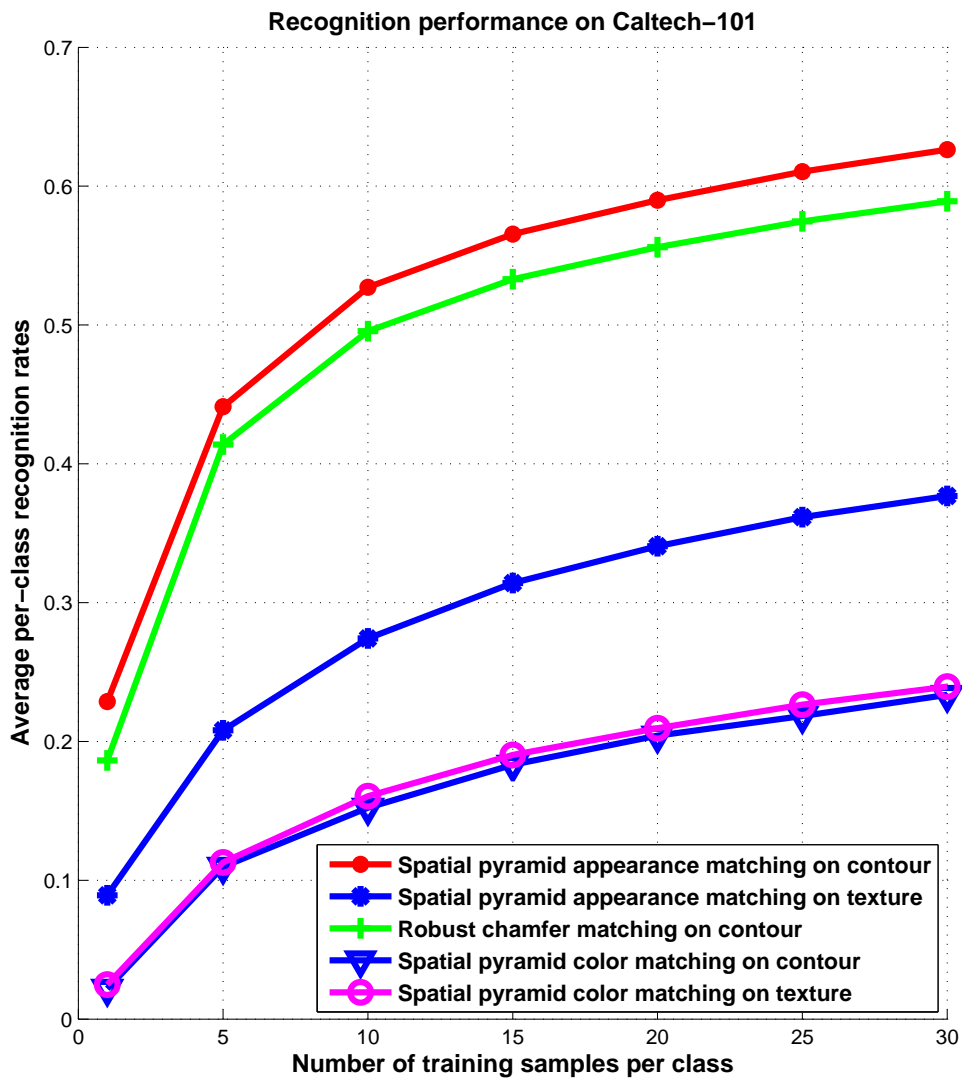


Figure 5-10: Comparison of average per-class recognition rates of different kernels.

5.5 Summary

In many psychophysical and physiological studies, contour and texture are shown to be processed in a dissociative manner in human visual perception. Based on this, in this chapter, suitable features and matching schemes are introduced for each of the decomposed contour and texture channels. Feature points are extracted by dense sampling on each channel. Appearance in the contour and texture channels are captured with modified SIFT descriptors of the sampled feature points. Colors of the sampled feature points are represented in a quantized HSV color space. Weak geometric matching is incorporated with the spatial pyramid matching of the sampled features. As a complementary scheme to the weak geometric matching, since salient contours are decomposed into a sparse channel, a robust chamfer matching is used to apply strong geometric matching in the contour channel.

Empirical evaluation of some parameters, such as the size of sampled patches and the size of visual vocabularies, of various matching schemes is carried out. It is shown that medium-sized patches perform better than smaller or larger patches. The reason is conjectured to be that small patches are not sufficient in capturing inter-class differences, and large patches are not able to accommodate intra-class variations. The size of visual vocabulary is observed to have no significant impact on the recognition performance, for both the appearance and color matching. It is expected that the matching schemes would perform well in a large range of vocabulary sizes.

An interesting fact is noticed that, even with the decomposed contour channels being much sparser than the original images, the spatial pyramid matching and the robust chamfer matching on the contour channel are shown to achieve recognition performance comparable to many state-of-the-art methods which operate on the original images. This suggests that in many object classes the shape contour is the most prominent visual information for object recognition.

In the next chapter, the various matching schemes on the decomposed contour and texture channels introduced in this chapter will be adaptively combined, as a counterpart to the integration process of human visual perception. The combination

will be shown to be able to adapt to object class pairs, selecting better visual cues for distinguishing each pair of classes and leading to improved object recognition performance.

Chapter 6

Adaptive Multiple Visual Information Combination

As discussed in Chapter 1, natural images of objects generally contain a great amount of rich visual information about the objects of interest and their backgrounds. Human observers can effortlessly and efficiently disassociate visual information such as salient contour, characteristic texture and prominent color distinctions, and recombine this information in a joint effort to recognize different objects. As an emulation to human perception, this thesis proposes the coupled Conditional Random Field model in Chapter 4 to decompose contour and texture in natural images. In Chapter 5, proper matching schemes are introduced to match various visual cues, *i.e.*, contour, texture and color, with each of the matching schemes addressing a different perceptual aspect of object recognition. The developed decomposition and matching schemes naturally enable methods of visual information fusion to fully leverage various visual cues and integrate them into a complex whole. To this end, this chapter studies a principled method of adaptively combining the decomposed contour and texture channels, and demonstrates the effectiveness of the approach of “recognition-through-decomposition-and-fusion” with recognition experiments on a challenging dataset.

6.1 Types of Information Fusion Schemes

Traditionally, pattern recognition systems are designed to use one particular classification procedure to estimate the class of a given pattern. The last decade has seen that combining multiple classifiers can be an efficient technique for improving classification performance. If the combination can take advantage of the strength of the constituent individual classifiers and avoid their respective weakness, the overall classification accuracy is expected to be boosted. The proposed “recognition-through-decomposition-and-fusion” scheme, with different classifiers for various decomposed visual cues, falls well into the field of multiple classifier ensemble.

Generally speaking, information fusion can be implemented on several levels of a classification ensemble:

1. **Feature level:** When there are multiple types of features in images, features can be combined in an integrated feature representation. For example, face and gait features can be normalized and directly concatenated into a synthetic feature vector for human recognition [119]. In [39], features such as curves, blobs and corners are extracted from images, and this heterogeneous feature representation is fed into a star geometric model for object categorization.
2. **Distance function level:** At a higher level than the feature level, distance functions can be learned to optimally fuse various sources of information. In [43, 44], Frome *et al.* used the maximal-margin criterion to learn linearly combined distance functions for shape features at two different scales and a color feature. Zhang *et al.* [117] designed distance functions to combine geometric blur features and texture features, and converted the distance matrix to a kernel matrix and applied multiclass SVM for object recognition.
3. **Kernel level:** Multiple sources of information can also be integrated via kernel combination. Diego *et al.* [32, 85] studied various non-linear weighting schemes, such as using absolute values, squared quantity and MaxMin, to combine multiple kernels. A boosting algorithm was proposed by Bennett *et al.*

[8] to construct a heterogeneous kernel from a large library of kernel matrices. Cristianini *et al.* [30] developed a kernel alignment theory to linearly combine multiple kernels to best align with a pre-defined ideal kernel.

4. **Classifier output level:** Much work has been done in integrating the classification outputs from a set of classifiers. Popular ways to combine classifier outputs include using max/min, median and majority vote rules [65], encoding classifier outputs with error-correcting codes [33], and boosting [3].

While these combination schemes can all be explored for visual information combination, kernel-level combination is of particular interest to the proposed framework of combining multiple visual cues for object recognition. In this thesis, each of the matching kernels is designed to leverage a different perceptual characteristic of object recognition. By combining them at the kernel level, adaptive weights can be learned for different kernels. These adaptive weights are good indicators of the relative importance and effectiveness of different visual cues for recognizing various object classes. This well mirrors the concept of dissociation and integration of human perception discussed in Section 1.1.

6.2 Adaptive Information Fusion by Kernel Alignment

On the kernel-level fusion, the kernel alignment theory proposed by Cristianini *et al.* [30] provides a principled way of learning the optimally combined kernel. The kernel alignment method first defines an alignment score, *i.e.*, the goodness of a kernel matrix with respect to another kernel matrix. More formally:

Let K_1 and K_2 denote two kernel matrices. The kernel alignment score is defined based on the Frobenius inner product, which is the component-wise inner product of two matrices regarding them as vectors, as follows:

$$A(K_1, K_2) = \frac{\langle K_1, K_2 \rangle_F}{\sqrt{\langle K_1, K_1 \rangle_F \langle K_2, K_2 \rangle_F}} \quad (6.1)$$

where $A(K_1, K_2)$ is the kernel alignment score between kernels K_1 and K_2 , $\langle K_a, K_b \rangle_F = \text{trace}(K_a^T K_b) = \sum_{i,j=1}^n K_a(\mathbf{x}_i, \mathbf{x}_j) K_b(\mathbf{x}_i, \mathbf{x}_j)$ is the Frobenius inner product of kernel matrices K_a and K_b , and $K_a(\mathbf{x}_i, \mathbf{x}_j)$ and $K_b(\mathbf{x}_i, \mathbf{x}_j)$ are the kernel matrix entries corresponding to object instances of \mathbf{x}_i and \mathbf{x}_j respectively. The kernel alignment score is simply the normalized distance when regarding the two kernels as vectors. The alignment score is larger when corresponding entries of the two kernels are more comparable, *i.e.*, when the two kernels are more aligned.

Now consider the task of object classification. As a basic building block, let us consider a two-class (binary) classification problem. Let $D = \{\mathbf{x}_n, l_n : n = 1 \dots N\}$ denote a labeled dataset, where \mathbf{x}_n represents n th data sample and $l_n \in \{+1, -1\}$ is the corresponding class label, and N is the number of samples in the dataset. On this labeled dataset, an *ideal kernel* can be defined in the following way:

$$K_{ideal}(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} v_s & l_i = l_j \\ v_{ns} & l_i \neq l_j \end{cases} \quad (6.2)$$

That is, when the two data samples $\mathbf{x}_i, \mathbf{x}_j$ belong to the same class, the corresponding entry in the *ideal kernel* takes a value of v_s ; otherwise, the entry takes a value of v_{ns} . Typically v_s is much larger than v_{ns} . In some embodiment of the *ideal kernel* such as the one used in [73], these values can be defined as

$$K_{ideal}(\mathbf{x}_i, \mathbf{x}_j) = l_i \cdot l_j = \begin{cases} +1 & l_i = l_j \\ -1 & l_i \neq l_j \end{cases} \quad (6.3)$$

Or the value of v_s and v_{ns} can depend on the dynamic ranges of constituent kernels. For example, v_s can be 1 for diagonal entries and 0.2 for off-diagonal entries, and v_{ns} can be 0.1, which are the values used in our implementation. These values are comparable to the entries of the spatial pyramid matching kernels and the robust chamfer matching kernel used in this thesis.

Now, denote $K_{collection} = \{K_m : m = 1 \dots M\}$ as the set of kernels derived from different visual matching schemes on the two-class dataset D , and $K_{comb} = \sum_{m=1}^M w_m K_m$ as the linearly combined kernel with w_m as the weight assigned to kernel K_m . This set

of linear weights $\mathbf{w} = \{w_m : m = 1..M\}$ essentially captures the relative importance of each individual visual cue for discriminating the positive and negative classes. Take the examples in Section 1.1 for instance. To classify *beaver* versus *emu*, the matching kernels derived from the decomposed contour channel could have larger weights since the differences between these two classes mainly lie in contour and shape; and for *laptop* versus *inline skate*, the weight of the matching kernel from the texture channel could be relatively larger to emphasize the difference in the keyboard texture of laptops.

By changing these weights, the combined kernel can be tuned to approximate an optimal discriminative kernel where intra-class entries are large and inter-class entries are small. This is equivalent to moving data samples of the same class closer to each other and in the meantime separating data samples from different classes further away in an induced high-dimensional feature space. One choice of the optimal discriminative kernel is the *ideal kernel* defined above. It is logical to expect that, when the combined kernel is tuned to approximate the corresponding *ideal kernel*, best separation and maximum discriminability can be reached. Hence, to achieve the optimal combined kernel, one can use the kernel alignment theory to best align the combined kernel K_{comb} with the *idea kernel* K_{ideal} . More specifically, this optimization problem can be formulated as follows:

$$\begin{aligned}
& \max_{\mathbf{w}} && A(K_{comb}, K_{ideal}) \\
& \text{subject to} && K_{comb} = \sum_{m=1}^M w_m K_m, \\
& && \text{trace}(K_{comb}) = 1, \\
& && w_m \geq 0, m = 1..M.
\end{aligned} \tag{6.4}$$

The above optimization problem maximizes the alignment score between the combined kernel K_{comb} and the ideal kernel K_{ideal} , with respect to the combination weights \mathbf{w} . The first constraint requires K_{comb} to be a linear combination of constituent kernels. The second condition is a constraint on scale, without which infinite solutions of \mathbf{w} up to a scale factor will exist. The third constraint means no constituent kernel has a negative weight.

The optimization problem in Equation 6.4 belongs to convex optimization, and Lanckriet *et al.* [69] show that it can be solved with semidefinite programming. Semidefinite programming in many cases is not computationally efficient. Hoi *et al.* [56] derived an equivalent algorithm by translating the optimization problem in Equation 6.4 into a Quadratic Programming problem, which can be solved very efficiently. The equivalent Quadratic Programming problem is formulated as follows:

$$\begin{aligned}
& \min_{\mathbf{w}} && \mathbf{w}^T V^T V \mathbf{w} \\
& \text{subject to} && \text{vec}(K_{ideal})^T V \mathbf{w} = 1, \\
& && w_m \geq 0, m = 1 \dots M.
\end{aligned} \tag{6.5}$$

where $V = [\text{vec}(K_1) \text{vec}(K_2) \dots \text{vec}(K_M)]$ and $\text{vec}(K)$ is the column vectorization of matrix K . Solving the optimization problem 6.5 gives the set of weights \mathbf{w} that best align the linearly combined kernel K_{comb} with the corresponding *ideal kernel* on the two-class dataset D .

For multi-class categorization, when multi-class categorization schemes are composed of multiple binary sub-classifiers, the extension of adaptive kernel combination to multi-class categorization is straightforward, that is, to simply learn adaptive combination weights for each of the constituent binary sub-classifiers. There are two popular ways of using a set of binary classifiers for multi-class classification: one-versus-one and one-versus-the-rest. For N classes, one-versus-one schemes construct one binary sub-classifier for every pair of distinct classes. There are $\frac{N(N-1)}{2}$ sub-classifiers all together. For a test data t , t is classified by each of the $\frac{N(N-1)}{2}$ sub-classifiers, and one-versus-one schemes assign t to the class with the largest number of votes. One-versus-the-rest schemes construct N binary sub-classifiers. The i th classifier is trained to distinguish class i from all other $N - 1$ classes. A test data t is classified with each of the N sub-classifiers, and t is assigned to the class with largest decision value, which, for example, can be the distance to decision boundary. One-versus-one multi-class categorization is used in this thesis where adaptive kernel combination weights can be learned for each pair of classes in this scheme. This class-pair specific adaptation is consistent with the concept illustrated by Figure 1.1.

6.3 Experiments

The dataset of Caltech-101 [37] is used for evaluation of multiple visual information fusion. As in Chapter 5, the evaluation runs with different numbers of training samples per class, and tests on up to 50 images per class. For each experiment, the algorithm is evaluated with 10 runs with different randomly selected training and test samples, and the average of per-class recognition rates is reported.

The ideal kernel is defined with v_s as 1 for diagonal entries and 0.2 for off-diagonal entries, and v_{ns} as 0.1 in Equation 6.2. For each class pair, the optimal adaptive combination weights \mathbf{w} are learned with Equation 6.5, by aligning the combined kernel with the ideal kernel. With the adaptively combined kernel, a one-versus-one multi-class Support Vector Machine is trained and used for classification.

6.3.1 Combining Multiple Scales

The first experiment is carried out to combine multiple scales to evaluate the scale factor in Caltech-101. While the feature extraction scheme in Section 5.2.1 is not invariant to similarity or affine transformation, some level of scale adaptation can be achieved at the stage of visual cue combination in the proposed framework. As can be seen from Table 5.2 and Figure 5-8, the densely sampled features with patch size of 25 and 50 have comparable recognition performance, especially for the contour channel. This suggests that, while it is widely accepted that the Caltech-101 dataset does not have much scale variation, the Caltech-101 dataset still displays some extent of scale changes. That is, objects of some classes are complex and fill a large portion of their images, while objects of some classes are relatively small in the field of view. In the first case, larger patches are expected to capture well the distinct appearance of objects, and in the latter case, smaller patches are suitable to describe the objects of interest. The following experiment demonstrates this level of inter-class scale variability in recognizing objects in Caltech-101.

Denote the spatial pyramid matching kernels with SIFT patch size 25 and 50 as $K_{cSIFT25}$ and $K_{cSIFT50}$ (contour channel) and $K_{tSIFT25}$ and $K_{tSIFT50}$ (texture chan-

Kernel used	$K_{cSIFT25}$	$K_{cSIFT50}$	$K_{SIFT_{avg}}$	K_{comb}
Reco. rate (contour)	62.06% (1.20)	62.63% (0.84)	63.39% (0.64)	64.07% (0.92)
Reco. rate (texture)	31.41% (1.02)	37.68% (1.09)	37.52% (0.66)	37.69% (0.90)

Table 6.1: Comparison of recognition performance on Caltech-101 with kernels $K_{cSIFT25}$ and $K_{cSIFT50}$ (spatial pyramid matching kernels for patch size 25 and 50 respectively), their average combination $K_{SIFT_{avg}}$ and adaptive combination K_{comb} , for 30 training samples per class. The second row are average per-class recognition rates on the contour channel. The third are average per-class recognition rates on the texture channel. Numbers in parenthesis are standard deviation.

nel) respectively. These kernels of different feature scales are combined using kernel alignment on the contour and texture channels respectively. That is, for example, a combined kernel is defined on the contour channel by

$$K_{SIFT_{comb}} = w_{25}K_{cSIFT25} + w_{50}K_{cSIFT50} \quad (6.6)$$

where w_{25} and w_{50} are linear combination weights. Equation 6.5 solves for the optimal combination weights, and for 30 training samples per class, the recognition performance of each individual kernel $K_{cSIFT25}$ and $K_{cSIFT50}$, and the combined kernel $K_{SIFT_{comb}}$ is shown in Table 6.1. A non-adaptive combined kernel $K_{SIFT_{avg}}$ by simply averaging $K_{cSIFT25}$ and $K_{cSIFT50}$ is also implemented and compared:

$$K_{SIFT_{avg}} = \frac{K_{cSIFT25} + K_{cSIFT50}}{2} \quad (6.7)$$

In Table 6.1, it can be seen that combining kernels on multiple scales helps to improve the recognition performance by 1.44% in the contour channel. To compare the relative effectiveness of patch size 25 and 50 on different classes, the characteristic w_{25} and w_{50} for each class are computed as follows: since one-versus-one SVM is used for classification, for an object class c , there are 100 binary sub-classifiers to discriminate object class c against the rest of classes of objects in Caltech-101. w_{25} and w_{50} are learned for each of these 100 sub-classifiers for class c , and the characteristic

w_{25} and w_{50} for class c are defined as the 5th smallest w_{25} and w_{50} , denoted as \hat{w}_{25} and \hat{w}_{50} respectively, among these 100 sub-classifiers. \hat{w}_{25} and \hat{w}_{50} can act as a robust statistical measurement of the relative importance and effectiveness of $K_{cSIFT25}$ and $K_{cSIFT50}$ for recognizing class c . Larger \hat{w}_{25} means matching with small patch size 25 is more effective for class c , and smaller \hat{w}_{25} indicates matching with larger patches is more effective instead. Table 6.2 shows the classes with 10 largest \hat{w}_{25} and the classes with 10 smallest \hat{w}_{25} .

In the left three columns of Table 6.2, for some classes with large \hat{w}_{25} , the objects of interest are either small in the field of view (e.g., *leopard*, *mayfly*), or narrow and long (e.g., *minaret*, *wrench*, *saxophone*, *crocodile*, *octopus*, *stapler*). So it is reasonable that these objects are better distinguished by small patches of size 25. For some other classes with large \hat{w}_{25} (e.g., *stop sign*, *menorah*), the reason could be that these classes are relatively simple yet distinct enough that a collection of small patches of size 25 is able to well represent the classes. For the classes with small \hat{w}_{25} as shown in the right three columns of Table 6.2, their objects typically fill the images, and most of them are complex. It is conjectured that small patches are not distinctive enough for these classes and the recognition has to resort to larger patches or the combination of small and large patch features. In this sense, some level of inter-class scale variation does exist in the dataset of Caltech-101, and scale adaptation is able to explore the performance margin provided by adaptive kernel combination.

As shown in the third row of Table 6.1, no similar effect is observed in the texture channel. The reason could be that, for the texture channel, matching with large patches of size 50 dominates over small patches of size 25, *i.e.*, $K_{tSIFT50}$ has much better recognition performance than $K_{tSIFT25}$, as shown in Figure 5-8(b), and patches of size 25 and 50 are not complementary to each other. The texture channels of objects in Caltech-101 can be well represented by patches of size 50. Hence $K_{SIFTcomb}$ has the same recognition performance as $K_{tSIFT50}$ for the texture channel.







Class	\hat{w}_{25}	Image	Class	\hat{w}_{25}	Image
Leopard	1.0000		Camera	0.2817	
Minaret	0.9216		Wheelchair	0.2864	
Wrench	0.8600		Face	0.3226	
Stop sign	0.8155		Dalmatian	0.3286	
Mayfly	0.8092		Ferry	0.3418	
Saxophone	0.7950		Buddha	0.3441	
Menorah	0.7910		Inline skate	0.3484	
Crocodile	0.7751		Watch	0.3523	
Octopus	0.7382		Panda	0.3656	
Stapler	0.7131		Rooster	0.3700	

Table 6.2: The left three columns are the 10 classes where small patches are of more importance in recognition. The first column is the class name, the second is the weight of the kernel $K_{cSIFT_{25}}$ with patches of size 25, and the third column is exemplar images of corresponding classes. The right three columns show the 10 classes where large patches of size 50 play more important roles in recognition. See text for details.

6.3.2 Complementarity of Weak and Strong Shape Matching in Contour Channel

As discussed in Section 5.3.2, the SIFT descriptors capture local appearance and the spatial matching pyramid applies weak geometric constraints to recognition, and in the meantime the decomposition also enables robust chamfer matching to employ strong shape cues for recognition. The two matching schemes address different perceptual aspects of object recognition. The experiment in this section demonstrates the complementarity of SIFT-spatial-pyramid matching and robust chamfer matching.

In the contour channel, weak geometric matching is implemented with the SIFT-spatial-pyramid-matching kernels $K_{cSIFT25}$ and $K_{cSIFT50}$, and strong geometric matching is implemented with the robust chamfer matching kernel $K_{rchamfer}$ defined by Equation 5.4 in Chapter 5. The integrated kernel $K_{contour}$ for the contour channel is defined by adaptively combining $K_{cSIFT25}$, and $K_{cSIFT50}$ and $K_{rchamfer}$:

$$K_{contour} = w_{cSIFT25}K_{cSIFT25} + w_{cSIFT50}K_{cSIFT50} + w_{rchamfer}K_{rchamfer} \quad (6.8)$$

Table 6.3 and Figure 6-1 compare the recognition performance of the combined contour channel kernel $K_{contour}$ (6.8), combined multiscale SIFT-spatial-pyramid-matching kernel $K_{SIFTcomb}$ (6.6) and robust chamfer matching kernel $K_{rchamfer}$ (5.4). For various numbers of training samples per class $\{1, 5, 10, 15, 20, 25, 30\}$, the combined contour kernel $K_{contour}$ consistently gives higher recognition performance. This suggests that weak geometric matching with the spatial pyramid kernels and strong geometric matching with the robust chamfer kernel are complementary to each other, and the kernel adaptation scheme is able to explore their complementarity for better recognition.

Training sample	1	5	10	15	20	25	30
$K_{contour}$	22.99	48.34	57.10	60.75	63.29	65.29	67.08
$K_{SIFTcomb}$	22.74	45.61	54.20	57.79	59.90	62.37	64.07
$K_{rchamfer}$	18.64	41.39	49.56	53.30	55.60	57.46	58.91

Table 6.3: Comparison of average per-class recognition rates of the combined SIFT-spatial-pyramid-matching kernel $K_{SIFTcomb}$, the robust chamfer matching kernel $K_{rchamfer}$ and the combined contour kernel $K_{contour}$ which adaptively integrates $K_{cSIFT25}$, and $K_{cSIFT50}$ and $K_{rchamfer}$. Numbers are in percentile.

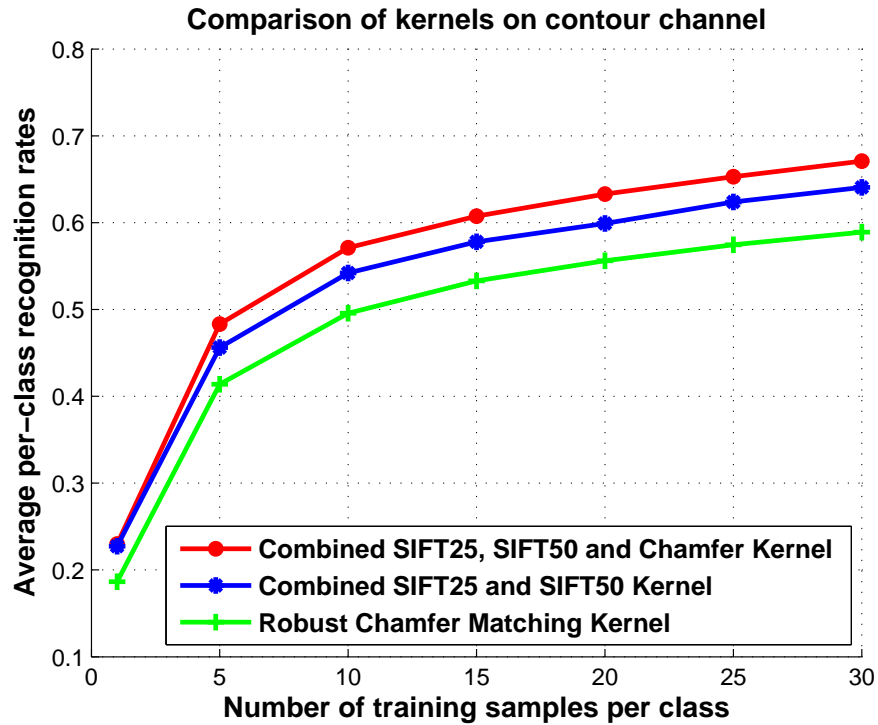


Figure 6-1: Comparison of average per-class recognition rates of the combined SIFT-spatial-pyramid-matching kernel $K_{SIFTcomb}$, the robust chamfer matching kernel $K_{rchamfer}$ and the combined contour kernel $K_{contour}$ which adaptively integrates $K_{cSIFT25}$, and $K_{cSIFT50}$ and $K_{rchamfer}$.

6.3.3 Combining Contour, Texture and Color

Effectiveness of Adaptive Combination of Multiple Visual Cues

The experiment in this section adaptively combines all available matching schemes on contour, texture and color for object categorization. The matching kernels used are: the SIFT-spatial-pyramid-matching kernels $K_{cSIFT25}$, and $K_{cSIFT50}$, the robust chamfer matching kernel $K_{rchamfer}$, the texture-spatial-pyramid-matching kernel $K_{tSIFT50}$ with patch size 50, and the color-channel kernels K_{cColor} and K_{tColor} which denote the spatial pyramid matching kernels of color in contour and texture channels with color vocabulary of size 100 (see Chapter 5 Section 5.2.3 and 5.4.2). Let K_{all_comb} denote the adaptively combined kernel:

$$K_{all_comb} = w_{cSIFT25}K_{cSIFT25} + w_{cSIFT50}K_{cSIFT50} + w_{rchamfer}K_{rchamfer} \\ + w_{tSIFT50}K_{tSIFT50} + w_{cColor}K_{cColor} + w_{tColor}K_{tColor} \quad (6.9)$$

An average kernel K_{all_avg} is also tested. The recognition rates for various numbers of training samples per class $\{1, 5, 10, 15, 20, 25, 30\}$ are shown in Table 6.4 and Figure 6-2. For comparison, the kernel $K_{contour}$ (Equation 6.8) which combines matching schemes on the contour channel alone is also shown Table 6.4 and Figure 6-2.

If the combination is done by simply averaging contour, texture and color channels, the average kernel K_{all_avg} only gives marginal improvement over the contour kernel $K_{contour}$. Compared with K_{all_avg} , when all available visual cues (contour, texture and color) are adaptively combined, the kernel K_{all_comb} gives additional boost to the recognition performance when trained with sufficient number of training samples. This shows it helps to improve recognition accuracy when multiple visual information are adaptively combined, giving more weights to the more discriminative visual cues for each pair of classes. These experimental results corroborate the postulation of human perception as discussed in Section 1.1.

The only exception lies in the recognition performance when using 1 or 5 training

samples per class, where simple average combination gives the best recognition rates, especially when there is only 1 training sample per class. In this case, for each pairwise sub-classifier, the adaptive combination by kernel alignment is only determined by two data samples, and the learned optimal combination is sensitive to data scarcity.

Training sample	1	5	10	15	20	25	30
K_{all_comb}	18.80	48.74	58.75	63.31	66.02	67.98	69.84
K_{all_avg}	24.43	48.90	57.00	61.32	63.85	65.65	67.63
$K_{contour}$	22.99	48.34	57.10	60.75	63.29	65.29	67.08

Table 6.4: Average per-class recognition rates of the combined contour kernel $K_{contour}$, and the adaptive kernel K_{all_comb} and the average kernel K_{all_avg} that combine matching schemes of contour, texture and color. Numbers are in percentile.

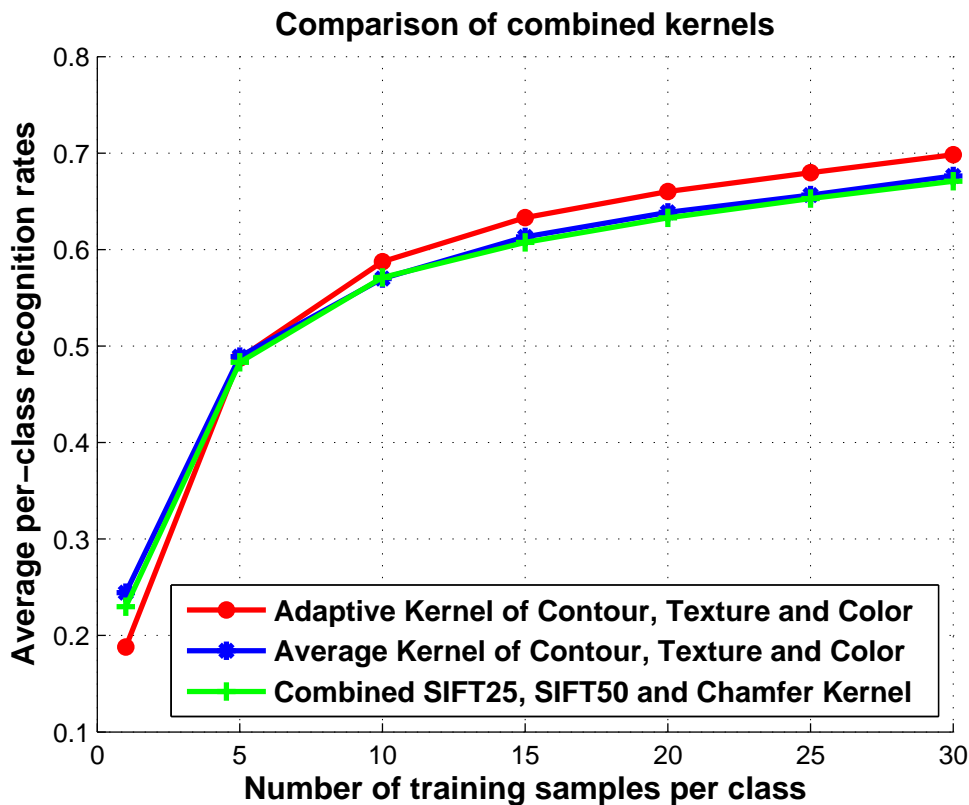


Figure 6-2: Comparison of average per-class recognition rates of the combined contour kernel $K_{contour}$, and the adaptive kernel K_{all_comb} and the average kernel K_{all_avg} that combine matching schemes of contour, texture and color.

Examples of Adaptive Combination of Multiple Visual Cues

This section demonstrates the effectiveness of adaptive visual cue combination by embedding various kernels in a 2D plane and visually showing the improvement of class pair separation. For two classes, using the weights learned in Equation 6.9, we can define kernels on different channels by summing and normalizing corresponding kernels for each channel as follows:

$$K_{contour} = \frac{w_{cSIFT25}K_{cSIFT25} + w_{cSIFT50}K_{cSIFT50} + w_{rchamfer}K_{rchamfer}}{w_{cSIFT25} + w_{cSIFT50} + w_{rchamfer}} \quad (6.10)$$

$$K_{texture} = K_{tSIFT50} \quad (6.11)$$

$$K_{color} = \frac{w_{cColor}K_{cColor} + w_{tColor}K_{tColor}}{w_{cColor} + w_{tColor}} \quad (6.12)$$

And also we define integrated weights for each channel by summing corresponding weights over each channel as follows:

$$w_{contour} = w_{cSIFT25} + w_{cSIFT50} + w_{rchamfer} \quad (6.13)$$

$$w_{texture} = w_{tSIFT50} \quad (6.14)$$

$$w_{color} = w_{cColor} + w_{tColor} \quad (6.15)$$

For a pair of classes, the kernels $K_{contour}$, $K_{texture}$, K_{color} and K_{all_comb} are embedded into a 2D Euclidean space using classical Multi-dimensional Scaling (MDS). These embeddings give a visual presentation of how well each kernel can separate the two classes. For example, for class Bonsai versus Joshua Tree, the corresponding embeddings are shown in Figure 6-4. As seen in Figure 6-4 (a)-(c), while each single visual cue can recognize some object instances of Bonsai and Joshua Tree, no single visual cue is able to separate the two classes in a clean way. It also can be seen that, contour and texture are relatively better in differentiating the two classes of Bonsai and Joshua Tree, whereas color distributions of the two classes are quite similar. The embedding of the adaptively combined kernel is shown in Figure 6-4(d). In this combined representation, the separation between Bonsai and Joshua Tree is improved. The learned weights of different channels are $\{w_{contour} = 0.6856, w_{texture} = 0.1918, w_{color} = 0.1225\}$. These weights are well adapted to the relative importance of various visual cues as discussed above, with contour as the

most important cue and texture as the second, and color as the least effective cue. As another example, the embeddings for Pizza versus Soccer Ball are shown in Figure 6-6, which shows that color plays a larger role in distinguishing these two classes, compared with Bonsai versus Joshua Tree. The learned adaptive weights are $\{w_{contour} = 0.6028, w_{texture} = 0.1572, w_{color} = 0.2399\}$. The combined kernel as shown in Figure 6-6 (d) again gives a visually better distinction between the classes of Pizza versus Soccer Ball.

To demonstrate the relative effectiveness of contour, texture and color for different classes, similar to Section 6.3.1, for each of the weights $w_{contour}, w_{texture}, w_{color}$, the characteristic weight for a class is defined as the 5th largest weight among all 100 sub-classifiers for that class, denoted as $\hat{w}_{contour}, \hat{w}_{texture}, \hat{w}_{color}$ respectively. These characteristic weights are robust indicators of the relative importance of each visual cue for different object classes. The classes with the 10 largest characteristic contour weight $\hat{w}_{contour}$, texture weight $\hat{w}_{texture}$ and color weight \hat{w}_{color} are shown in Table 6.5, 6.6 and 6.7 respectively. In Table 6.5, most of the classes have prominent shape cues. Most of the classes in Table 6.6 are rich in texture, such as dollar bill print patterns, buttons and reeds on accordions, background trees in car side views and fur of emus and cougars. And the classes in Table 6.7 have distinct colors such as black and white yin-yang images, yellow sunflowers, red flamingos and white water lilies. It is interesting to see that the classes of airplanes and ferries also have relative important color cues. The reason could be the distinctive sky and water colors in these classes.

Also noticeable is that, by comparing the characteristic weights $\hat{w}_{contour}, \hat{w}_{texture}$ and \hat{w}_{color} in Table 6.5, 6.6 and 6.7, it is apparent that the 10 largest contour weights are much larger than the 10 largest texture weights, which in turn are larger than the 10 largest color weights. Figure 6-7 shows, for each pair of classes of Caltech-101, the learned adaptive weights of contour, texture and color for classification. Overall, the same trend is observed: weights for contour are significantly larger than texture and color weights, and texture weights are slightly larger than color weights. This corroborates the observation in Section 5.4.1 that salient contours play an important role and are the dominant visual information in recognizing objects in Caltech-101.



Figure 6-3: Example images of Bonsai and Joshua Tree.

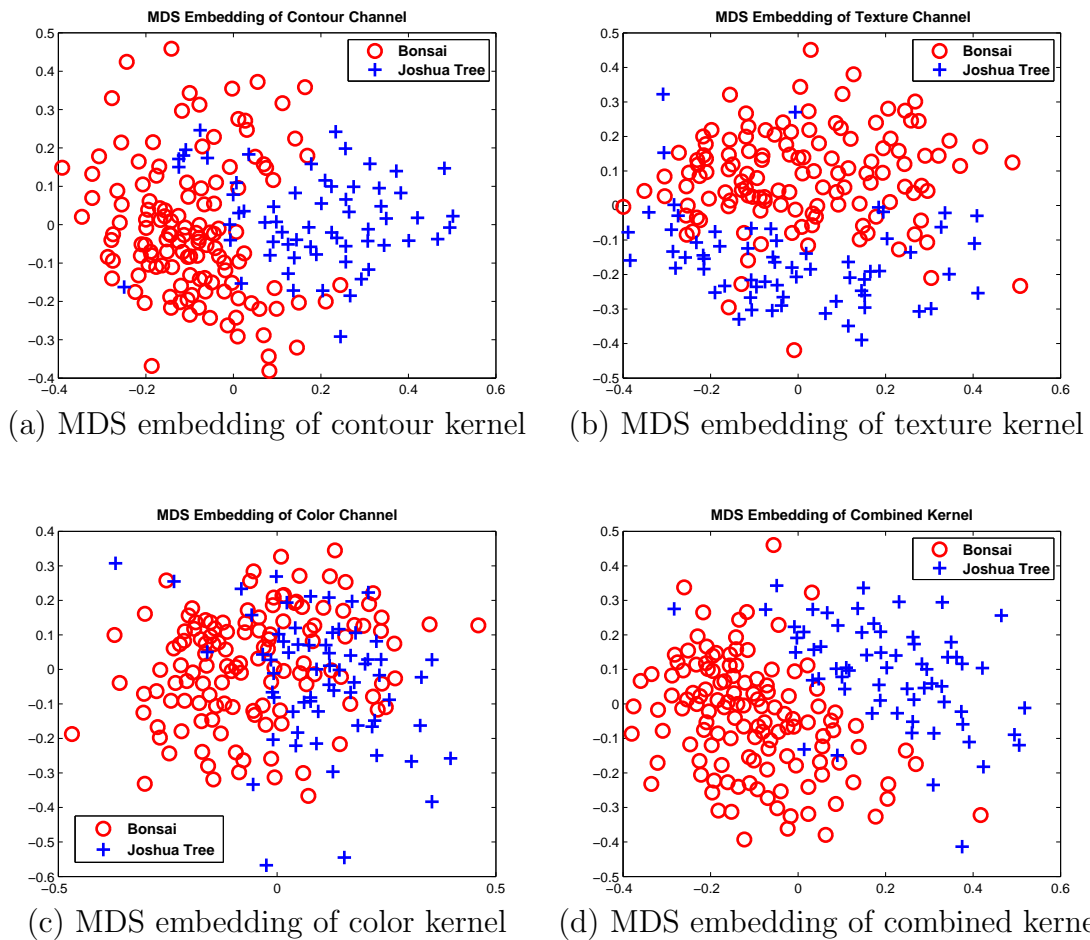


Figure 6-4: Multi-dimensional Scaling (MDS) embedding of different kernels: contour kernel, texture kernel, color kernel and adaptively combined kernel, for the class of Bonsai versus the class of Joshua Tree.



Figure 6-5: Example images of Pizza and Soccer Ball.

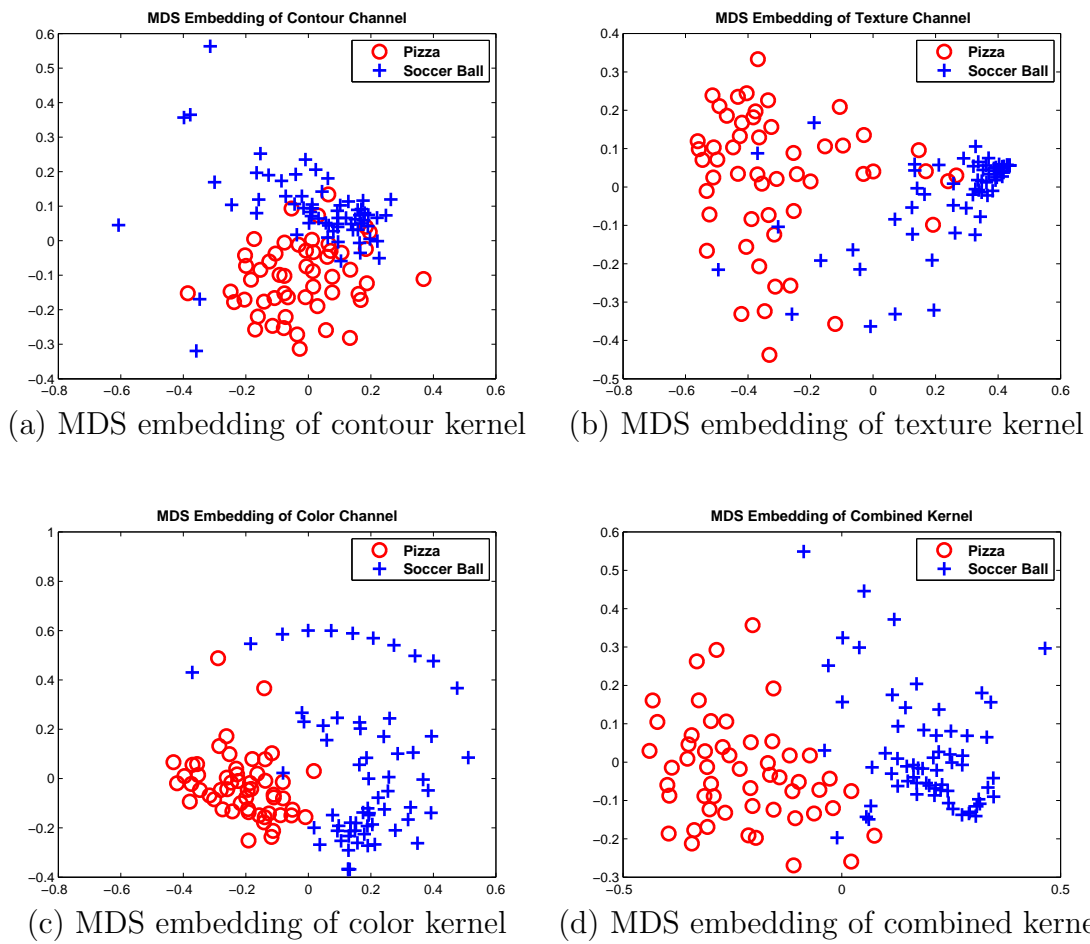


Figure 6-6: Multi-dimensional Scaling (MDS) embedding of different kernels: contour kernel, texture kernel, color kernel and adaptively combined kernel, for the class of Pizza versus the class of Soccer Ball.











Class	$\hat{w}_{contour}$	Image
Wheelchair	0.9726	
Buddha	0.9726	
Tick	0.9620	
Rooster	0.9620	
Panda	0.9548	
Ant	0.9538	
Butterfly	0.9485	
Headphone	0.9474	
Inline skate	0.9460	
Dragonfly	0.9455	

Table 6.5: Classes where the visual cue of contour plays a larger role for recognition compared with other classes.











Class	$\hat{w}_{texture}$	Image
Dollar bill	0.6677	
Accordion	0.6509	
Euphonium	0.6192	
Emu	0.6008	
Trilobite	0.5872	
Car side	0.5754	
Hedgehog	0.5620	
Hawksbill	0.5271	
Pizza	0.5242	
Cougar face	0.5101	

Table 6.6: Classes where the visual cue of texture plays a larger role for recognition compared with other classes.











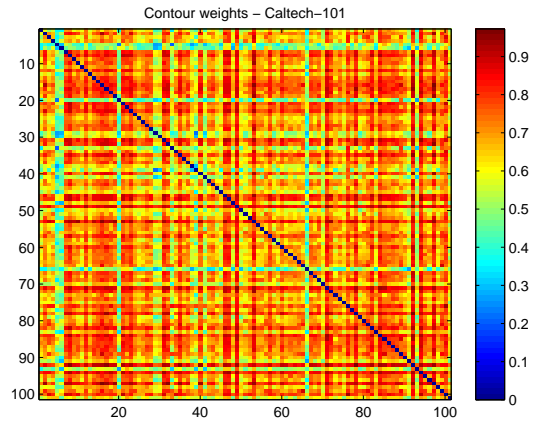
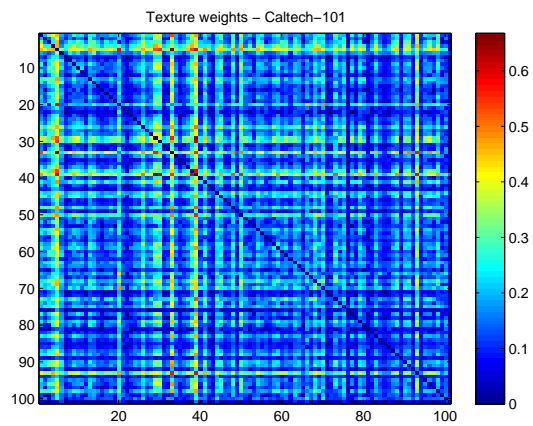
Class	\hat{w}_{color}	Image
Yin yang	0.4766	
Sunflower	0.4766	
Water lily	0.4733	
Flamingo head	0.4659	
Flamingo	0.4623	
Airplanes	0.4551	
Lotus	0.4546	
Cougar body	0.4481	
Bonsai	0.4469	
Ferry	0.4435	

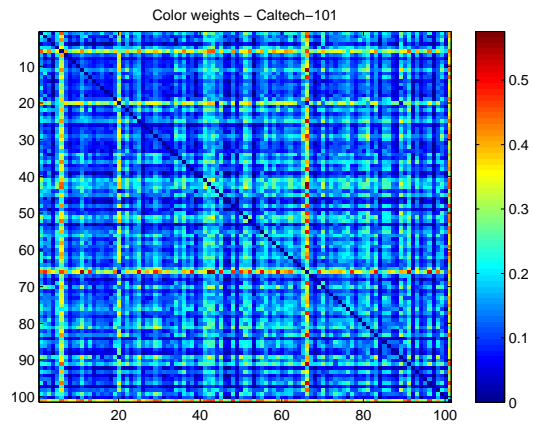
Table 6.7: Classes where the visual cue of color plays a larger role for recognition compared with other classes.



(a) Contour weights



(b) Texture weights



(c) Color weights

Figure 6-7: Learned adaptive weights for contour, texture and color when combining visual cues for classifying Caltech-101 (better view in color).

6.3.4 Comparison with Other Methods

Comparison with Non-multi-visual-cue Combination Methods

In this section, the performance of the proposed scheme in this thesis is compared against many state-of-the-art methods where visual information in patches is represented as an integrated whole [9, 49, 51, 57, 70, 88, 113], *i.e.*, visual information is not decomposed and all visual cues in patches are effectively given uniform weights. One exception is the method in [117, 118] where local appearance and global texture information are combined with fixed trade-off weights between them. I put this method in this category since in this method the two visual cues of appearance and texture are not adaptively combined. The methods mentioned above are tested on the original Caltech-101 dataset where a class of background is included.

Table 6.8 and Figure 6-8 show the comparison the proposed scheme against these methods. In this setting (*i.e.*, multiple visual cues are not decomposed and adaptively combined; the background class of Caltech-101 is included), the best performance of previously published results is achieved by [51] and [117]. Compared with [51], for {5,10,15,20,25,30} training samples per class, the proposed scheme in this thesis achieves recognition improvement of about {7.24%, 5.75%, 4.31%, 2.02%, 1.98%, 2.24%,} respectively. Compared with [117], for {5,10,15,20,30} training samples per class, the proposed scheme in this thesis achieves recognition improvement of about {3.51%, 4.14%, 4.19%, 4.13%, 3.16%,} respectively.

Since these methods, especially [51, 70, 117], and the proposed scheme in this thesis use comparable features and feature representations, it is reasonable to attribute the performance improvements to the visual decomposition model introduced in this thesis. Consistent with the behavioral and psychophysical evidence of perceptual disassociation and integration as discussed in Chapter 1, these comparison experiments demonstrate the effectiveness of the proposed visual decomposition and recombination scheme for object recognition.

The performance improvements are more significant when only a few training samples, *e.g.*, 5, 10 or 15, are available for each class. This suggests that when there

are not enough training samples, it is more important to decompose various visual cues, leverage each of them to their full potential and recombine them for a better understanding of image contents. This conforms with the capability of learning to recognize a class of objects from a few sample images in the human visual system.

The confusion matrix for classification of Caltech-101 by the proposed method in this thesis is shown in Figure 6-9.

Caltech-101 with Background Class							
Training sample	1	5	10	15	20	25	30
This Thesis	18.80	48.74	58.75	63.31	66.02	67.98	69.84
Griffin et al. CIT-TR2007[51]		41.5	53	59	64	66	67.6
Zhang PhD Thesis07[118]				62.4			
Zhang et al. CVPR06[117]	21.95	45.23	54.61	59.12	61.89		66.23
Lazebnik et al. CVPR06[70]				56.4			64.6
Mutch&Lowe CVPR06[88]				51			56
Grauman&Darrell MIT-TR06[49]	17.76	34.41	43.7	49.52	53.24	56.02	58.23
Berg et al. CVPR05[9]				45			
Wang et al. CVPR06[113]		19		44	50	58	63
Holub et al. ICCV05[57]		16.1		35.7	40.1		42.9

Table 6.8: Comparison of the proposed scheme to state-of-the-art methods where multiple visual cues are not adaptively combined.

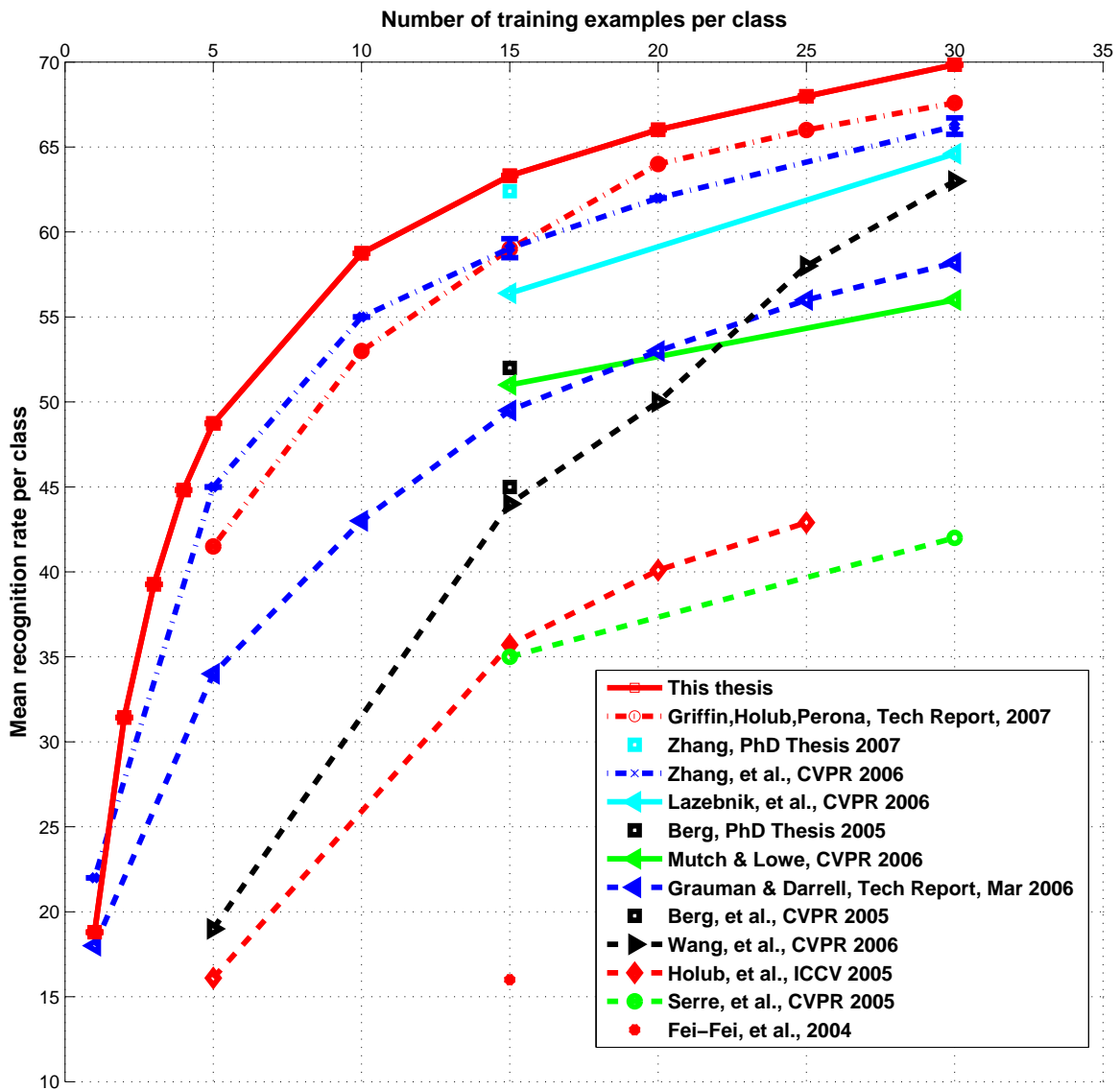


Figure 6-8: Comparison of the proposed scheme to state-of-the-art methods where multiple visual cues were not adaptively combined.

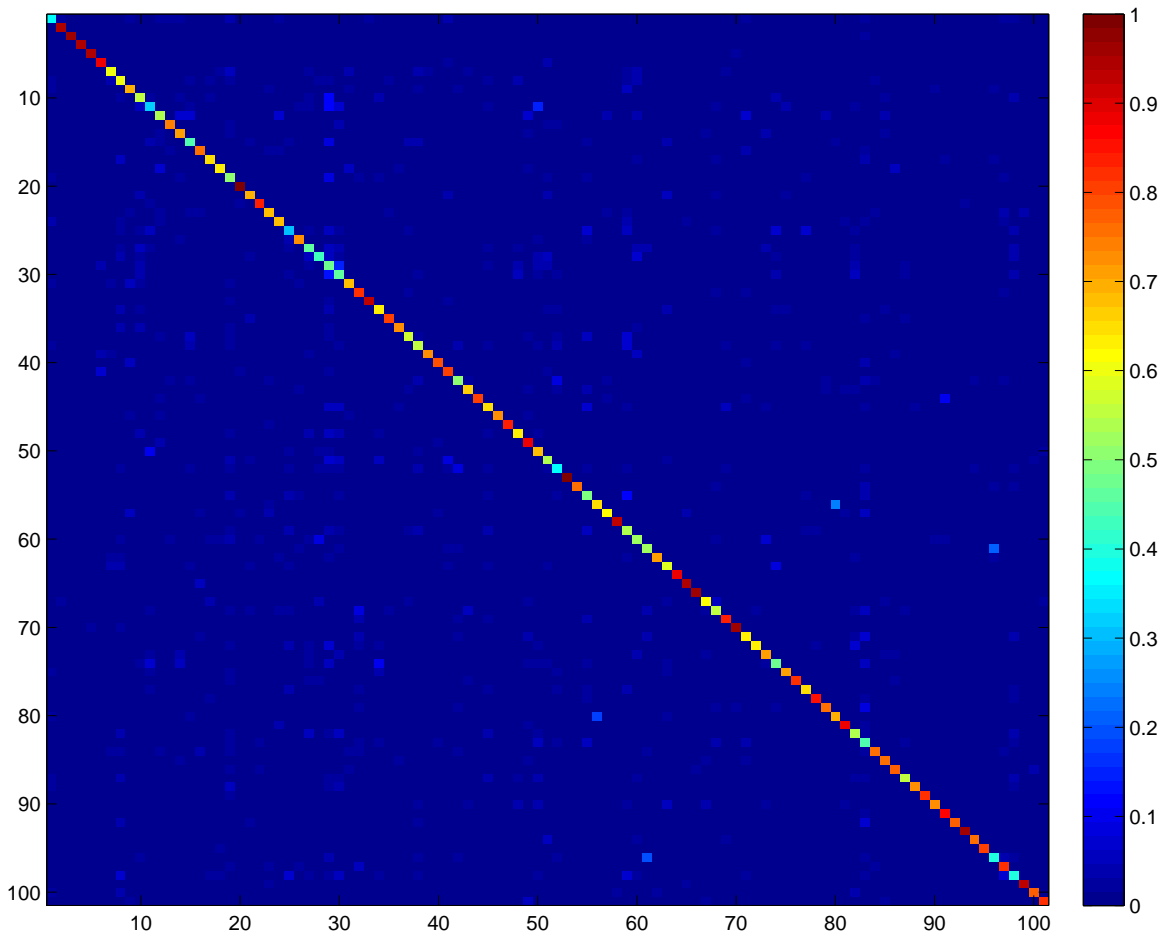


Figure 6-9: The confusion matrix for classification of Caltech-101 by the proposed method in this thesis.

Comparison with Adaptive Combination Methods with Similar Features

In [43, 44], Frome *et al.* proposed methods of adaptively learning distance functions which combine shape features and color features. In [73], Lin *et al.* extended the kernel alignment theory to incorporate localized kernel alignment for recognition tasks. Although these methods used adaptive combination of multiple distance functions or kernels, visual information is still treated in an integrative manner without visual decomposition. Since these adaptive combination methods use similar features and feature representation to the ones used in this thesis, the comparisons to these adaptive combination methods show the importance and effectiveness of visual decomposition, especially when the number of training samples is limited.

In the experiments in [43, 44], the class of background is excluded from Caltech-101 and the class of ‘faces_easy’ is added for test. The experiment in this section follows this setting. Table 6.9 and Figure 6-10 show the comparison. Except for a marginal difference for 20 training samples per class, the proposed scheme performs better, especially when there are only 5 or 10 training samples per class. The reason is again postulated to be that the proposed “recognition-through-decomposition-and-fusion” scheme is able to fully use the potential of decomposed visual cues. Whereas in [43, 44, 73], since visual information is used in an integral manner, each individual visual cue is not fully explored. When there are limited training samples, the performance of these methods drops significantly.

Caltech-101 without Background, with ‘Faces_easy’							
Training sample	1	5	10	15	20	25	30
This Thesis	18.99	49.32	59.30	63.82	66.53	68.41	70.38
Frome et al. ICCV07[44]		43.9	53.3	63.2	66.6		
Lin et al. CVPR07[73]				61.25			
Frome et al. NIPS06[43]		37.9		60.3			65.7

Table 6.9: Comparison of the proposed scheme to state-of-the-art methods where multiple distance functions or kernels were adaptively combined.

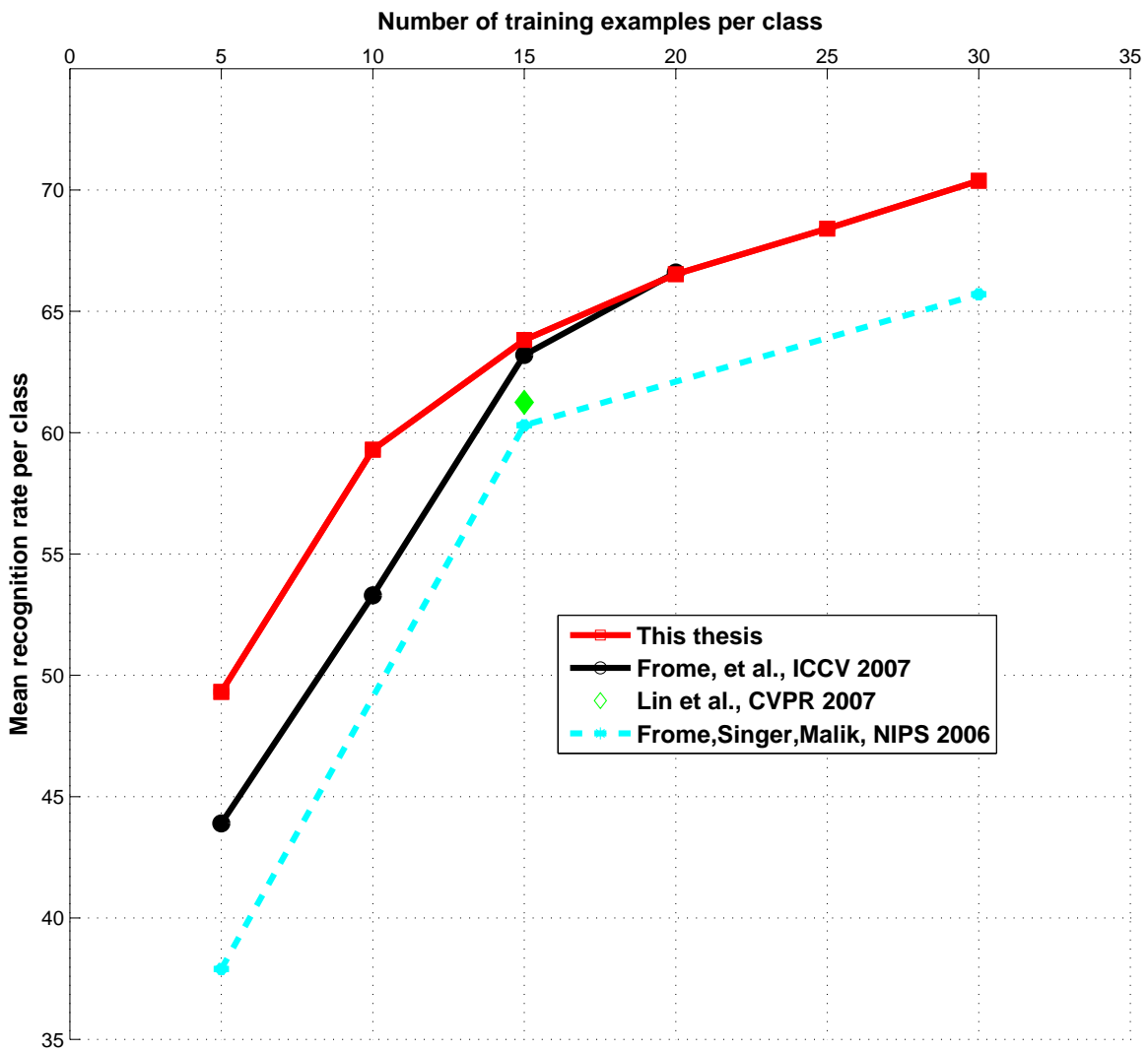


Figure 6-10: Comparison of the proposed scheme to state-of-the-art methods where multiple distance functions or kernels were adaptively combined.

6.4 Summary

In this chapter, the decomposed contour and texture channels are adaptively combined via kernel alignment for the task of object recognition, as an emulation to human observers' capability of selectively integrating multiple visual stimuli in a joint effort to distinguish different object classes. By achieving the optimal alignment to an ideal kernel, various matching kernels on the contour and texture channels are linearly combined, with learned linear weights reflecting the relative importance of each visual cue in discriminating different class pairs.

By adaptively combining multiple visual cues, the proposed computational model of "recognition-through-decomposition-and-fusion" achieves better performance than most of the state-of-the-art methods. These experimental results demonstrate the effectiveness of the visual decomposition and recombination scheme developed in this thesis. It is noticeable that, when the number of training samples is limited, the performance improvements of the proposed system are more significant, which suggests that it is more important to decompose and combine various visual stimuli when there are not enough training samples.

It is also observed that inter-class level of scale variation exists in the dataset of Caltech-101. Adaptive combination of multiple scales is able to select suitable feature scales for different classes. Weak and strong geometric matching schemes are shown to be complementary to each other. Shape contour is demonstrated to be the most important visual cue in recognizing objects in Caltech-101, with texture and color playing substantial roles in some texture-rich and color-rich classes, which are consistent with intuitive observations.

Chapter 7

Conclusion and Discussion

This chapter first reviews some recent developments in improving feature representation and kernel combination for object categorization, then shows the typical differences between the decomposed contours by the proposed coupled Conditional Random Field model and the computed probabilities of boundary by a learned model by Martin *et al.* [80], and discusses the avenues for future work based on these reviews and comparisons. In the last section, we summarize the key components and the key results of the proposed system.

7.1 Recent Developments

The experiments in this thesis have demonstrated the effectiveness of “recognition-through-decomposition-and-fusion”, by comparing to previous state-of-the-art methods with similar descriptors and feature representations. At the time of this thesis, some researchers are concurrently investigating similar ideas of combining multiple matching schemes, with improved features and enhanced adaptive combination methods than the kernel alignment theory. These enhanced methods are shown to achieve significant performance improvements. It is expected that incorporating these enhanced elements into the proposed visual decomposition and recombination model will be able to achieve further improvements.

For completeness, a brief discussion of these methods is included as follows.

In [18], Bosch *et al.* developed a spatial shape descriptor PHOG (Pyramid of Histograms of Orientation Gradients) which extends the spatial pyramid representation [70]. Combination of PHOG and a series of SIFT features on various scales is done on the kernel level. The optimal combination is carried out with an exhaustive search. Classification is done with a one-versus-all SVM. When kernels are combined with globally optimal weights, they achieve 71.5% average recognition rates with 30 training samples per class for Caltech-101. When kernels are combined with optimal weights for each class, their method has 77.8% average recognition rates.

In an improved version [17], Bosch et al. used the same set of features as in [18], added automatically selected ROI (Region of Interest) to confine the matching to objects of interest only, and incorporated random forests and ferns to automatically combine multiple features. For 30 training samples per class, they achieve 79.2% - 81.3% recognition rates for various versions of classifiers.

In [110], Varma and Ray developed a multi-kernel combination method, by augmenting the original SVM training scheme to learn the optimal kernel combination with a sparsity constraint. They adopted the two kernels in [117] and the four enhanced features and kernels in [18]. For 15 training samples per class, with an adaptive one-versus-one SVM, they achieve 78.43% for Caltech-101; with an adaptive one-versus-all SVM, they achieve 87.82%.

It appears that the main sources of the performance improvements of the above methods are improved feature representations such as PHOG and detected ROI, and enhanced kernel combination schemes such as learning kernel combination in the SVM training with sparsity constraints. In this thesis, decomposition of visual stimuli has been shown to be effective for fully leveraging various visual cues and achieve significant performance improvements compared to non-adaptive methods with comparable features, especially when the number of training samples is limited. It will be interesting to incorporate the above recently developed features and kernel combination methods to the proposed framework of “recognition-through-decomposition-and-fusion”.

7.2 Comparison to “Learning Probabilities of Boundary”

In [80], Martin *et al.* used a similar set of low-level image measurements, such as the orientation energy and the gradient of brightness, color and texture, to learn the probabilities of boundary in natural images. In Figure 7-1, we visually compare the results by the approach of “learning the probabilities of boundary” in [80] and the coupled Conditional Random Field model in this thesis. As can be seen in Figure 7-1(b), for two images from the Weizmann horse database [14] in Figure 7-1(a), the learned boundary model in [80] generates many spurious strong contours. To suppress these spurious contours, nearby compatible contour and texture pixels need to be considered to correct the errors. As discussed in Chapter 4, these compatibilities of contour and texture are captured by the proposed coupled Conditional Random Field model. The decomposed contours of the two horse images by the coupled Conditional Random Field model are shown in Figure 7-1(c). The coupled Conditional Random Field model gives much cleaner contours, especially on the horse bodies.

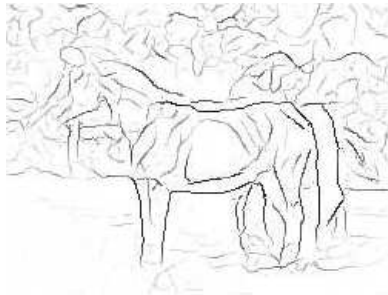
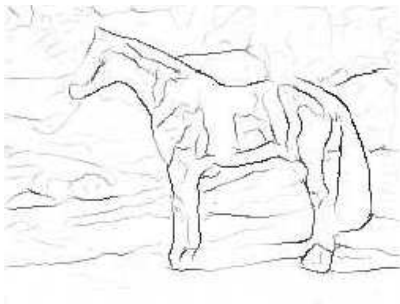
Since the seminal work of Martin *et al.* [80], there has been much work which extends the original framework of learning the probabilities of boundary. For example, Arbelaez [4] combined region information with local contour cues to achieve better boundary detections. Ren *et al.* [92] used a manually segmented training set to build a class-specific shape model to improve the results. We expect the coupled Conditional Random Field model can be improved in similar ways, by incorporating mid-level and high-level image measurements, such as coherent image regions and class-specific shapes.

7.3 Summary and Conclusion

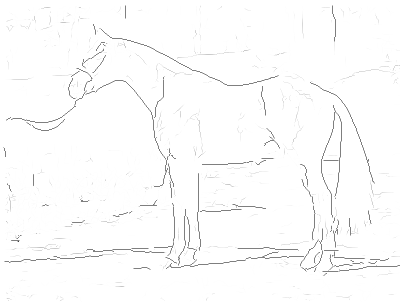
It is shown by behavioral and psychophysical experiments in associationist psychology that the visual stimuli of contour and texture are processed separately in early stages of human visual perception, and recombined at a higher level of visual



(a) Horse images from the Weizmann horse database [14]



(b) Probabilities of boundary obtained by the learned model by Martin *et al.* [80]



(c) Decomposed contours by the coupled Conditional Random Field model

Figure 7-1: Comparisons of the probabilities of boundary obtained by the learned model by Martin *et al.* [80] and the decomposed contour channels by the coupled Conditional Random Field model.

processing. This thesis proposes a computational system of “recognition-through-decomposition-and-fusion” to emulate the dissociation and integration properties of human visual perception. Four key components of the proposed system are introduced and studied. At the lowest level, contour and texture processes are defined and measured. In the mid-level, a novel coupled Conditional Random Field is proposed to model the contour and texture processes in natural images. The learned coupled Conditional Random Field model is able to well decompose the different visual stimuli of contour and texture. Various matching schemes are introduced to match the decomposed contour and texture channels in a dissociative manner. The decomposition enables the system to fully leverage each decomposed visual stimulus to its full potential in discriminating different object classes. As a counterpart to the integrative process in the human visual system, various matching schemes on the decomposed contour and texture channels are adaptively combined. The learned adaptive linear weights for visual cue combination mirror the fact that different visual cues play different roles in distinguishing various object classes. Experimental results of object recognition on Caltech-101 demonstrate that the proposed computational model of “recognition-through-decomposition-and-fusion” achieves better performance than most of the current best methods.

The key results are two-fold. The proposed coupled Conditional Random Field model is shown to be an important extension of popular single-layer Random Field models for modeling image processes. By dedicating a separate layer of random field grid to each individual image process, the proposed model is able to capture the distinct properties of multiple visual processes, by explicitly modeling different interactive dynamics of different image processes. On the contrary, in order to accommodate different characteristics of multiple visual processes, a single-layer Conditional Random Field is shown to be forced to model the disparate image processes with only a single layer of random field, which leads to degraded modeling power. The coupled Conditional Random Field is demonstrated to outperform the single-layer Conditional Random Field for the task of dissociating the visual stimuli of contour and texture. The second key result is shown by empirical object recognition exper-

iments on Caltech-101. By decomposing and recombining contour and texture, the proposed computational model is able to select best visual cues for discriminating different class pairs, and achieve a better object categorization system. More importantly, when there are only a small number of training images, it is shown that decomposition of various visual cues is more valuable, by leveraging each visual cue to its full potential and recombining multiple visual stimuli for a better understanding of image contents. In this case, the proposed computational model is demonstrated to achieve significant improvements of recognition performance. This is consistent with the fact that human observers are able to recognize object classes with only a few sample images.

Some other aspects of object recognition are also studied in this thesis. It is observed by both matching decomposed channels individually and combining multiple visual channels that the shape contour information is more prominent and important in recognizing the objects in Caltech-101. It is reasonable to postulate that salient contours are the dominant visual cue for many classes of objects in Caltech-101. Weak and strong geometric matchings are demonstrated to be complementary to each other. Employing better geometric matching schemes are expected to further improve the performance. Inter-class level of scale variation is demonstrated to exist in Caltech-101 and scale adaptation is able to improve the performance by selecting the best scales for different object classes.

While in this thesis we explored many aspects of the proposed system, the key message to be conveyed is that by decomposing and recombining multiple disparate visual cues in object images, the proposed computational system of object categorization is able to adapt to the discriminative visual cues for different classes and achieve improved object recognition performance, especially when only a limited number of training image are available. The framework of recognition by decomposition and recombination of visual stimuli is expected to be a promising direction for building effective and efficient object categorization systems.

Bibliography

- [1] R. Abadi, J. J. Kulikowski, and P. Meudell. Visual performance in a case of visual agnosia. In M. van Hof and E. Mohn, editors, *Functional Recovery from Brain Damage*. Elsevier, Amsterdam, 1981.
- [2] S. Agarwal, A. Awan, , and D. Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11).
- [3] E. Allwein, R. Schapire, and Y. Singer. Reducing multiclass to binary: a unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1:113–141, September 2001.
- [4] P. Arbelaez. Boundary extraction in natural images using ultrametric contour maps. In *IEEE CVPR Workshop on Perceptual Organization in Computer Vision*, New York, NY, June 2006.
- [5] A. Bar-Hillel, T. Hertz, and D. Weinshall. Object class recognition by boosting a part-based model. In *Proceedings of the IEEE Conference on Computer Vision Pattern Recognition*, pages 702–709, San Diego, CA, June 2005.
- [6] L. Battelli and G. Sartori. Dissociation between contour-based and texture-based shape perception: A single case study. *Visual Cognition*, 4(3):275–310, September 1997.
- [7] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(24):590–522, April 2002.
- [8] K. Bennett, M. Momma, and M. Embrechts. Mark: a boosting algorithm for heterogeneous kernel models. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 24–31, Edmonton, Canada, July 2002.
- [9] A. Berg, T. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. In *Proceedings of the IEEE Conference on Computer Vision Pattern Recognition*, pages 26–33, San Diego, CA, June 2005.

- [10] A. Berg and J. Malik. Geometric blur for template matching. In *Proceedings of the IEEE Conference on Computer Vision Pattern Recognition*, pages 607–614, Hawaii, December 2001.
- [11] J. Besag. Statistical analysis of non-lattice data. *The Statistician*, 24:179–195, 1975.
- [12] R. Bolles and R. Cain. Recognizing and localizing partially visible objects: The local-features-focus method. *International Journal of Robotics Research*, 1(3):57–82, 1982.
- [13] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585, 1989.
- [14] E. Borenstein, E. Sharon, and S. Ullman. Combining top-down and bottom-up segmentations. In *IEEE CVPR Workshop on Perceptual Organization in Computer Vision*, Washington, DC, June 2004.
- [15] G. Borgefors. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6).
- [16] G. Borgefors. Hierarchical chamfer matching: a parametric edge matching algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):849–865, November 1988.
- [17] A. Bosch, A. Zisserman, and X. Munoz. Image classification using random forests and ferns. In *Proceedings of the Tenth IEEE International Conference on Computer Vision*, Rio de Janeiro, Brazil, October 2007.
- [18] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 401–408, 2007.
- [19] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, November 2001.
- [20] R.A. Brooks. Symbolic reasoning among 3d models and 2d images. *Artificial Intelligence*, 17:285–348, 1981.
- [21] H. Bulthoff and S. Edelman. Psychophysical support for a two-dimensional view interpolation theory of object recognition. In *Proceedings of the National Academy of Sciences*, pages 60–64, 1992.
- [22] H. Bulthoff, S. Edelman, and M. Tarr. How are three-dimensional objects represented in the brain? *Cerebral Cortex*, 5:247–260, 1995.

- [23] H. H. Bulthoff, S. Edelman, and M. Tarr. How are three-dimensional objects represented in the brain? *Cerebral Cortex*, 5(3):247–260, May–Jun 1995.
- [24] M. C. Burl, M. Weber, and P. Perona. A probabilistic approach to object recognition using local photometry and global geometry. In *Proceedings of the 5th European Conference on Computer Vision*, pages 628–641, London, UK, 1998.
- [25] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.
- [26] P. Cavanagh, M. Arguin, and A. Treisman. Effect of surface medium on visual search for orientation and size features. *Journal of Experimental Psychology: Human Perception and Performance*, 163(3):479–491, August 1990.
- [27] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [28] H. Chiu, L. P. Kaelbling, and T. Lozano-Perez. Virtual training for multi-view object class recognition. In *Proceedings of the IEEE Conference on Computer Vision Pattern Recognition*, Minneapolis, MN, June 2007.
- [29] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):853–857, June 2001.
- [30] N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. Kandola. On kernel-target alignment. In *Advances in Neural Information Processing Systems*, pages 367–373, Vancouver, Canada, December 2001.
- [31] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Proceedings of the ECCV International Workshop on Statistical Learning in Computer Vision*, pages 1–22, Prague, Czech Republic, May 2004.
- [32] I. Diego, J. Moguerza, and A. Munoz. Combining kernel information for support vector classification. *Lecture Notes in Computer Science*, 3077:102–111, September 2004.
- [33] T. G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1995.
- [34] S. Edelman and T. Poggio. Bringing the grandmother back into the picture: A memory-based view of object recognition. Technical Report AIM-1181, Massachusetts Institute of Technology, Cambridge, MA, USA, 1990.

- [35] D. C. Van Essen, C. H. Anderson, and D. J. Felleman. Information processing in the primate visual system: an integrated systems perspective. *Science*, 255(5043):419–423, January 1992.
- [36] D. C. Van Essen and J. H. R. Maunsell. Hierarchical organization and functional streams in the visual cortex. *Trends in Neuroscience*, 6(9):370–375, 1983.
- [37] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In *IEEE CVPR Workshop on Generative Model Based Vision*, Washington, DC, June 2004.
- [38] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proceedings of the IEEE Conference on Computer Vision Pattern Recognition*, pages 264–271, Madison, WI, June 2003.
- [39] R. Fergus, P. Perona, and A. Zisserman. A sparse object category model for efficient learning and exhaustive recognition. In *Proceedings of the IEEE Conference on Computer Vision Pattern Recognition*, pages 380 – 387, San Diego, CA, June 2005.
- [40] M.A. Fischler and R.A. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computer*, 22(1):67–92, January 1973.
- [41] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael. Learning low-level vision. *International Journal of Computer Vision*, 40(1):25–47, October 2000.
- [42] W.T. Freeman and E.H. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906, September 1991.
- [43] A. Frome, Y. Singer, and J. Malik. Image retrieval and classification using local distance functions. In *Advances in Neural Information Processing Systems*, pages 417–424, Vancouver, Canada, December 2006.
- [44] A. Frome, Y. Singer, F. Sha, and J. Malik. Learning globally-consistent local distance functions for shape-based image retrieval and classification. In *Proceedings of the Tenth IEEE International Conference on Computer Vision*, Rio de Janeiro, Brazil, October 2007.
- [45] B. V. Funt and G. D. Finlayson. Color constant color indexing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):522–529, May 1995.
- [46] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):721–741, November 1984.

- [47] S. Geman and C. Graffigne. Markov random field image models and their applications to computer vision. In *Proceedings of the International Congress of Mathematicians*, 1986.
- [48] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 1458–1465, Beijing, China, October 2005.
- [49] K. Grauman and T. Darrell. Pyramid match kernels: Discriminative classification with sets of image features. CSAIL Technical Report MIT-CSAIL-TR-2006-020, Massachusetts Institute of Technology, Cambridge, MA, USA, 2006.
- [50] D. M. Greig, B. T. Porteous, and A. H. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society: Series B*, 51(2):271–279, 1989.
- [51] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007.
- [52] W. E. L. Grimson and T. Lozano-Perez. Localizing overlapping parts by searching the interpretation tree. *IEEE Trans. PAMI*, 9(4):469–482, July 1987.
- [53] C. Harris and M. Stephens. A combined corner and edge detection. In *Proceedings of The Fourth Alvey Vision Conference*, pages 147–151, 1988.
- [54] X He, R.S Zemel, and M.A Carreira-Perpinan. Multiscale conditional random fields for image labeling. In *Proceedings of the IEEE Conference on Computer Vision Pattern Recognition*, pages 695–702, Washington, DC, June 2004.
- [55] Thomas Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(2):177–196, January 2001.
- [56] S. Hoi, M. Lyu, and E. Chang. Learning the unified kernel machines for classification. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 187–196, Philadelphia, PA, August 2006.
- [57] A. D. Holub, M. Welling, and P. Perona. Combining generative models and fisher kernels for object recognition. In *Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 17–21, Beijing, China, October 2005.
- [58] B.K.P. Horn. Image intensity understanding. Technical Report 335, M.I.T. Artificial. Intelligence Laboratory Memo, 1975.
- [59] G. W. Humphreys and M. J. Riddoch. *To See But Not To See: A Case Study Of Visual Agnosia*. Psychology Press, 1987.

- [60] D. P. Huttenlocher, G. A. Klanderman, and W. A. Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, September 1993.
- [61] D. P. Huttenlocher and S. Ullman. Recognizing solid objects by alignment with an image. *International Journal of Computer Vision*, 5(2):195–212, November 1990.
- [62] A. A. Efros, A. Zisserman, J. Sivic, B. C. Russell and W. T. Freeman. Discovering objects and their location in images. In *Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 370–377, Beijing, China, October 2005.
- [63] B. Julesz. Textons, the elements of texture perception, and their interactions. *Nature*, 290(12):91–97, 1981.
- [64] T. Kadir and M. Brady. Saliency, scale and image description. *International Journal of Computer Vision*, 45(2).
- [65] J. Kittler, M. Hatef, R. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3).
- [66] S. Kumar and M. Hebert. Discriminative random fields: A discriminative framework for contextual interaction in classification. In *Proceedings of the Ninth IEEE International Conference on Computer Vision*, pages 1150–1157, Nice, France, October 2003.
- [67] J Lafferty, A McCallum, and F Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289, Williamstown, MA, June 2001.
- [68] W. Lamdan and H. Wolfson. Geometric hashing: A general and efficient model-based recognition scheme. In *Proceedings of the Second IEEE International Conference on Computer Vision*, pages 5–8, Tampa, FL, December 1988.
- [69] G. Lanckriet, N. Cristianini, P. Bartlett, L. Ghaoui, and M. Jordan. Learning the kernel matrix with semi-definite programming. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 323–330, Sydney, Australia, July 2002.
- [70] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the IEEE Conference on Computer Vision Pattern Recognition*, pages 2169–2178, New York, NY, June 2006.
- [71] B. Leibe, A. Leonardis, and B. Schiele. An implicit shape model for combined object categorization and segmentation. *Lecture Notes in Computer Science*, 4170:590–522, 2006.

- [72] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1):29–44, 2001.
- [73] Y. Lin, T. Liu, and C. Fuh. Local ensemble kernel learning for object category recognition. In *Proceedings of the IEEE Conference on Computer Vision Pattern Recognition*, Minneapolis, MN, June 2007.
- [74] T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):79–116, November 1998.
- [75] D. G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3):355–395, March 1987.
- [76] D. G. Lowe. Edge detection and ridge detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):117–156, November 1998.
- [77] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.
- [78] J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. *International Journal of Computer Vision*, 43(1):7–27, 2001.
- [79] R. Maree, P. Geurts, J. Piater, and L. Wehenkel. Random subwindows for robust image classification. In *Proceedings of the IEEE Conference on Computer Vision Pattern Recognition*, pages 34–40, San Diego, CA, June 2005.
- [80] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):530–549, 2004.
- [81] J. Matas, O. Chum, U. Martin, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the British Machine Vision Conference*, pages 384–393, Cardiff, UK, September 2002.
- [82] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, October 2006.
- [83] T. Minka. A comparison of numerical optimizers for logistic regression. In *Carnegie Mellon University Statistics Tech Report 758*, 2004 revision.
- [84] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):696–710, July 1997.
- [85] J. Moguerza, A. Munoz, and I. Diego. Improving support vector classification via the combination of multiple sources of information. *Lecture Notes in Computer Science*, 3138:592–600, October 2004.

- [86] M. C. Morrone and R. A. Owens. Feature detection from local energy. *Pattern Recognition Letters*, 6:303–313, December 1987.
- [87] H. Murase and S. K. Nayar. Visual learning and recognition of 3-d objects from appearance. *International Journal of Computer Vision*, 14(1):5–24, 1995.
- [88] J. Mutch and D. G. Lowe. Multiclass object recognition with sparse, localized features. In *Proceedings of the IEEE Conference on Computer Vision Pattern Recognition*, pages 11–18, New York, NY, June 2006.
- [89] S. Palmer, E. Rosch, and P. Chase. Canonical perspective and the perception of objects. In J. Long and A. Baddeley, editors, *Attention and Performance IX*, pages 135–151. Lawrence Erlbaum Associates, Hillsdale, NJ, 1981.
- [90] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc., 1988.
- [91] P. Perona and J. Malik. Detecting and localizing edges composed of steps, peaks and roofs. In *Proceedings of the Third IEEE International Conference on Computer Vision*, pages 4–7, December 1990.
- [92] X. Ren, C. Fowlkes, and J. Malik. Cue integration in figure/ground labeling. In *Advances in Neural Information Processing Systems*, 2005.
- [93] M. J. Riddoch and G. W. Humphreys. A case study of integrative visual agnosia. *Brain*, 110(6):1431–1462, 1987.
- [94] C.J. Van Rijsbergen. *Information Retrieval*. London: Butterworths, 1979.
- [95] L. G. Roberts. *Machine Perception of Three Dimensional Solids*. PhD thesis, Massachusetts Institute of Technology, 1963.
- [96] K. Rose, E. Gurewitz, and G. Fox. A deterministic annealing approach to clustering. *Pattern Recognition Letters*, 11(11):589–594, 1990.
- [97] Dan Roth. Learning to resolve natural language ambiguities: a unified approach. In *Proceedings of the fifteenth national conference on Artificial Intelligence*, pages 806–813, Madison, WI, 1998.
- [98] W. J. Rucklidge. Efficiently locating objects using the hausdorff distance. *International Journal of Computer Vision*, 24(3):251–270, November 2004.
- [99] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or how do i organize my holiday snaps? In *Proceedings of the 7th European Conference on Computer Vision*, pages 414–431, 2002.
- [100] B. Schiele and J. L. Crowley. Object recognition using multidimensional receptive field histograms. In *Proceedings of the 4th European Conference on Computer Vision*, pages 610–619, London, UK, 1996.

- [101] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–535, May 1997.
- [102] J. Shotton, A. Blake, and R. Cipolla. Contour-based learning for object detection. In *Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 503–510, Beijing, China, October 2005.
- [103] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proceedings of the Ninth IEEE International Conference on Computer Vision*, pages 1470–1477, Nice, France, October 2003.
- [104] D. Slater and G. Healey. The illumination-invariant recognition of 3d objects using local color invariants. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(2):206–210, February 1996.
- [105] E. Sudderth, A. Torralba, W. Freeman, , and A. Willsky. Describing visual scenes using transformed dirichlet processes. In *Neural Information Processing Systems*, 2005.
- [106] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, November 2001.
- [107] A. Thayananthan, B. Stenger, P. H. S. Torr, and R. Cipolla. Shape context and chamfer matching in cluttered scenes. In *Proceedings of the IEEE Conference on Computer Vision Pattern Recognition*, pages 127–133, Madison, WI, June 2003.
- [108] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *Proceedings of the IEEE Conference on Computer Vision Pattern Recognition*, pages 586–591, 1991.
- [109] S. Ullman and R. Basri. Recognition by linear combinations of models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):992–1006, October 1991.
- [110] M. Varma and D. Ray. Learning the discriminative power-invariance trade-off. In *Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 14–21, Rio de Janeiro, Brazil, October 2007.
- [111] M. Varma and A. Zisserman. A statistical approach to texture classification from single images. *International Journal of Computer Vision*, 62(1-2):61–81, April 2005.
- [112] T. Vetter and T. Poggio. Linear object classes and image synthesis from a single example image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):733–742, July 1997.

- [113] G. Wang, Y. Zhang, and L. Fei-Fei. Using dependent regions for object categorization in a generative framework. In *Proceedings of the IEEE Conference on Computer Vision Pattern Recognition*, pages 1597–1604, New York, NY, June 2006.
- [114] M. Weber, M. Welling, and P. Perona. Towards automatic discovery of object categories. In *Proceedings of the IEEE Conference on Computer Vision Pattern Recognition*, pages 101–108, Hilton Head Island, SC, June 2000.
- [115] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *Proceedings of the 6th European Conference on Computer Vision*, pages 18–32, London, UK, 2000.
- [116] S. M. Zeki. The functional organization of projections from striate to prestriate visual cortex in the rhesus monkey. In *Cold Spring Harbor Symposia on Quantitative Biology*, pages 591–600, 1976.
- [117] H. Zhang, A. C. Berg, M. Maire, and J. Malik. Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In *Proceedings of the IEEE Conference on Computer Vision Pattern Recognition*, pages 2126–2136, New York, NY, June 2006.
- [118] Hao Zhang. *Adapting Learning Techniques for Visual Recognition*. PhD thesis, University of California, Berkeley, 2007.
- [119] X. Zhou and B. Bhanu. Feature fusion of face and gait for human recognition at a distance in video. In *Proceedings of the 18th International Conference on Pattern Recognition*, pages 529–532, Atlanta, GA, October 2006.